Developing an artificial algorithm to predict energy consumption in a traditional house adapting to external environmental conditions and occupation

by

Daniel Nicholas de Bruyn

Thesis for the degree of

Doctor of Engineering in Electrical Engineering

in the

Department of Electric, Electronic and Computer Engineering

of the

Faculty of Engineering, Built Environment and Information Technology

at the

Central University of Technology, Free State

March 2020

Promotor: Dr. BJ Kotze, D Tech. (Eng.)

Declaration

I, DANIEL NICHOLAS DE BRUYN, identity number and student number , do hereby declare that this research project submitted to the Central University of Technology, Free State for the degree DOCTER OF ENGINEERING: ELECTRICAL, is my own independent work, and complies with the Code of Academic Integrity, as well as other relevant policies, procedures, rules and regulations of the Central University of Technology, Free State; and has not been submitted before to any institution by myself of any other person in fulfilment (or partial fulfilment) of the requirements for the attainment of any qualification.

2020/11/30

……………………………………………. ……………………

Signature of Student Date

Acknowledgments

- I firstly wish to thank my Heavenly Father for giving me this ability to complete this study. Secondly, I would like to thank my wife and children for their extreme support and understanding of the times I could not spend with them during this study. I would like to thank the following individuals and institutions in particular:
    - Special thanks to Dr Ben Kotze for his supervision, guidance and advice.
    - The Central University of Technology, Free State for the financial support.
    - My parents for giving me opportunities in life and always believes in me.

Abstract

The aim of the research was to development a prediction model that could control electricity consumption in a traditional house, adapting to external environmental conditions and occupation. As identified high-end users of electricity, a backup geyser element in a solar geyser solution, including a swimming pool pump, was identified to be controlled more efficient than a time-based controller, as an end result with the same or higher customer satisfaction index to the occupant.

A system was developed, recording multiple remote sensor readings from Internet of Things (IoT) devices to a central SQL database which included the hot water usage and heating patterns. Official weather predictions replaced physical sensors, recording environmental conditions.

Fuzzifying the sensor recordings into four linguistic terms simplified hot water usage. Partitioning clustering was used to determine relationship patterns between weather predictions and solar heating efficiency. A Hidden Markov Model was used to profile the system, with the Viterbi algorithm calculating the geyser and pool solar heating predictions, while the Baum-Welch algorithm trained the system.

The solar geyser's standby element was controlled by a time-based controller heating water to pre-set value if required. During this investigation the time-based controller was simulated with a controller profiling weather conditions and hot water usage. Adjusting pre-set temperatures accordingly without human intervention. The same algorithm was used to simulate a timer controlling a swimming pool pump, but excluding the hot water usage.

Simulations found that profile base prediction controller can increase user satisfactions, including a maximum of twenty percent electrical power savings against time-based controllers.

Keywords: solar geyser heating; solar pool heating; profile usage; profile weather conditions; Fuzzy logic; partial clustering; Hidden Markov Model; Esp8266.

Table of Contents

List of Tables

List of Figures

Figure 5.72 Geyser Parameters 239

Acronyms and abbreviations

Abbreviation	**Description**
ADC	Analog to Digital Converter
AES	Advance Encryption Standards
AI	Artificial Intelligence
AP	Access Point
CoG	Centre of Gravity
CLARA	Clustering LARge Application
CLARANS	Clustering LARge Application based upon Randomised Search
EM	Expectation Maximisation
FLC	Fuzzy Logic Controller
HMM	Hidden Markov Model
HWC	Hotwater cylinder
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
I2C	Inter-Integrated Circuit
IoT	Internet of Things
IP	Internet Protocol
LED	Light Emitting Diode
MC	Markov Chain
MoM	Mean of Maximum
PAM	Partition Around Medoids
Pcb	Printed circuit board
PLC	Power-line Communications
PWM	Pulse Width Modulation

RAM	Random Access Memory
RTC	Real Time Clock
SPI	Serial Peripheral Interface
SSID	Service Set Identifier
SSE	Sum of the squared error
TCP	Transport Control Protocol
UDP	User Datagram Protocol
UART	Universal Asynchronous Receiver/Transmitter
WPA2	Wi-Fi Protected Access Version 2

Chapter 1

1 Introduction

Modelling human behaviour is challenging due to human nature to make unpredictable actions. The goal of Artificial Intelligence (AI) is to design systems that can simulate human behaviour [1, pp.1-2].

This research investigated the availability of a suitable model to predict energy consumption of a controlled environment, such as a normal household. This model also needed to adapt to external environmental conditions and occupation.

An example of this model would be a smart geyser controller that ensures that the correct water temperature is obtained for a predefined period. If the property on which the geyser is located is unoccupied, the controller executes its program to heat the geyser water to the predefined temperature. Current controllers would not adapt their start time to external temperature conditions or changes in the number of occupants. The controller only takes temperature and time into account and cannot adapt its program according to the surrounding conditions; the same solution used will then be implemented for a heated swimming pool and automated lighting.

1.1 *Problem Statement*

Time based smart home controllers controls its environment with a predefined routine, unknowledgeable if this is the most energy efficient way. It activates or deactivates specific applications based on a timed trigger. It does not have the ability to alter its predefined algorithm to support changes in surrounding conditions. Developing a suitable prediction model to predict energy consumption could proof to be a more efficient way in control electricity usage. The model could also adapt to external environmental conditions and occupation.

1.2 *Hypothetical Solution*

A solution would be the development of a prediction model in controlling the energy usage in a traditional household, with the added advantage of minimal human interaction setting up the controller, hot water availablity and electrical energy saving without impacting the service delivered.

1.3 *Methodology & Research Design*

The breakup of the project entailed the following foreseen as feasible requirements:

- The ability to monitor and record the following values or data:
  1. power consumption,
  2. time and date,
  3. movement of occupants,
  4. temperature and
  5. humidity.

- A computer program to be developed that recorded data from the sensors or internet and analysed the information. It simulated and controlled the remote devices in the same behavioural pattern as recorded by the household.

- The research was to prove that a
  1. geyser with solar heating,
  2. swimming pool pump with solar heating, and
  3. bedroom light

could be controlled more efficiently in a non-laboratory environment, adapting to changing environmental conditions and occupation.

- Identificaion of the most appropriate model/s to be investigated and to determine which one or combination would be able to control end devices more efficiently than a time-based model.

- Implementing a smart house system in an established building is expensive. An affordable communication method was required without the need for any building alterations.

- Evaluating the algorithm based on three months' recorded data as the prediction model had only three months' knowledge for a step-through each day predicting the following day's conditions.

- The smart home controller used the investment in current security systems sensors.

  All existing smart home controllers on the market are integrated with other systems, but the advantage of this innovation is that current motion detection sensors, such as door contacts could now be used to record human movement as an input for the algorithm to predict the household occupancy.

1.4 *Limitations*

The research was only implemented and tested at one household with the assumption it could be transferred to others and is seen as a norm.

1.5 *Layout of the dissertation*

The organisation of the thesis is as follows:

- Chapter 1

  This chapter presents the background and aim of the study with an outline of the research project.

- Chapter 2

  Different models such as neurological influences specific to the Circadian rhythm, Fuzzy logics, Clustering, Bayes theorem, Hidden Markov model and previous research done in this field, are discussed to find a solution for this hypothesis.

- Chapter 3

  An overview of the solution is discussed. The microcontroller software for the sensors or Internet of Things (IoT) is developed. An algorithm is developed for the computer-developed software to communicate with the IoT.

- Chapter 4

  Design software to analyse and simplify recorded data to formulate a prediction method, is set out

  The data were analysed as follows:

  1. Fuzzified temperatures for the geyser environments to establish water usage;
  2. Calculated heat loss;
  3. Calculated energy inserted into the system, and
  4. Simplified recorded data via partition clustering into three Medoids.

- Chapter 5

  A detailed discussion is given of the developed prediction model. Simulator software was developed and tested as follows:

  1. Developed and compared three different HMM models for Rainy: None, Rainy: Low and Rainy: High for pool and geyser solar heating against one HMM model.
  2. The Viterbi algorithm calculated pool and geyser solar heating prediction
  3. Baum-Welch algorithm training the HMM models
  4. Trained geyser usage based on Fuzzy linguistic terms: “None”, “Small”, “Medium” and “Large”.

  The IoT sensors switched from learning mode to simulating the device studied. Results are compared with simulations.

- Chapter 6

  This chapter comprises the conclusion of the thesis.

Chapter 2

2 Artificial intelligence prediction models and major energy consumers

Internet of Things (IoT) is nowadays being mentioned in almost any technology related discussion. IoT as not only the implementation of internet controlled sensors, but also the intelligent algorithms that controls the sensors more economical than humans. Algorithms, known as Artificial Intelligence (AI), learn environmental and human behaviours.

The study of circadian rhythm is the behavioural changes in a human, responding to the solar cycles of light and darkness [2, p.1]. Human activities are synchronised around the human biological clock and require light to reset the clock each morning and keep the human in sync with the external world [3, p.1]. Medical research has proven that the master circadian clock in humans is the suprachiasmatic nuclei (SCN) [4, p.1]. This research will proof that a predictive model can be developed based on the principle that human behaviours are time orientated.

In this chapter models that were identified and have a prediction ability to be used in the research will be further investigated to improve smart house controllers to predict energy consumption. Those investigated are Fuzzy logics, Clustering, Bayes Theorem and Hidden Markov Model (HMM).

The major energy consumers identified, the geyser element and pool motor are investigated as they are in operation the whole day without knowledge of such.

Different communication techniques that can be used conveying this type of information in an existing property are also discussed. Other major energy users such as stoves, irons and diswashers etc are excluded from research as they are only enabled when needed.

2.1 *Fuzzy Logics*

L. A. Zadeh introduced fuzzy logic in 1965 in his paper "Fuzzy Sets". It was designed to represent uncertainty and formalised tools for dealing with the uncertainty. Fuzzy logic is the logic behind approximate reasoning instead of exact reasoning [5, p.1].

Traditional Boolean logic can only define a state as true or false. Fuzzy logics overcome the shortfalls of Boolean logics. With fuzzy logics an entity has a percentage probability to be part of a member. The outcome of the set is still true or false [6, p.1].

2.1.1 **Fuzzy sets concepts**

The fundamental concepts of a set can be defined where S represents a universal set that includes all possible elements of a problem.

- If set A is in the universal set, it can be written as

$$A \subseteq S. \tag{2.1}$$

- If set A is not part of the universal set, it is written as

$$A \nsubseteq S \tag{2.2}$$

- If an element x is part of set A it is written as

$$x \in A \tag{2.3}$$

- If element x is not part of set A it is written as

$$x \notin A \tag{2.4}$$

- A set without any elements is called an empty set and can be written as [6, pp.1-2], [7, pp.1-2]

$$A = \varnothing \tag{2.5}$$

The relationship between sets is defined where:

- Every element x a member is of set A and a member of set B, then A is a subset B when $A \subseteq B$ and $A \neq B$. It can be written as

$$A \subset B = \{ x \mid x \in A, x \in B\} \tag{2.6}$$

- Both sets are identical when $A \subseteq B$ and $B \subseteq A$ and can be written as [5, p.2, 7, p.2]

$$A = B = \{ x \mid x \in A, x \in B\} \tag{2.7}$$

2.1.2 Membership functions

In Fuzzy logics a value can be fuzzified to belong to multiple memberships functions. If an object's temperature is 40 degrees Celsius, it can be 0.2 Cold, 0.8 Hot as shown in Figure 2.1.

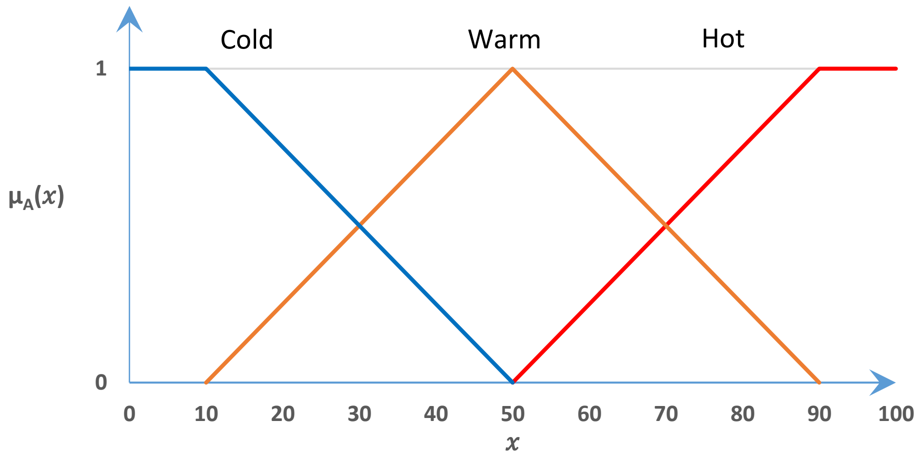


Figure 2.1 Temperature Membership Function

Fuzzy set membership functions are defined in many ways if it follows the rules of a fuzzy set. A membership function can define if an element is part of a set or not [7, p.2]. The membership function μ_A for set A with element x can be written as

$$\mu_A(x) = 1 \text{ if } x \in A \tag{2.8}$$

$$\mu_A(x) = 0 \text{ if } x \notin A. \tag{2.9}$$

A Crisp set membership function μ_A maps whole members in the universal set that includes 0 and 1 and is written as {0, 1}. A fuzzy set maps all real numbers from 0 to 1 and is written as [0, 1] [7, p.8].

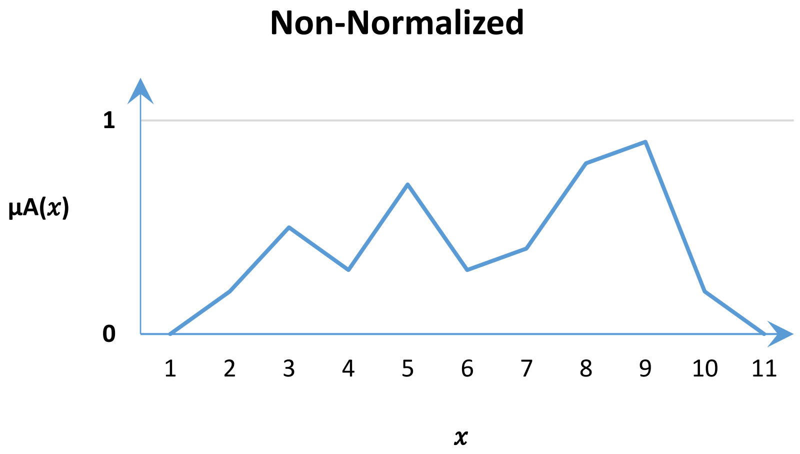


Figure 2.2 Fuzzy set type – Non-Normalised

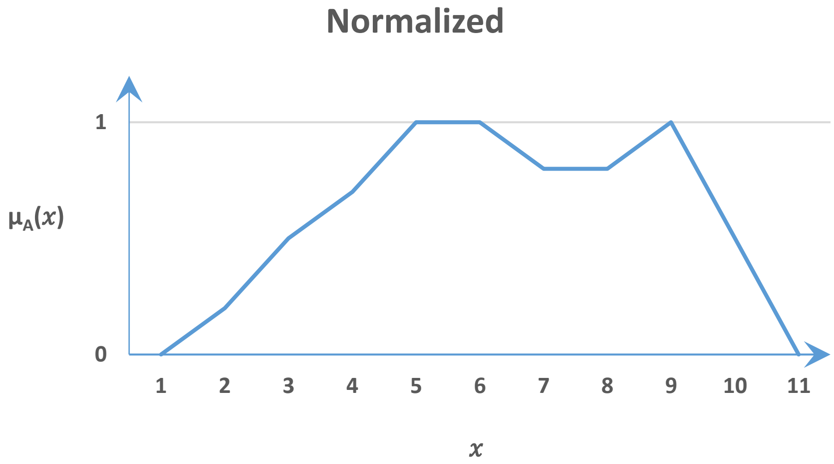


Figure 2.3 Fuzzy set types - Normalised

A fuzzy set that never reaches one is called a non-normalized fuzzy set and one that reaches one a normalized fuzzy set as shown in Figure 2.2 and Figure 2.3 [7, p.15].

The shape of a membership function is defined mathematically by various shapes and is dependent on the purpose. Zadeh classified the membership functions into two groups: Linear and Curved [5, p.5]. A membership function $\mu_A(x)$ defines the degree in which x verifies in the fuzzy set A [5, p.4].

There are a many membership functions, but the more common ones will be discussed below. The following sets are known as convex fuzzy sets, except for the extended trapezium if it is not semantic.

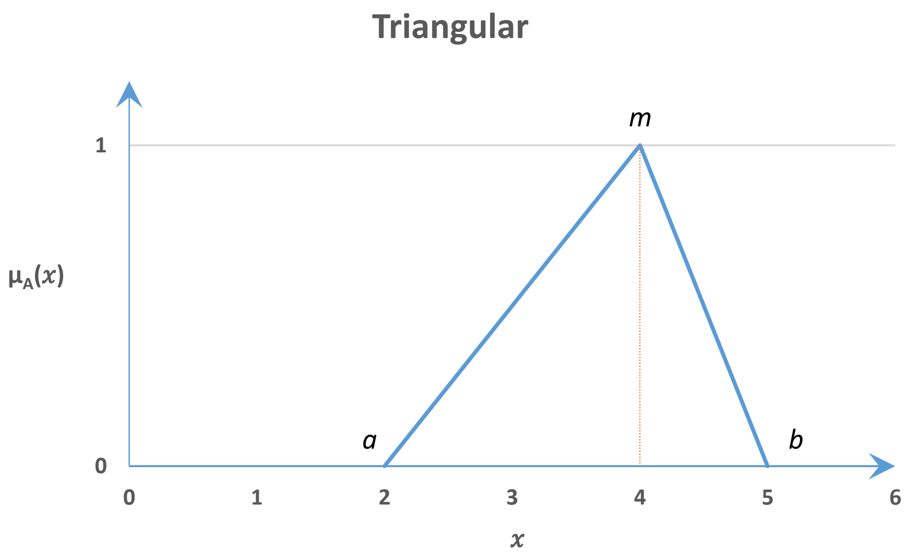


Figure 2.4 Triangular membership function

A Triangular function is defined by its lower limit a, its upper limit b and peak value at m where $a < m < b$ as shown in Figure 2.4 where [5, p.6]

$$\mu_A(x) = \begin{cases} 0 & \text{if } x \leq a \\ (x\text{-}a)/(m\text{-}a) & \text{if } x \in (a,m) \\ (b\text{-}x)/b\text{-}m) & \text{if } x \in (m,b) \\ 1 & \text{if } x \geq b \end{cases} \tag{2.10}$$

The Trapezoid membership function is defined by its limits *a* and *b* and the limits of the nucleus *m* and *n* as shown in Figure 2.5 where [5, p.7]

$$\mu_A(x) = \begin{cases} 0 & \text{if } x \leq a \\ (x\text{-}a)/(m\text{-}a) & \text{if } x \in (a,m) \\ (b\text{-}x)/b\text{-}m) & \text{if } x \in (m,b) \\ 1 & \text{if } x \geq b \end{cases} \tag{2.11}$$

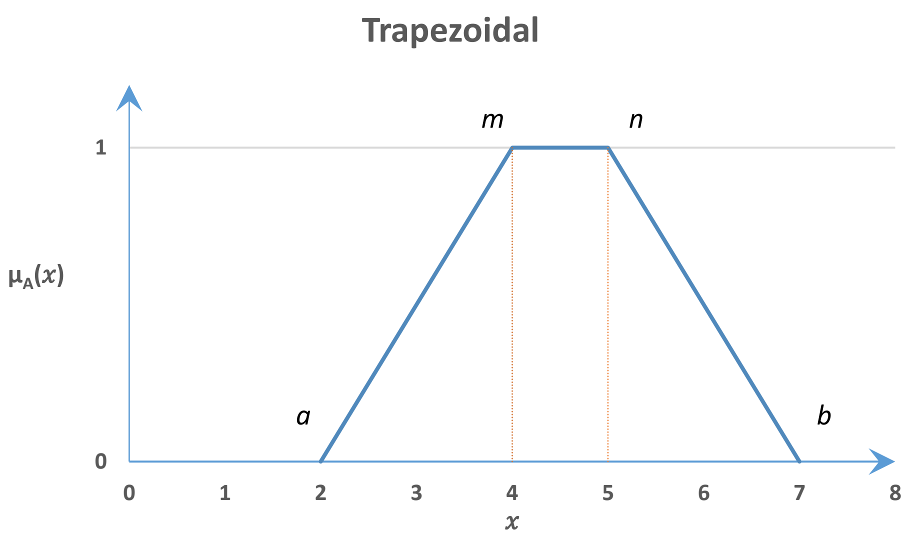


Figure 2.5 *Trapezoidal membership function*

The R and L membership function are special trapezoidal functions. The R function is shown in Figure 2.6 and the L function in Figure 2.7 where [5, p.7]

$$\mu_A(x) = \begin{cases} 0 & \text{if } x \leq a \\ (b-x)/(b-a) & \text{if } a < x < b \\ 1 & \text{if } x \geq b \end{cases} \qquad (2.12)$$

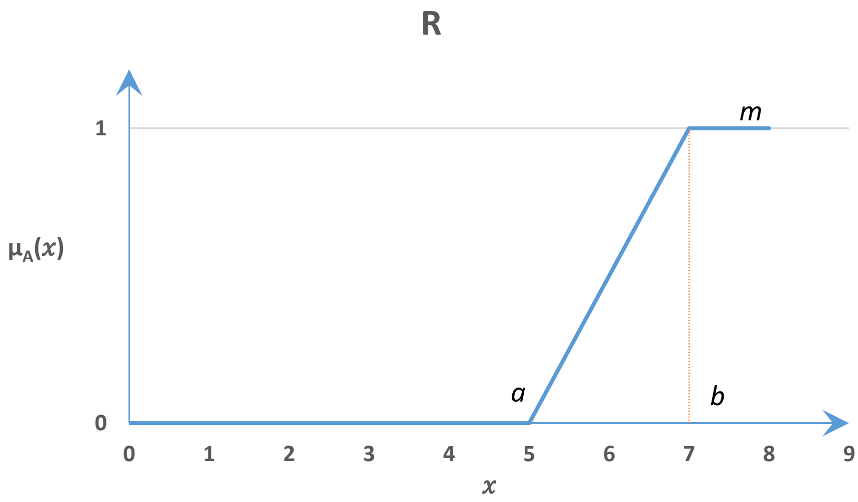


Figure 2.6 R membership function

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \leq a \\ (a-x)/(b-a) & \text{if } a < x < b \\ 0 & \text{if } x \geq b \end{cases} \qquad (2.13)$$

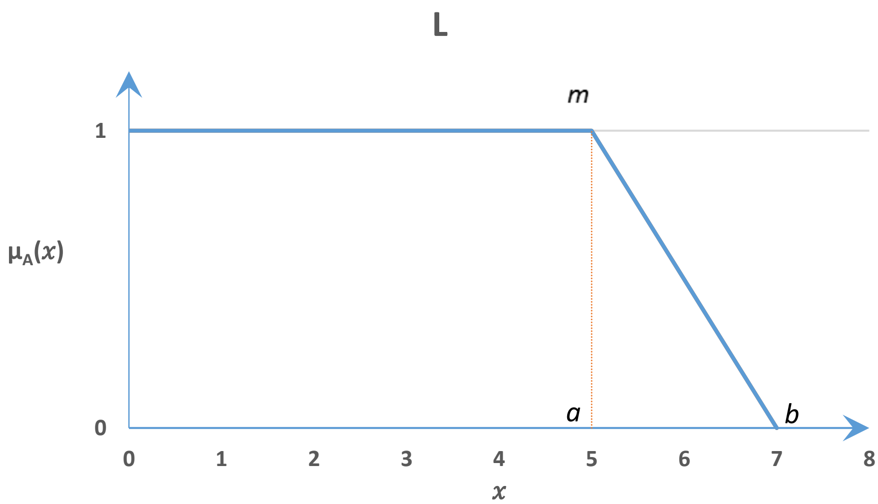


Figure 2.7 L membership function

The Gaussian function is a typical Gauss Bell graph where *m* is the mid value and $k > 0$. The bigger *k*, the narrower the bell graph as shown in Figure 2.8 [5, p.9].

$$\mu_A(x) = e^{-k(x-m)^2} \tag{2.14}$$

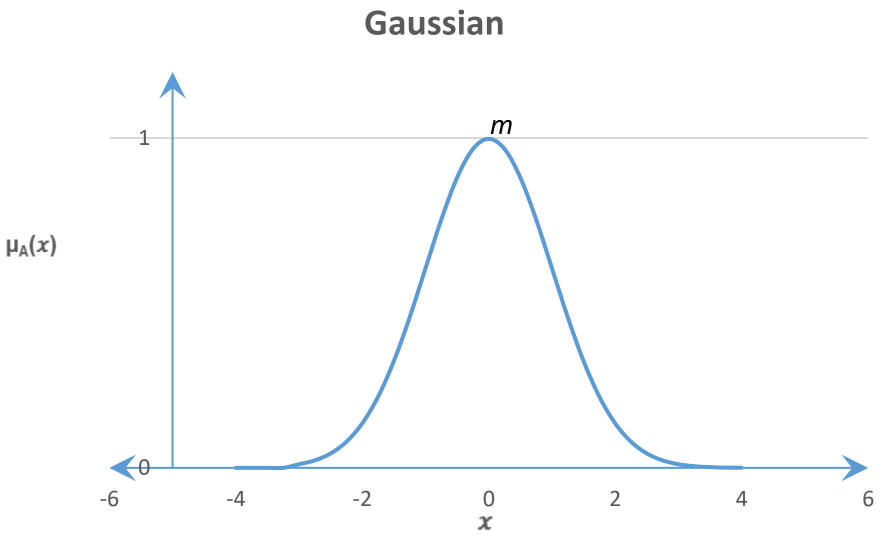


Figure 2.8 Gaussian membership function

2.1.3 Fuzzy sets operations

Age can be defined in a universal set X where:

X = {5,15,25,35,45,55,65,75,85}

Fuzzy sets "Young" and "Adult" is defined in the universal set X. The possibility of each element of x in both sets is shown in Table 2.1 [7, p.14]. This will be used to explain the following fuzzy set operations.

Table 2.1 Age Universal Set [7, p.14].

Age	Young	Adult
5	0	0
15	0.2	0.1
25	1	0.9
35	0.8	1
45	0.4	1
55	0.1	1
65	0	1
75	0	1
85	0	1

2.1.3.1 Complement

There are several types of operation, but Zadeh's version [5, p.19] is that the membership degree of the complement is illustrated in Figure 2.1 and is calculated as follows [6, p.3].

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x),\ \forall x \in X \qquad (2.15)$$

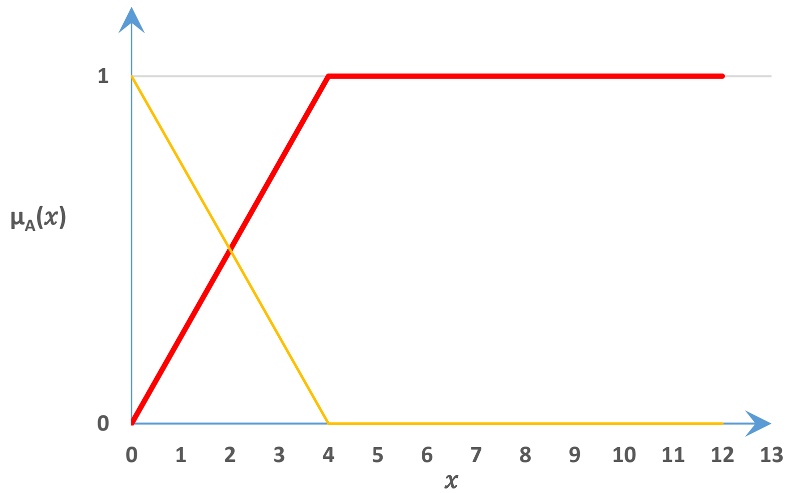


Figure 2.9 Complement of fuzzy set

An example for Complement for "adults" in Table 2.1 is

$\bar{A}$ = {(5, 1),(15, 0.9),(25, 0.1)} [7, p.21].

2.1.3.2 Union

The standard union function between two membership values is the greater value between them and is defined as [6, p.3].

$$\mu_{A\cup B}(x) = \mathrm{Max}[\mu_A(x), \mu_b(x)],\ \forall\ x \in X \qquad (2.16)$$

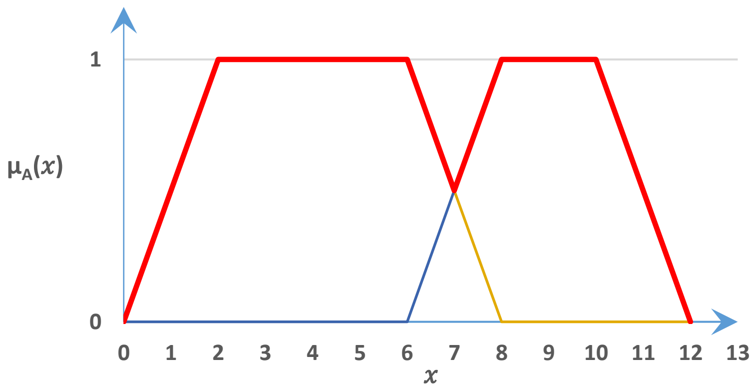
or

$\mu_{A\cup B}(x) = [\mu_A(x) \vee \mu_b(x)],\ \forall\ x \in X.$

An example of the union for "adults" and "young" in Table 2.1 is [7, p.21]

"young" ∪ "adults" = {(15, 0.2), (25, 1), (35, 1), (45, 1), (55, 1), (65, 1), (75, 1), (85, 1)}.

The union of two convex fuzzy set is not always convex as shown in Figure 2.10.



.

Figure 2.10 Union between fuzzy sets

The s-norms or t-conorms generalise the "OR" operator and must satisfy five axioms, which are: [7, p.32].

- Axiom U1 — U(0, 0) = 0 , U(0, 1) = 1, U(1, 0) = 1, U (1, 1) = 1 — Boundary
- Axiom U2 — U(a, b) = U(b, a) — Commutativity
- Axiom U3 — if a $\le a'$ and $b \le b'$ then $U(a, b) \le U(a', b')$ — Monotonic

- Axiom U4 $\quad U(U(a, b), c) = U(a, U(b, c))$ $\quad$ Associativity
- Axiom U5 $\quad U(a, a) = a$ $\quad$ Idempotency

Frequently used fuzzy union operators are [5, p.21, 7, pp.34-35]:

- Algebraic or Probabilistic sum [$A \hat{+} B$]

$$\mu_{A \hat{+} B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x).\ \mu_B(x) \tag{2.17}$$

- Bounded sum or Bold union [$A \oplus B$]

$$\mu_{A \oplus B}(x) = \mathrm{Min}\ [1, \mu_A(x) + \mu_B(x)] \tag{2.18}$$

- Drastic sum [$A \dot{\cup} B$]

$$\mu_{A \dot{\cup} B}(x) = \begin{cases} \mu_A(x), \text{ when } \mu_b(x) = 0 \\ \mu_B(x), \text{ when } \mu_A(x) = 0 \\ 1, \text{ otherwise.} \end{cases} \tag{2.19}$$

- Hamacher's sum [$A \cup B$]

$$\mu_{A \cup B}(x) = \frac{\mu_A(x) + \mu_B(x) - (2 - \gamma)\ \mu_A(x)\ \mu_B(x)}{1 - (1 - \gamma)\ \mu_A(x)\ \mu_B(x)} \quad ; \gamma \geq 0 \tag{2.20}$$

2.1.3.3 Intersection

The intersection between two membership values is the smallest value between them and is defined as [6, p.3]

$$\mu_{A \cap B}(x) = \mathrm{Min}[\mu_A(x), \mu_b(x)], \ \forall\ x \in X \tag{2.21}$$

or

$$\mu_{A \cap B}(x) = [\mu_A(x) \wedge \mu_b(x)], \ \forall\ x \in X.$$

An intersection example for “adults” and “young” in Table 2.1 is [7, p.21]:

“young” $\cap$ “adults” = {(15,0.1),(25,0.9),(35,0.8),(45,0.4),(55,0.1)}.

The intersection of two convex fuzzy sets is always convex as shown in Figure 2.11.

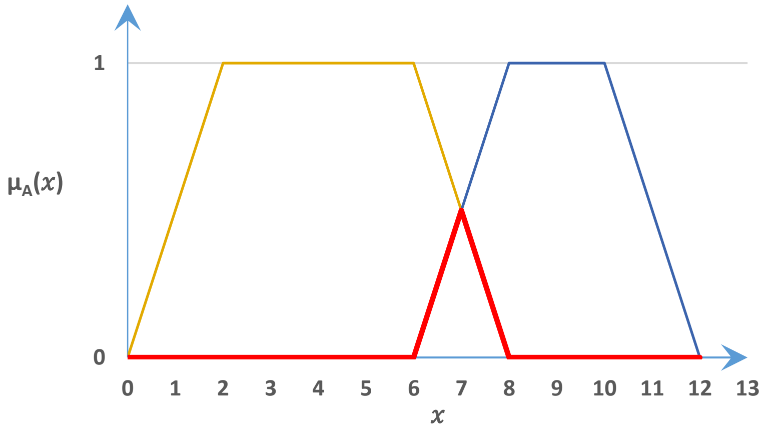


Figure 2.11 Intersection between fuzzy sets

The t-norms generalize the “AND” operator and must satisfy five axioms [7, p.35].

• Axiom I1	I(0, 0) = 0, I(0, 1) = 0, I(1, 0) = 0, I(1, 1) = 1	Boundary
• Axiom I2	I(a, b) = I(b, a)	Commutativity
• Axiom I3	if $a \leq a'$ and $b \leq b'$ then U(a, b) $\leq$ U(a', b')	Monotonic
• Axiom I4	I(I(a, b), c) = I(a, U(b, c))	Associativity
• Axiom I5	I(a, a) = a	Idempotency

Frequently used fuzzy intersection operators are [5, p.20, 7, p.38]:

- Algebraic or Probabilistic product [$A \bullet B$]

$$\mu_{A \bullet B}(x) = \mu_A(x).\ \mu_B(x) \tag{2.22}$$

- Bounded product or Bold intersection [$A \odot B$]

$$\mu_{A \odot B}(x) = \text{Max}\ [0, \mu_A(x) + \mu_B(x) - 1] \tag{2.23}$$

- Drastic product $[A \odot B]$

$$\mu_{A \odot B}(x) = \begin{cases} \mu_A(x), \text{ when } \mu_b(x) = 1 \\ \mu_B(x), \text{ when } \mu_A(x) = 1 \\ 0, \text{ otherwise.} \end{cases} \tag{2.24}$$

- Hamacher's intersection $[A \cap B]$

If A and B is fuzzy numbers and $\gamma \geq 0$ a real number then $H\gamma$-sum is defined

$$\mu_{A \cap B}(x) = \frac{\mu_A(x) + \mu_B(x)}{\gamma + (1 - \gamma)\,(\mu_A(x) + \mu_B(x)) - \mu_A(x)\,\mu_B(x)} \tag{2.25}$$

2.1.3.4 Properties of standard fuzzy set operations

The following equations represent the different Fuzzy functions [6, p.3], [7, pp.5-6]:

1. Involution $\bar{\bar{A}} = A$ (2.26)

2. Commutativity
$A \cup B = B \cup A$ (2.27)
$A \cap B = B \cap A$ (2.28)

- Associativity
$(A \cup B) \cup C = A \cup (B \cup C)$ (2.29)
$(A \cap B) \cap C = A \cap (B \cap C)$ (2.30)

- Distributive
$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ (2.31)
$(A \cup B) \cap C = (A \cup B) \cap (A \cup C)$ (2.32)

- Idempotency
$A \cup A = A$ (2.33)
$A \cap A = A$ (2.34)

- Absorption
$A \cup (A \cap B) = A$ (2.35)
$A \cap (A \cup B) = A$ (2.36)
$A \cup (\bar{A} \cap B) = A \cup B$ (2.37)
$A \cap (\bar{A} \cup B) = A \cap B$ (2.38)
$A \cup S = S$ (2.39)
$A \cap \Phi = \Phi$ (2.40)

- Identity
$A \cap S = A$ (2.41)
$A \cup \Phi = A$ (2.42)

- Contradiction
$A \cup \bar{A} = S$ (2.43)

- Exclude middle $A \cap \bar{A} = \Phi$ (2.44)
- De Morgan $\overline{A \cap B} = \bar{A} \cap \bar{B}$ (2.45)

$\overline{A \cup B} = \bar{A} \cup \bar{B}$ (2.46)

2.1.4 **Linguistic**

Fuzzy logics goal is to give the basis for approximate reasoning using imprecise propositions based on fuzzy set theory in antinomy with classical precise reasoning where the output is always “True” or “False” [6, pp.66-67].

Example 2.1

Precise: Everyone 40 years old and older are old.

Imprecise: Everyone between 40 and 70 years old is old, but everyone older than 71 years is very old.

Approximate reasoning has been frequently used in everybody’s daily life.

Fuzzy logic allows imprecise linguistic terms like [6, p.67]:

- Fuzzy predicates

  Words such as *old*, *rare*, *severe*, *expensive*, *high*, *fast* are used.

  If it is said: “w is young”, then the term “young” is a fuzzy term and “young(w)” is a fuzzy set. The membership function will be $\mu_{young(w)}$ [7, p.205].

- Fuzzy modifier

  Words such as *very, likely, almost impossible, extremely unlikely* are used. If the variable "age" in linguistic terms is defined as "young" and "very young", it falls in a universal set U. When "very young" is presented as a membership function, it can be the square of $\mu_{young(u)}$ where

  $\mu_{very\ young(u)} = (\mu_{young(u)})^2$ [7, pp.205-206].

- Fuzzy truth values

  Baldwin defined the fuzzy truth qualifier as [7, p.207]:

$$\mu_{true}(v) = (v) \qquad v \in [0, 1] \qquad (2.47)$$

$$\mu_{very\ true}(v) = (\mu_{true}(v))^2 \qquad v \in [0, 1] \qquad (2.48)$$

$$\mu_{fairly\ true}(v) = (\mu_{true}(v))^{1/2} \qquad v \in [0, 1] \qquad (2.49)$$

$$\mu_{false}(v) = (1 - \mu_{true}(v)) \qquad v \in [0, 1] \qquad (2.50)$$

$$\mu_{very\ false}(v) = (\mu_{false}(v))^2 \qquad v \in [0, 1] \qquad (2.51)$$

$$\mu_{fairly\ false}(v) = (\mu_{false}(v))^{1/2} \qquad v \in [0, 1] \qquad (2.52)$$

$$\mu_{absolutely\ true}(v) = 1 \qquad v = 1 \qquad (2.53)$$

$$\mu_{absolutely\ false}(v) = 0 \qquad v = 0 \qquad (2.54)$$

2.1.5 Fuzzy logic rule base

If Lukasiewicz's definition is followed, the fuzzy operators for $a, b \in [0,1]$ are [6, p.76, 7, p.202]:

Negation	$\mu_A(\bar{a})$	$= \mu_A(1 - a);$	(2.55)
Conjunction	$\mu_A(a \wedge b)$	$= \text{Min}\ \{\mu_A(a), \mu_A(b)\};$	(2.56)
Disjunction	$\mu_A(a \vee b)$	$= \text{Max}\ \{\mu_A(a), \mu_A(b)\};$	(2.57)
Implication	$\mu_A(a \Rightarrow b)$	$= \text{Min}\ \{1, 1 + \mu_A(b) - \mu_A(a)\};$	(2.58)
Equivalence	$\mu_A(a \Leftrightarrow b)$	$= 1 - \lvert \mu_A(a) - \mu_A(b) \rvert.$	(2.59)

A fuzzy rule generally assumes the form, R: if x is A, then y is B, where x and y are linguistic variables representing the process state [7, pp. 220, 226].

An example, when a fuzzy rule is converted from a classical two-valued logic too terms of a membership is [6, p. 68]:

$(a \wedge (a \Rightarrow b)) \Rightarrow b$

If $\mu(a) > 0$ AND $\mu(a \Rightarrow b) = \min\{1, 1+\mu(b) - \mu(a)\} > 0$ then $\mu(b) > 0$.

2.1.6 **Fuzzy Logic Controller**

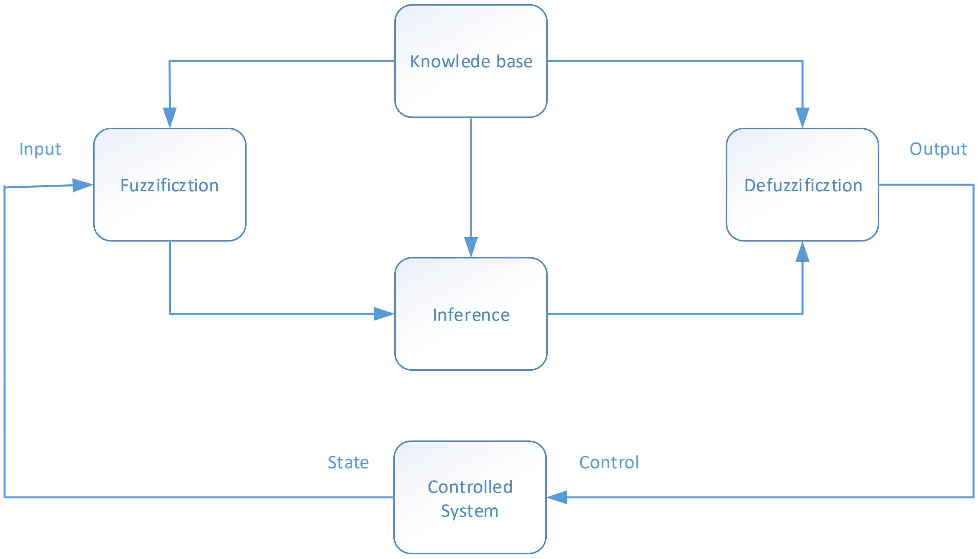


Figure 2.12 Basic block diagram of Fuzzy Logic Controller

Fuzzy Logic Controllers (FLC) can be broken into three components as illustrated in Figure 2.12 [8, p. 6].

2.1.6.1 Fuzzification

This module converts the physical input values into a normalized fuzzy subset and an associated membership function, which involves allocating suitable linguistic values and needs to take random input noise into account. The main purpose of this step is to make the input signal compatible with the fuzzy control rule base. The fuzzy set and the membership function need to be selected by the designer for each application [6, pp.148-149, 7, pp.254-255].

2.1.6.2 Knowledge base and inference

It typically contains some type of database which is used for control rules and data manipulation. The inference process simulates human decision-making based on linguistic rules to generate a fuzzy response. Mamdani, Larsen, Tsukamoto and TSK methods are used in the decision-making logic [7, pp.255-256].

The rule base depends on the developer expert knowledge about the specific system. The FLC needs to determine the choices in input and output variables and the IF-THEN rules [6, pp.148-149].

2.1.6.3 Defuzzification

This process is the interface between the rule base and the application’s physical output control. It converts all the output membership function back to crisp terms using several common defuzzification formulas. [6, p.149,158]. Based on the properties of the application the best suited defuzzification strategy needs to be selected. There are many strategies but the most common strategies are [7, p.269]:

- **Mean of Maximum**

Mean of Maximum (MoM) generates a value that represents the mean value of all control actions, whose membership function reach the maximum, as illustrated in Figure 2.13 where

$$x_0 = \sum_{j=1}^{k} \frac{x_j}{k} \tag{2.60}$$

x_j membership functions reach maximum

k number of such actions [7, p.269].

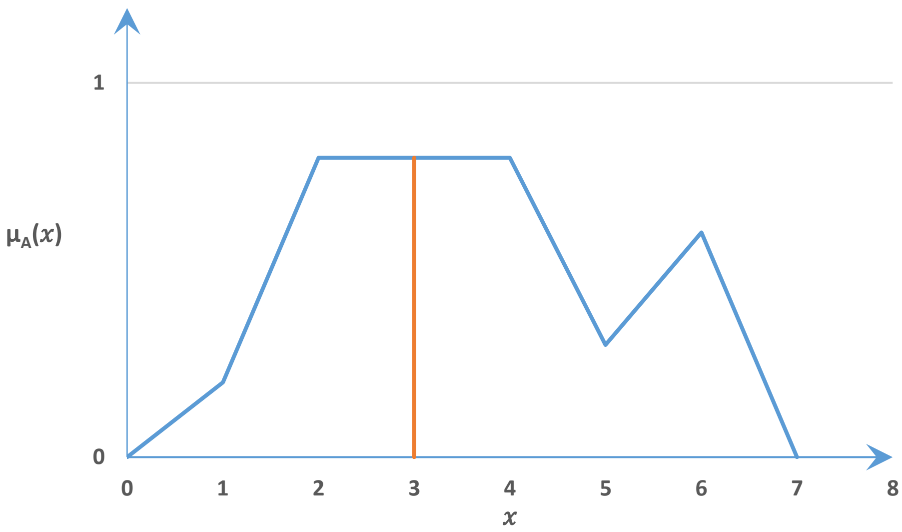


Figure 2.13 *Defuzzification Mean of Maximum*

- **Centre of Area**

This method is also called Centroid or Centre of Gravity (CoG) and is the most prevalent method used for defuzzification, but the calculations can become complex making it not suitable for fast inference cycles. MoM is illustrated in Figure 2.14 where

$$x_0 = \frac{\sum_{j=1}^{n} u_c(x_j) \cdot x_j}{\sum_{j=1}^{n} u_c(x_j)} \tag{2.61}$$

n number of quantization levels of the output,

c fuzzy set,

x output dimension [7, p.270].

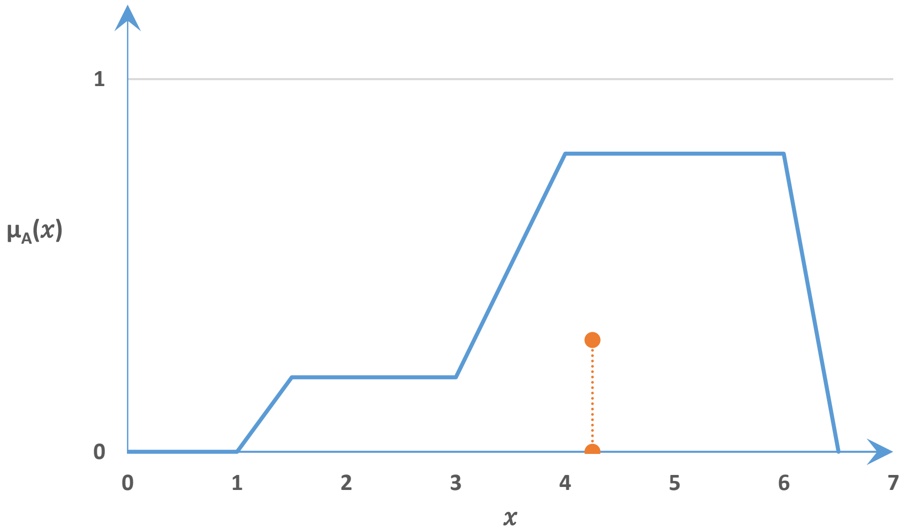


Figure 2.14 Centre of Gravity

- **Other**

  There are many more defuzzification methods available like, adaptive integration, bisector of area, first of maximum and last of maximum to name only a few.

2.1.6.4 Design procedure

When an FLC is designed the following decisions need to be made:

- State and control variables;
- Inference method;

- Fuzzification method;
- Discretization and normalisation;
- Shape of fuzzy set;
- Construction of fuzzy rule base;
- Defuzzification method [7, pp.271-272].

2.2 *Clustering*

Classification is used as a supervised learning method and clustering for unsupervised learning [9, p.321]. In 1998, Farley and Raftery suggested8 dividing clustering methods in two main groups: hierarchical and portioning methods [9, p.330].

2.2.1 Hierarchical Methods

This method is popular in biological, social and behavioural sciences [10, p89]. This method can be subdivided into two methods:

- Agglomerative hierarchical clustering

  Starting each object is in a cluster. Two clusters which are the most similar or smallest distance between clusters are merged together. The process is repeated until every object is merged into a single cluster. The partitioning instance is in a bottom-up fashion [9, p.330].

- Divisive hierarchical clustering

  It is the reverse process of Agglomerative hierarchical clustering. All objects are in one cluster subdividing it into smaller clusters until each leaf cluster contains only one object. The partitioning instance is in a top-down fashion [10, p.57].

The results of the hierarchical method are shown in a dendrogram diagram illustrating how the clusters are merged or split is illustrated in Figure 2.15.

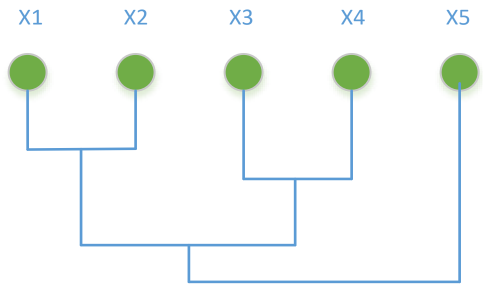


Figure 2.15 *Dendrogram*

Hierarchical methods can further be divided by the method the smallest inter-cluster distance is calculated.

- Single-link clustering

  The distance between two clusters is calculated as the shortest distance from any member in one cluster to any member of the other cluster as illustrated in Figure 2.16, Figure 2.17 and Equation 2.62. A disadvantage of this method is that a few points that form a bridge between two clusters can cause that the two clusters are unified in what is known as the “chaining effect”, making it sensitive to noise [9, p.331].

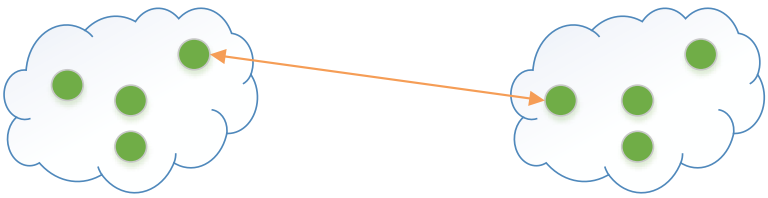


Figure 2.16 *Single-link clustering*

$$d_{\min}(C_i,C_j) = \min_{p\in C_i,\, q\in C_j} d(p,q) \tag{2.62}$$

- Complete-link clustering

  The distance between two clusters is calculated as the longest distance from any member in one cluster to any member of the other cluster as illustrated in Figure 2.17 and Equation 2.63 making it less susceptible to noise [9, p.331].

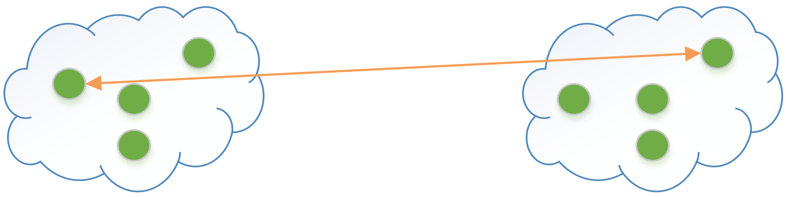


Figure 2.17 Complete-link clustering

$$d_{\min}(C_i,C_j) = \max_{p\in C_i,\, q\in C_j} d(p,q) \tag{2.63}$$

- Average-link clustering

  The distance between two clusters is calculated as the average distance from any member in one cluster to any member of the other cluster as illustrated in Figure 2.18 and Equation 2.64. A disadvantage with this method is that it may cause elongated clusters to split and for others to merge [9, p.331].

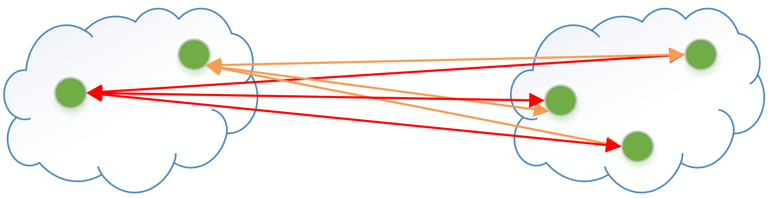


Figure 2.18 Average-link clustering

$$d_{\min}(C_i,C_j) = \operatorname*{avg}_{p\in C_i,\, q\in C_j} d(p,q) \tag{2.64}$$

The main disadvantage with any hierarchical methods is:

- Does not scale well;
- There is no backtracking or undo what was done previously [9, p.332].

2.2.2 Partitioning Methods

Non-hierarchical clustering method is called partitional clustering and is popular in engineering because of its efficient representation and compression of large databases. The method can be defined as follows: “Given *n* patterns in a *d*-dimensional metric space, determine a partition of the patterns into *K* clusters, such that the patterns in the cluster are more like each other than to the patterns in different clusters” [10, pp.89-90]. The most common methods used is the K-means and K-medoids algorithm.

- K-Means

  Figure 2.19 illustrates the flow chart for the K-means algorithm. In the first step the initial centroids are randomly chosen. Most of the convergence occurs in the first few steps and sometimes the centroid will always change and the check condition needs to be weaker like only 1% if the points change clusters [11, pp.497-498]. When random, initial centroids are used, it can produce different solutions. One solution is to use sum of the squared error (SSE) as shown in Equation 2.65. It calculates the error of each data point to the nearest centroid and then calculates the sum of all the errors.

$$\mathrm{SSE} = \sum_{j=1}^{K} \sum_{x \in C_j} (x - m_j)^2 \tag{2.65}$$

x	An object
C_j	The j^{th} cluster
K	The number of clusters
m_j	The centroid of cluster C_j [11, p.500].

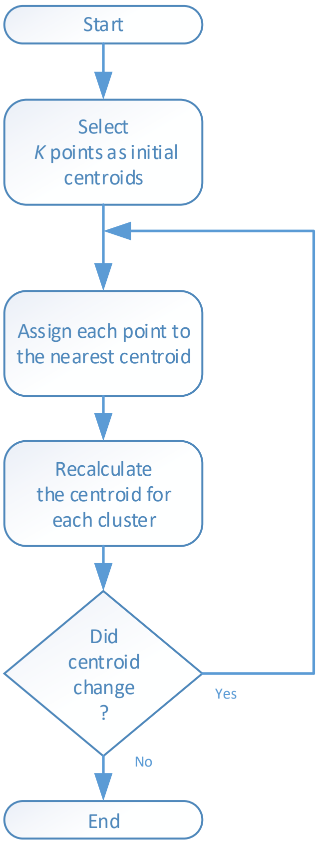


Figure 2.19 K-means flow chart

The centroids are moved around and the point with the smallest SSE is selected as the initial centroid point, but this strategy is computationally infeasible and impractical. [11, p.513].

The centroids that are given the smallest SSE can be calculated by Equation 2.60 where [11, p.500]

$$C_j = \sum_{x \in C_j} x \tag{2.66}$$

x An object

C_j The j^{th} cluster

m_j The centroid of cluster Cj [11, p.500].

Bisecting K-means is a mathematical solution to overcome the sensitivity problem the basic K-mean method has with selecting the initial centroids.

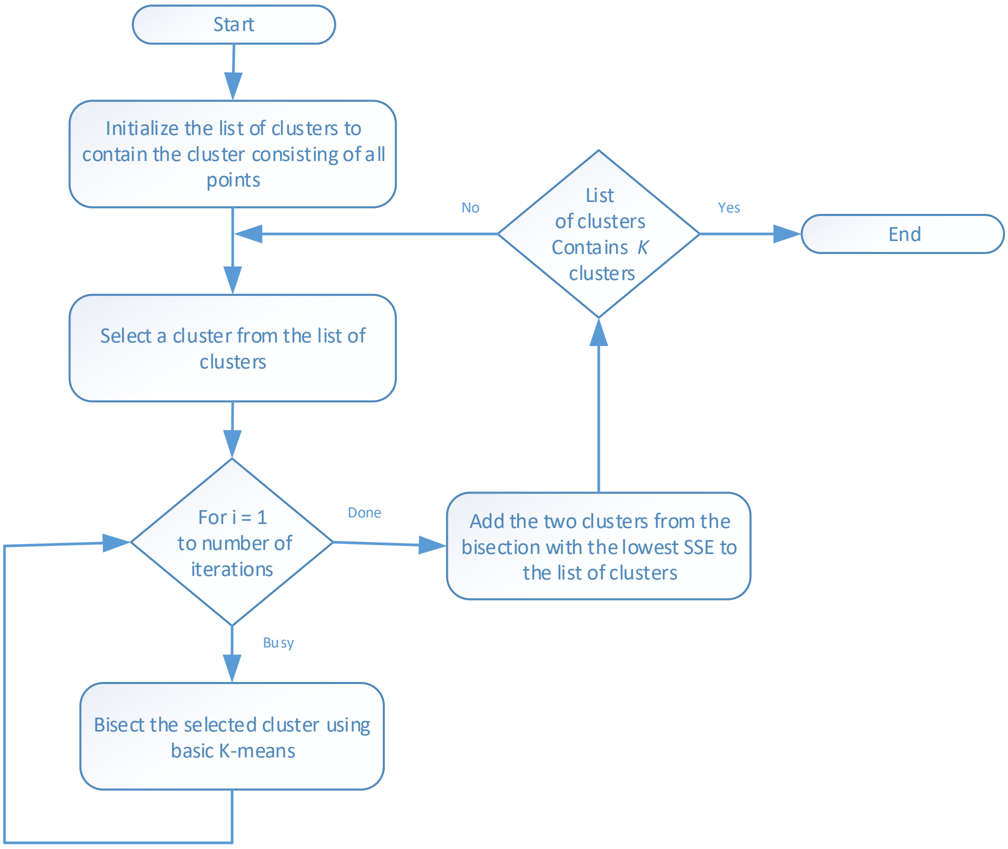


Figure 2.20 Bisecting K-means

The basic flow chart is illustrated in Figure 2.20. The clusters can be split based on size, largest SSE or both. The final set of clusters does not represent a clustering with the lowest total SSE. This is simple and efficient even if multiple runs are performed [11, pp.509-510]. Two problems with K-means algorithm is that empty clusters can occur and sensitive to noisy data and outliers from the use of SSE techniques [9, p.334].

- K-medoids

The main difference between the K-means and K-medoids is that with K-means the centroid almost never corresponds to an actual data point where K-medoids is the most representative point for a group of points making it more robust in in the presence of noise and outliers [12, p.334].

The most common realisation of the K-medoids clustering is the Partition Around Medoids (PAM) algorithm [12, p.455]. For the basic PAM algorithm, each non-medoid is swapped to be the medoid of the cluster and the absolute-error is calculated. If the absolute-error improves, the selective object becomes the new medoid. This process is costly and does not scale well for large data sets. To deal with large data sets, a sampling method called Clustering LARge Application (CLARA) can be used where only one random sample is taken to calculate the best medoids as illustrated in Figure 2.21. This increase the processing time but decrease the possibility to select the best medoids drastically. An algorithm called Clustering LARge Application based upon RANdomised search (CLARANS) is a trade-off between cost and effectiveness. The algorithm selects *n* times a temporally medoid and calculate if the newly selected medoid will improve the absolute-error criterion. If the error improves, the temporal medoid is chosen as the new medoid [12, p.455][13, p1945].

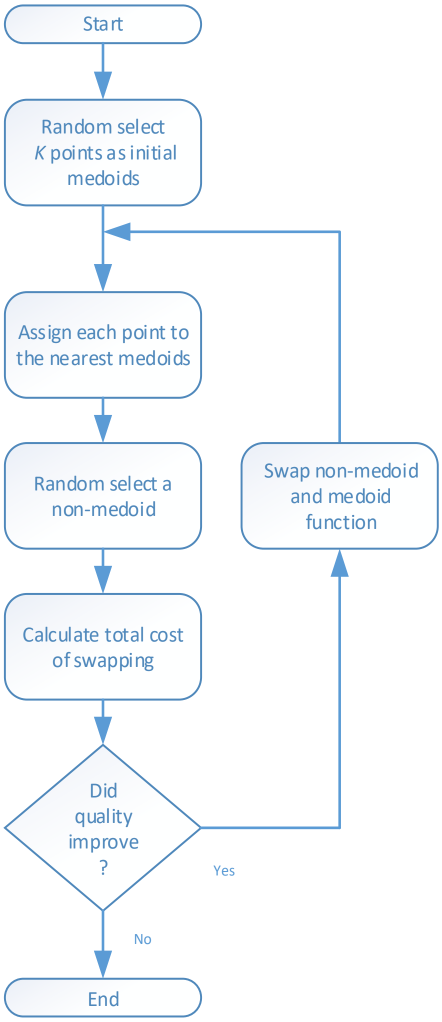


Figure 2.21 K-Medoids

The absolute-error is the sum of the dissimilarities between each object and its corresponding representative object as shown in Equation 2.68. The distance was calculated using the Manhattan instead of Eudiclean method to improve processing time.

$$E = \sum_{j=1}^{K} \sum_{x \in C_j} \text{distance}(x, m_j) \tag{2.68}$$

x	An object
C_j	The j^{th} cluster
K	The number of clusters
m_j	The medoid of cluster C_j [12, p.455].

2.3 *Probability and Bayes Theorem*

Bayes' theorem, named after the 18th century British mathematician Thomas Bayes, is a mathematical equation calculating the probability of an event, based on prior information associated with the event. Probability is a number between and including 0 and 1 indicating a degree of belief in a fact or prediction. The value 0 indicates that the probability is false and the value 1 represents certainty that the prediction will come true. The intermediate values represent the degree of likelihood that an event will occur [14, pp. 1].

Conditional probability is based on some background information and is noted as

$p(A|B)$ where (2.69)

p	Probability,
A	Prediction,
B	Set of conditions [14, p.1].

Conjoint probability is the probability that two events are true and is noted as

$p(A \text{ and } B) = p(A).p(B|A)$ and (2.70)

$p(B \text{ and } A) = p(B).p(A|B)$ (2.71) where

p(A and B) = p(B and A) and p($A|B$) <> p($B|A$)

p(B).p($A|B$) = p(A).p($B|A$) Substitute 2.71 in 2.70

p($A|B$) = [p(A).p($B|A$)] / p(B) [14, pp.2-4]. (2.72)

If the formula is rewritten with data D and model M instead of event A and B Bayes Theorem can be written as:

p($M|D$) = [p(M).p($D|M$)] / p(D) (2.73)

where

p($M|D$) Probability of the model after the date is known called the **Posterior**;

p(M) Probability of the model before the data is known called the **Prior**;

p($D|M$) Probability of the data under the model called the **Likelihoo**d and

p(D) Probability of the data under any model called the **Normalizing Constant** [15, p.7].

2.4 *Hidden Markov Module*

Hidden Markov Module (HMM) is basically a Markov chain with unobservable states. With HMM the state is not directly visible but only the output of the system as another stochastic process linked to the Markov chain (MC) [16, p.1]. The probability of the output only depends on the current state in which the MC is in. HMM is one of the most successful statistically modelling, developed in the 1960’s by Baum and colleagues, being used today in speech recognition, biological research, financial, information technology and engineering application [15, pp.19-20]. HMM has only become popular in the past several years because the theory of HMM was previously only published in mathematical journals which were not generally read by engineers [18, p.258].

2.4.1 **Markov Chain**

Basic knowledge of the MC is required to understand the HMM.

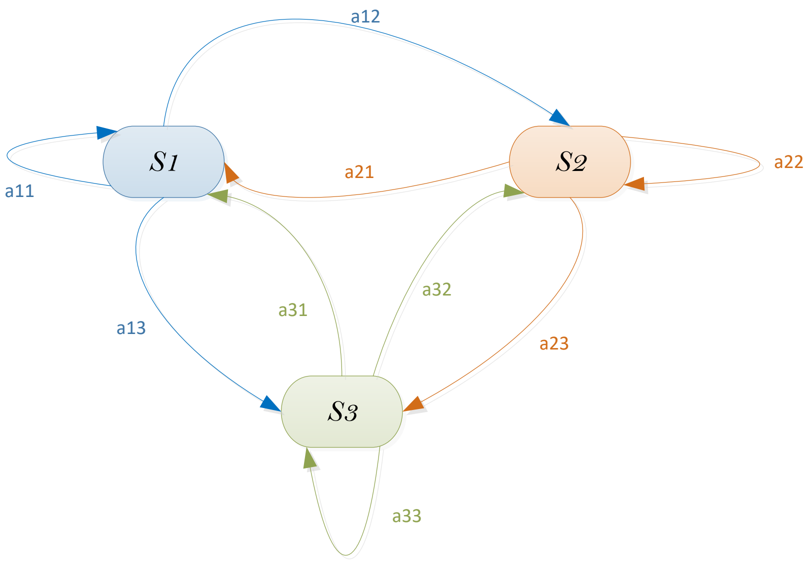


Figure 2.22 Transitional Markov Chain diagram

Figure 2.22 illustrates an MC with N distinct states where N = 3 and the states are S_1, S_2 and S_3. At regular discrete times *T*, the system changes state and the time instance, are denoted as *t*=1, 2, …n, where the actual state at time *t* is q_t. The probabilistic description of the system would require knowledge of the current and the predecessor states and can be written as:

$$P[q_t = S_j \mid q_{t-1} = S_i] \tag{2.74}$$

The transition probabilities a_{ij} can be written as:

$$a_{ij} = P[q_t = S_j \mid q_{t-1} = S_i] \quad \text{where} \tag{2.75}$$

$1 \leq ij \leq N$

$a_{ij} \leq 0$

$\sum_{j=1}^{N} a_{ij} = 1.$

as it obeys standard stochastic constrains [18, p.258].

The initial state probabilities describing the probability of the beginning of the state sequence are written as:

$$\pi_i = P[q_1 = s_i],\ 1 \leq i \leq N \qquad (2.76)$$

The transient diagram in Figure 2.22 can be written as a transition matrix where

$$P = \begin{bmatrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \end{bmatrix}$$

2.4.2 Elements and Mechanisms

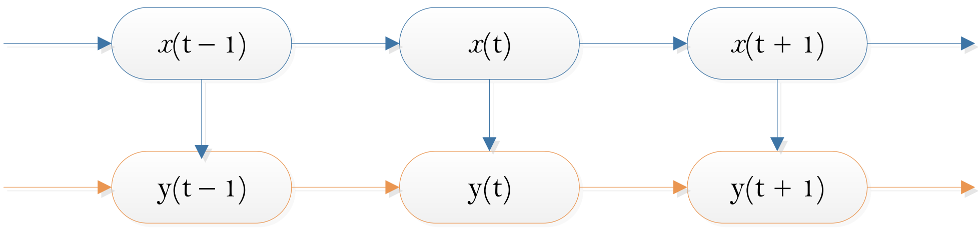


Figure 2.23 Graphical structural representation of a HMM

Figure 2.23 illustrates a graphical representation of the HMM where y(t) is the observable process and x(t) is the hidden chain [16, p.2]. The formal definition of HMM is that it is a doubly stochastic process with an underlying stochastic process that is not observable, but can only be observed through another set of stochastic processes that produce the sequence of observed symbols [19, p.2].

Defining an HMM model the following elements are needed: [15, p.20, 18, pp.7-8].

- N is the number of hidden states in the model where individual states are written as:

  $$S = \{S_1, S_2 \dots S_N\} \qquad (2.77)$$

  and the state at time t as q_t.

  Usually a state can reach any other state as shown in Figure 2.22.

- M is the number of distinct observation symbols per state and is called the alphabet size and corresponds to the physical output of the modelled system. The individual symbols are written as:

  $$V = \{v_1, v_2 \ldots v_M\} \qquad (2.78)$$

- The state transition probability distribution $A = \{a_{ij}\}$ are defined in Equation 2.75.

- The observation or emission symbol probability distribution in state *j*, B = $\{b_j(k)\}$ where:

  $$b_j(k) = P[v_k \text{ at } t \mid q_t = S_j] ,\ 1 \leq j \leq N ,\ 1 \leq k \leq M \qquad (2.79)$$

- The initial state probabilities are defined in equation 2.76.

- The observation sequence can be generated from appropriate values of *N*, *M*, *A*, *B* and π and is written as:

  $$O = O_1, O_2 \ldots O_T \qquad (2.80)$$

- λ represents the complete parameter set of the model where

  $$\lambda = (A, B, \pi) \qquad (2.81)$$

  π represents a path through the model

2.4.3 The Three Basic Problems and their Solutions

There are three basic problems that must be solved for HMM to be useful in real-world applications.

1. **Evaluation**. Given the observation sequence O and an HMM model λ, how can the probability of a given model producing the output sequence $P(O|\lambda)$ efficiently be computed? A solution is the Forward algorithm.
2. **Decoding**. Given the observation sequence O and an HMM model λ, how can a given sequence $Q = q_1, q_2 \ldots q_T$ best explains the observations O? A solution is the Viterbi algorithm.
3. **Learning**. Given the observation sequence O, how to choose a model parameter λ to maximize $P(O|\lambda)$? A solution is the Baum-Welch algorithm [22, p.1385].

2.4.3.1 Forward Algorithm

Calculating $P(O|\lambda)$ are resource intensive because $2T.N^T$ operations are required. The forward algorithm in the HMM context is used to calculate the probability of a state at a certain time, based on historical data. The forward variable is the probability of a partial observation sequence until time t, given the model λ where:

$$\alpha_t(i) = P(O_1, O_2 \ldots O_t, q_t = S_i | \lambda) \qquad (2.82)$$

$\alpha_t(i)$ is the probability of being in state t having observed the i element and can be solved inductively, as follows:

- Initialisation

$$\alpha_t(i) = \pi_i b_i O_1, \quad 1 \leq i \leq N \qquad (2.83)$$

Initialises the forward probabilities as the joint probability of state S_i and initial observation O_1 [23, p.105].

- Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\, a_{ij} \right] b_j O_{t+1}, \qquad (2.84)$$

$1 \leq j \leq N$ and $1 \leq t \leq T\text{-}1$

The induction step is illustrated in Figure 2.24 where state S_j can be reached at time t + 1 from the N possible states S_j, at time t. Since $\alpha_t(i)$ is the probability of the joint event that $O_1\, O_2 \ldots O_t$ are observed and the state time t is S_i, the product $\alpha_t(i)\, a_{ij}$ is the probability of the joint event that $O_1\, O_2 \ldots O_t$ are observed and state S_j is reached at time t + 1 via state S_i at time t. Calculate the sum of the product over all N possible states S_i at time t, results in the probability of S_j at time t + 1 with all the previous partial observations [20, pp. 262-263].

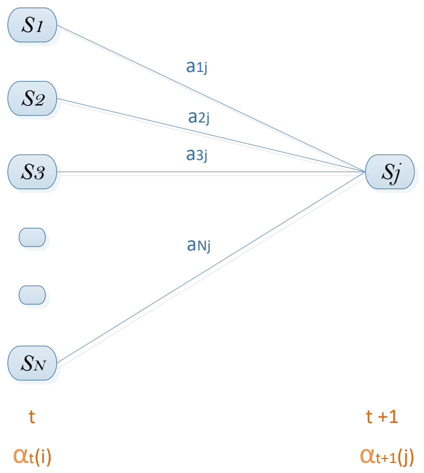


Figure 2.24 Sequence of operations required for computing of forward variable

- Termination

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{2.85}$$

The final step is to calculate the sum of the forward variables $\alpha_T(i)$.

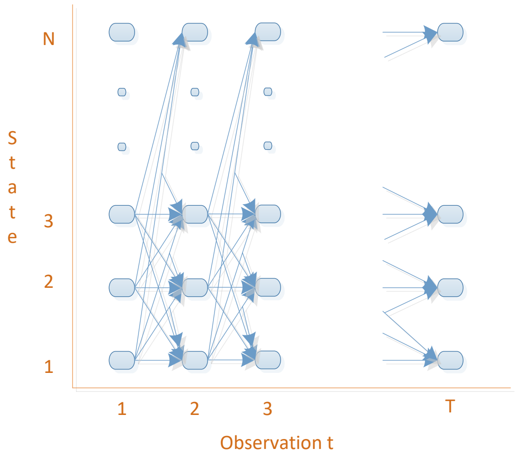


Figure 2.25 Implementation of recursive computation for the forward variable

The forward probability calculations require N^2T calculations and are based upon the trellis structure as shown in Figure 2.25 [20, p.263].

Example 2.2

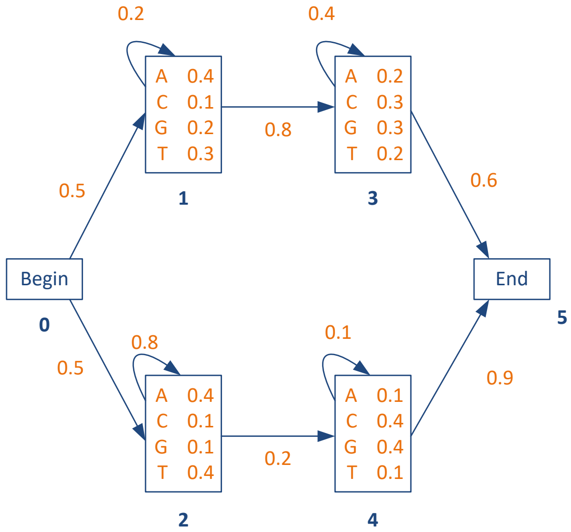


Figure 2.26 Forward algorithm

How likely is sequence "TAGA"?

$\alpha_0(0) = 1$, $\alpha_0(1) = 0$ .. $\alpha_0(5) = 0$

$\alpha_1(1) \quad = b_1(T) \times \alpha_0(0)a_{01} + \alpha_1(0)a_{11}$

$= 0.3 \times (1 \times 0.5 + 0 \times 0.2) = 0.15$

$\alpha_2(1) \quad = 0.4 \times (1 \times 0.5 + 0 \times 0.8) = 0.2$

$\alpha_1(2) \quad = b_1(A) \times \alpha_0(1)a_{01} + \alpha_1(1)a_{11}$

$= 0.4 \times (0 \times 0.5 + 0.15 \times 0.2) = 0.012$

$\alpha_2(2) \quad = b_2(A) \times \alpha_0(1)a_{01} + \alpha_2(1)a_{11}$

$= 0.4 \times (0 \times 0.5 + 0.2 \times 0.8) = 0.064$

……

P(TAGA) $= \alpha_5(4) = \alpha_3(4)a_{35} + \alpha_4(4)a_{45}$

2.4.3.2 Viterbi-Algorithm

Several solutions exit for the second problem, but one of the most popular are the Viterbi-Algorithm. It is a dynamic programming method used to find the best possible hidden state path. $\delta_t(i)$ is the highest probability of a partial state sequence, at time t, which accounts for the first t observations and ends in state S_j where [24, pp.583-584]:

$$\delta_t(i) = \max_{q_1, q_2 .. q_{t-1}} P(q_1 q_2 \ldots q_t = i, O_1, O_2 \ldots O_t | \lambda) \qquad (2.86)$$

and by induction the following equation is derived:

$$\delta_{t+1}(j) = \max [\delta_t(i) \, a_{ij}] \, . \, b_j O_{t+1} \qquad (2.87)$$

The best result for each t and j are stored in an array $\gamma_t(j)$

Finding the best state sequence is computed as follows:

- Initialisation

$$\delta_1(i) = \pi_i \, b_i O_1, \quad 1 \leq i \leq N \qquad (2.88)$$

$$\gamma_1(i) = 0 \qquad (2.89)$$

- Recursion

$$\delta_t(j) = \max \, [\delta_{t-1}(i) \, a_{ij}] \, . \, b_j O_t, \quad 1 \leq i \leq N, \, 2 \leq t \leq T \qquad (2.90)$$

$$\gamma_t(j) = \arg\max \, [\delta_{t-1}(i) \, a_{ij}], \quad 1 \leq i \leq N, \, 2 \leq t \leq T \qquad (2.91)$$

- Termination

$$P^* = \max(\delta_T(i)), \quad 1 \leq i \leq N \qquad (2.92)$$

$$q_T^* = \arg\max(\delta_T(i)), \quad 1 \leq i \leq N \qquad (2.92)$$

- Path backtracking

$$q_T^* = \gamma_{t+1}(q^*_{t+1}), \quad t = T-1, T-2, \ldots, 1 \qquad (2.93)$$

[18, pp.236-237, 21, p.50]

Example 2.3 [25, pp.1-4]

Table 2.2 illustrates a transition probability distribution and Table 2.3 the emission probability distribution. What is the best path if *x y y* is observed?

Table 2.2 *HMM Transition probability distribution*

	Next		
Current	A	B	En d
Start	0.7	0.3	0
A	0.2	0.7	0.1
B	0.7	0.2	0.1

Table 2.3 *HMM Emission probability distribution*

	Word		
State	*S*	*x*	*y*
Start	1	0	0
A	0	0.4	0.6
B	0	0.3	0.7

The trellis is constructed and illustrated in Figure 2.27.

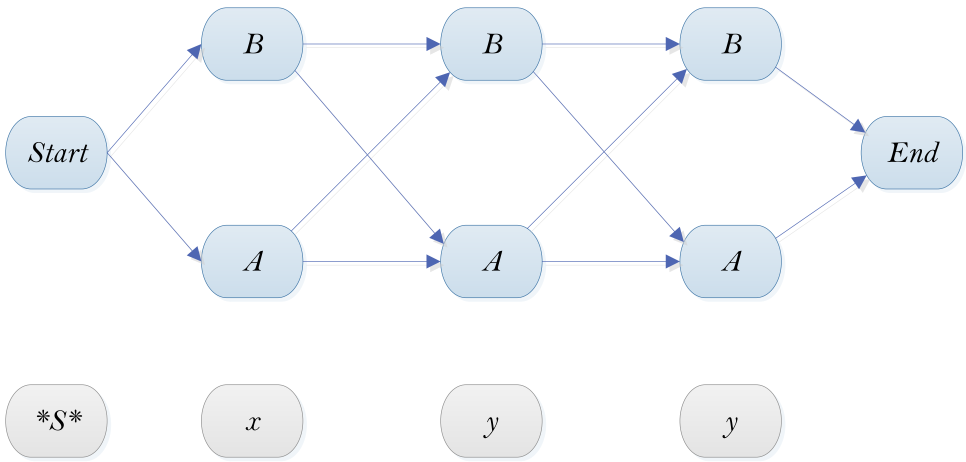


Figure 2.27 *Trellis*

At position 0: Highest probability at position start is always 1

$\delta_1(*S*) = 1$ (2.94)

At position 1:

$\delta_{t-1}(i) = 1$ from 2.94

$a_{ij} = 0.7$ from Transistion probability Table 2.2 *Start* to *A*

$b_j O_t = 1$ from Emmision probability Table 2.3 *Start* observing **S**

substitute 2.90

$\delta_1(A) = \max[1 \times 0.7 \times 1] = 0.7$ (2.95)

In same manner $\delta_1(B)$ is calculated.

$\delta_1(B)$ = max[1 x 0.3 x 1] = 0.3 (2.96)

The back trace is both trivial as $\gamma_1(A) = \gamma_1(B) = *S_0*$ as there is only one possible path back to *start*.

From position 2 to 4 there are two possible paths.

At position 2:

where path A_1 to A_2 observing $x\delta_{t-1}(i) = 0.7$ from 2.95

$a_{ij} = 0.2$ from Transistion probability Table 2.2 *A* to *A*

$b_jO_t = 0.4$ from Emision probability Table 2.3 *A* observing *x*

and where path B_1 to A_2 observing *x*

$\delta_{t-1}(i) = 0.3$ from 2.96

$a_{ij} = 0.7$ from Transistion propability Table 2.2 *B* to *A*

$b_jO_t = 0.4$ from Emmision propability Table 2.3 *B* observing *x*

$\delta_2(A)$ = max [0.7 x 0.2 x 0.4, 0.3 x 0.7 x 0.3]

= max [0.056, 0.063] = 0.063 (2.97)

$\gamma_2(A) = B_1$ Best path is from B_1 to A_2

In the same manner $\delta_2(B)$ is calculated.

$\delta_2(B)$ = max [0.7 x 0.7 x 0.4, 0.3 x 0.2 x 0.3]

= max [0.196, 0.018] = 0.196 (2.98)

$\gamma_2(B) = A_1$ Best path is from A_1 to B_2

In same manner position 3 and 4 is calculated but the observevation is *y*.

At position 3:

$\delta_3(A)$ = max [0.063 x 0.2 x 0.6, 0.196 x 0.7 x 0.7]

= max [0.00756, 0.09604] = 0.09604

$\gamma_3(A) = B_2$

$\delta_3(B)$ = max [0.063 x 0.7 x 0.6, 0.196 x 0.2 x 0.7]

= max [0.02646, 0.02744] = 0.02744

$\gamma_3(B) = B_2$

At position 4:

$\Delta_4(E)$ = max [0.09604 x 0.2 x 0.6, 0.02744 x 0.1 x 0.7]

= max [0.0115248, 0.019208] = 0.019208

$\gamma_4(E) = B_3$

Viterbi sequence: ABB

P(ABB,*xyy*) = 0.019208

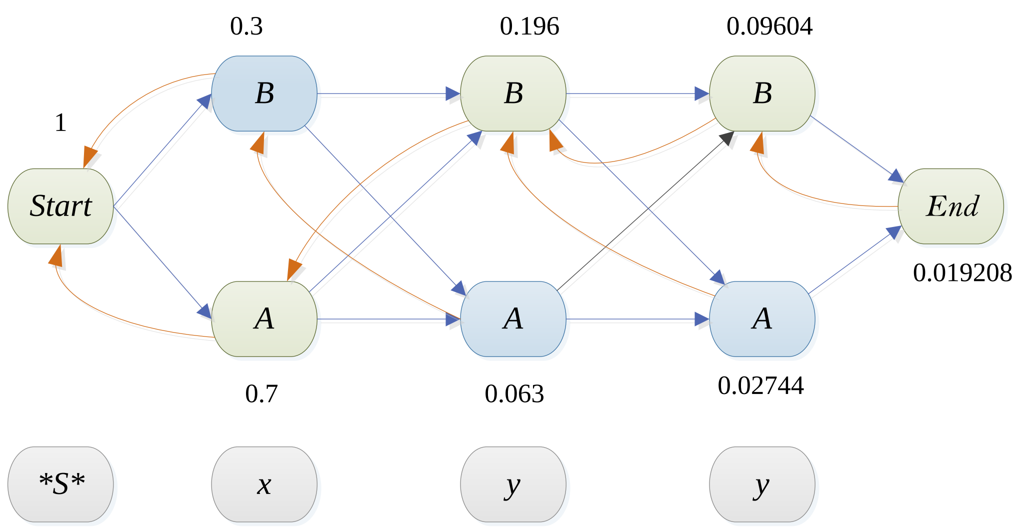


Figure 2.28 Viterbi sequence

2.4.3.3 Baum-Welch algorithm

The process of adjusting the HMM parameter is known as training the model. This is the most difficult HMM problem to be solved because there is no efficient procedure of estimating the model parameters, given any observation sequence. Choosing $\lambda = (A, B, \pi)$ such that $P(O|\lambda)$ is locally maximised can be done via the Baum-Welch algorithm that is part of the Expectation Maximisation (EM) algorithms [17, pp.2-3].

The Baum-Welch algorithm requires the forward and backwards algorithm. The forward algorithm was discussed in section 2.4.3.1 [19, pp.2-3]. Calculating the backwards variable $\beta_t(i)$ are similar and done as follows:

- Initialisation

$$\beta_T(i) = 1 \qquad 1 \leq i \leq N \tag{2.99}$$

- Induction

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j)\, a_{ij}\, b_j\, O_{t+1}, \tag{2.100}$$

$1 \leq j \leq N$ and $1 \leq t \leq T\text{-}1$

Re-estimation is the interactive update and improvement of the HMM parameters and is defined as the probability of being in state S_j at time t and the state S_j at time $t + 1$.

$$\zeta(i,j) = P(q_t = S_i, q_{t+1} = S_j \mid O|\lambda) \tag{2.101}$$

$$= \frac{\alpha_t(i)\, a_{ij}\, b_j\, O_{t+1}\, \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i)\, a_{ij}\, b_j\, O_{t+1}\, \beta_{t+1}(j)}$$

and is illustrated in [15, pp.26-28, 20, p.264].

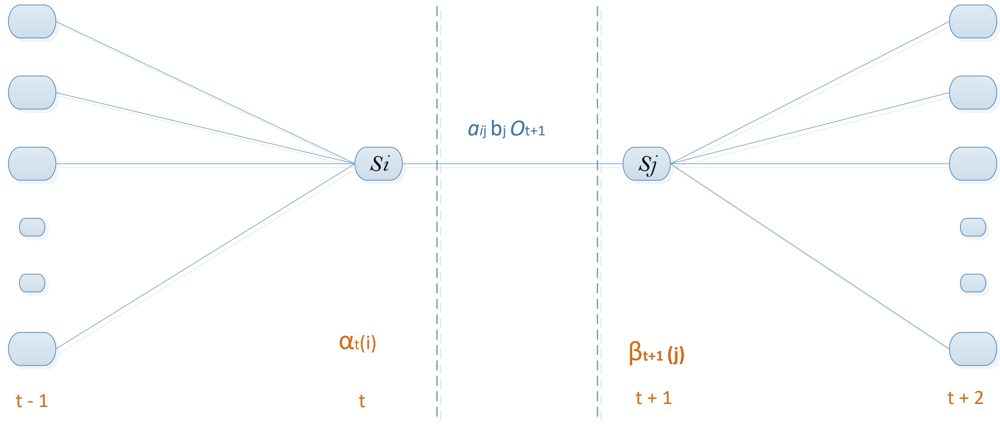


Figure 2.29 Breakdown of events and respective probabilities making up ζ(i,j)

Methods for re-estimation of the HMM parameters are [20, p.265]:

- The probability of being in state S_i at time t is:

$$\gamma_t(i) = \sum_{j=1}^{N} \zeta(i,j) \tag{2.102}$$

- Expected number of transitions from S_i is:

$$\sum_{t=1}^{T-1} \gamma_t(i) \tag{2.103}$$

- Expected number of transitions from Si to S_j is:

$$\sum_{t=1}^{T-1} \zeta(i,j) \tag{2.104}$$

A set of reasonable re-estimation formulas for A, B, π are: [15, p.27]

$$\pi'_i = \text{expected number of times } S_i \text{ at time } (t=1)$$

$$= \gamma_1(i) \qquad (2.105)$$

$$a'_{ij} = \frac{\text{expected number of transitions from state } S_i \text{ to } S_j}{\text{expected number of tansitions from } S_i}$$

$$= \frac{\sum_{t=1}^{T-1} \zeta(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \qquad (2.106)$$

$$b'_j(k) = \frac{\text{expected number of times in state j and observing symbol } V_k}{\text{expected number of times in state j}}$$

$$= \frac{\sum_{\substack{t=1 \\ \text{s.t } O_t = V_k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T-1} \gamma_t(j)} \qquad (2.107)$$

Baum did proof that maximisation of Q(λ, λ') leads to increased likelihood where

$$\text{Max}[Q(\lambda, \lambda'] \Rightarrow P(O, \lambda') \geq P(O, \lambda) \qquad (2.108)$$

Example 2.4

Three friends (F1, F2, F3) regular play Racquetball. The represented states are at which home court, they are playing at (C1, C2, C3). The observations are who wins each time. How can the model parameters be determined based on a training set of observation sequences? The recordings are presented as follows [26, pp.55-62]:

$$\pi = \begin{bmatrix} 0.3 & 0.3 & 0.4 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.2 & 0.5 & 0.3 \\ 0.4 & 0.4 & 0.2 \\ 0.1 & 0.4 & 0.5 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.2 & 0.3 & 0.5 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

O = F1, F3, F3 = Training Set

Table 2.4 illustrates the calculations for the forward algorithm where formula 2.83 where used in column 1 and formula 2.84 in column 2 and 3.

Table 2.4 Racquetball example forward algorithm

	α1(1)	α2(1)	α3(1)
	t = 1, Ot = F1	t=2, Ot=F3	t=3, Ot=F3
C1	0.3x0.5 = 0.15	(0.15x0.2 + 0.6x0.4 + 0.4x0.1)x0.3 = 0.017	(0.017x0.2 + 0.058x0.4 + 0.062x0.1)x0.3 = 0.010
C2	0.3x0.2 = 0.06	(0.15x0.5 + 0.6x0.4 + 0.4x0.4)x0.5 = 0.058	(0.017x0.5 + 0.058x0.4 + 0.062x0.4)x0.5 = 0.028
C3	0.4x0.1 = 0.04	(0.15x0.3 + 0.6x0.2 + 0.4x0.5)x0.8 = 0.062	(0.017x0.3 + 0.058x0.2 + 0.062x0.5)x0.8 = 0.038

Table 2.5 illustrates the calculations for the backwards algorithm where formula 2.11 where used in column 1 and 2 and formula 2.99 in column 3.

Table 2.5 *Racquetball example backword algorithm*

	$\beta_1(1)$	$\beta_2(1)$	$\beta_3(1)$
	t = 1, O_{t+1} = F3	t = 2, O_{t+1} = F3	T = 3
C1	0.2x0.3x0.55 + 0.5x0.5x0.48 + 0.3x0.8x0.63 = 0.3	0.2x0.3x0.1 + 0.5x0.5x0.1 + 0.3x0.8x0.1 = 0.55	1
C2	0.4x0.3x0.55 + 0.4x0.5x0.48 + 0.2x0.8x0.63 = 0.26	0.4x0.3x0.1 + 0.4x0.5x0.1 + 0.2x0.8x0.1 = 0.48	1
C3	0.1x0.3x0.55 + 0.4x0.5x0.48 + 0.5x0.8x0.63 = 0.36	0.1x0.3x0.1 + 0.4x0.5x0.1 + 0.5x0.8x0.1 = 0.63	1

$P(O|\lambda) = P(F1\ F3\ F3|\lambda)$

$= \Sigma\ \pi_i\ b_i\ O_1\ \beta_t(i)$

$= 0.3 \times 0.30 \times 0.5 + 0.3 \times 0.26 \times 0.2 + 0.4 \times 0.36 \times 0.1 = 0.076$ (2.109)

The Baum-Welch π Vector is:

$\pi'_i = \gamma_1(i)$ Substitute 2.109 in 2.105

$\pi'_1 = 0.15 \times 0.30 / 0.076 = 0.60$ (2.110)

$\pi'_2 = 0.06 \times 0.26 / 0.076 = 0.21$ (2.111)

$\pi'_3 = 0.04 \times 0.36 / 0.076 = 0.19$ (2.112)

The Baum-Welch Transition Matrix is: Substitute 2.106

$a_{ij}' = \Sigma_t\ \zeta(i,j) / \Sigma_t\ \gamma_t(j)$

$= [\Sigma_t\ \alpha_t(i)\ a_{ij}\ b_j\ O_{t+1}\ \beta_{t+1}(j)] / [\Sigma_t\ \alpha_t(i)\ \beta_t(i)]$ (where sum is from 1 to T-1)

$a_{12}' = (0.15 \times 0.5 \times 0.5 \times 0.48 + 0.17 \times 0.5 + 0.5 \times 0.1) / (0.15 \times 0.30 + 0.17 \times 0.55)$

$= 0.41$ (2.113)

The Baum-Welch Observation Matrix is: Substitute 2.107

$b'_i(k) \quad = \Sigma_{t \text{ Observing } V_k}\ \gamma_t(j) \ / \ \Sigma_t\, \gamma_t(j)$

$b_{jk}' \quad = [\Sigma_{t \text{ and } O_t = V_k}\ \alpha_t(j)\ \beta_t(j)] \ / \ [\Sigma_t\ \alpha_t(j)\ \beta_t(j)]$ (where sum is from 1 to T)

$b_{23}' \quad = (0 + 0.058 \times 0.48 + 0.28 \times 1) \ / \ (0.06 \times 0.26 + 0.58 \times 0.48 + 0.028 \times 1)$

$= 0.79$

2.5 *More efficient heating water techniques*

2.5.1 Heating water with solar thermal energy with a backup element

In South Africa the common name for a Hot Water cylinder (HWC) is a Geyser. Research done by Prof. Dintchev OD at Pretoria (60km North from where this research was done) indicate that in the summer month the performance of the solar geyser is very satisfactory and the electrical backup element active time is minimal as indicated in Figure 2.30. It was recommended that in summer month the electrical element is switched of during the day for maximum solar heating benefits, but switch on when it is dark or when it is cloudy and rainy. During the winter month the solar heating is reduced by 70% and the backup element should be controlled by a controller all the time [27, p.5].

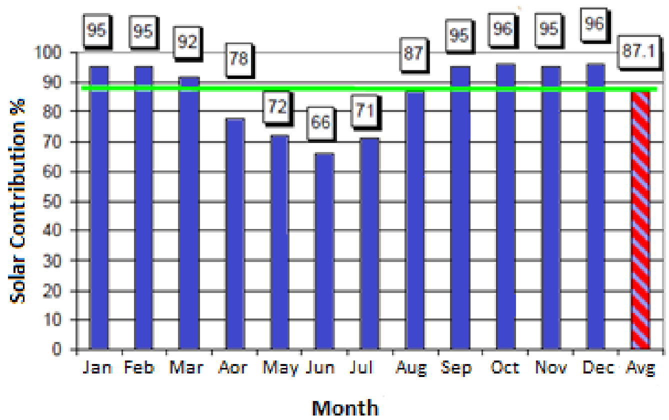


Figure 2.30 Annual solar contribution to water heating

Research done by Sauer Frank Joseph at the Missouri university of science and technology did proof by using Clustering and Hidden Markov Models (HMMs) solar radiation can be calculated for a specific location from temperature, sky coverage and previous years' radiation from a national database [28, pp. 5,6,24].

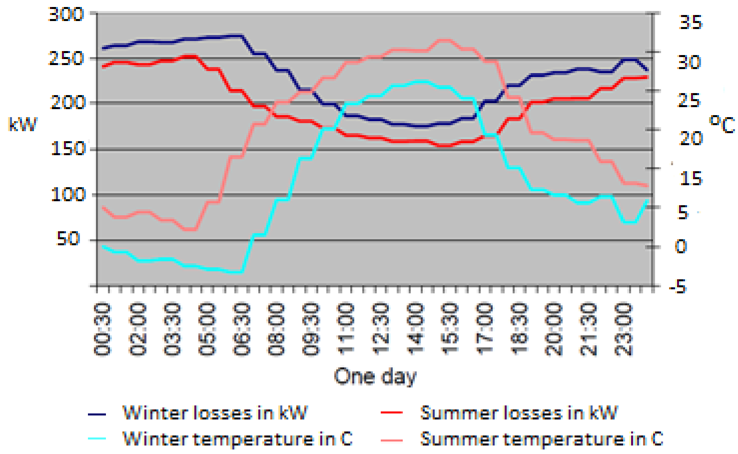


Figure 2.31 Comparison of losses and outside temperature

Research done by Johan Delport indicates that the geyser loses in winter is higher than in summer as indicated in Figure 2.31. His research also shown the use of hot water in winter is higher than in summer as illustrated in Figure 2.32 [29, p.2].

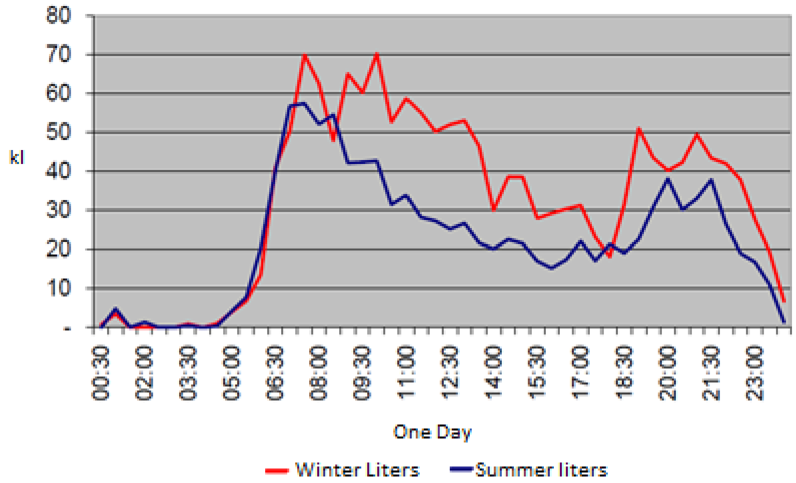


Figure 2.32 Comparison of winter and summer hot water usage

Heating water with a solar radiation without electrical backup for long periods during inefficient cycles can be dangerous due to a disease known as Legionellosis. This is caused by a bacterium known as Legionella pneumophila responsible for several fatalities worldwide [29, p.6]. The bacteria react to water temperatures as follows:

- Sub-zero to 20°C: Dormant;
- 20°C to 45°C: Suitable growing conditions.

Suitable growing conditions for the bacteria is between 20°C and 45°C, with 32°C to 42°C being the ideal growth range [30, p.13]. The bacteria are dormant below 20°C, even below freezing point. The bacteria survive but do not multiply for temperature between 48°C and 50°C. The bacteria die within 5 to 6 hours at 55°C and dies within 32 minutes at 60°C. At temperatures above 70°C the bacteria are killed immediately [31, p.30]. It is important to incorporate the temperature and times it takes to kill the bacteria into the design of a geyser controller.

2.5.2 Dual element

One of the geyser biggest energy losses is the small draws of hot water. The element will switch on because the thermostat detected a decrease in water temperature. The idea is to replace one 4kW element with two 2kW elements or two 1.5kW elements instead of one 3kW element.

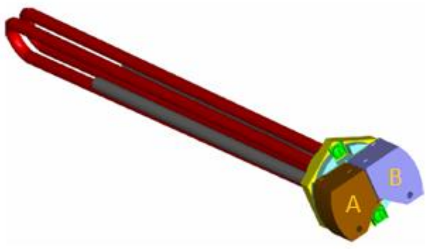


Figure 2.33 Dual element

Each element is controlled by its own thermostat, but one is set 5°C lower. Now with small water usage, approximately up to 8 litres, only one element is switched on to reheat to water to the maximum temperature. Only one element will switch on for normal heat losses when no hot water is used. With a single element, it is found that the system overheats by average of 4°C. With the dual element the temperature is regulated near the setpoint and overheating temperatures are halved [32, pp. 1-2].

2.5.3 Element and thermostat position

Hot water that is drawn by a geyser is replaced with cold water. Cold water remains at the bottom of the cylinder because it is denser than hot water [33, p.1]. For this reason, geysers cold water inlet is at the bottom and hot water outlet at the top. Research done have proof that the hot water temperature differs at different levels in a geyser [33, p.2]. Figure 2.34 shown a modified 100 litre geyser with six positions to read the temperature.

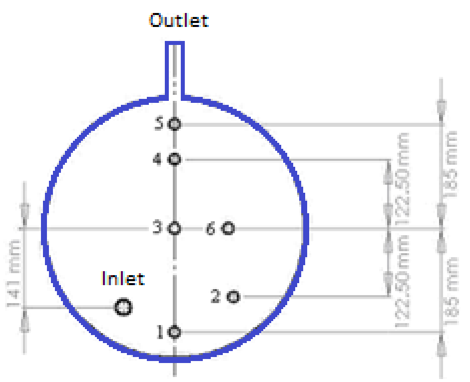


Figure 2.34 Geyser thermostat position

Position three is the standard place where geyser elements and thermostats are installed.

Each position corresponds to the following volume of water:

1. 5 litre;
2. 20 litre;
3. 50 litre;
4. 80 litre and
5. 98 litre [33, p.1].

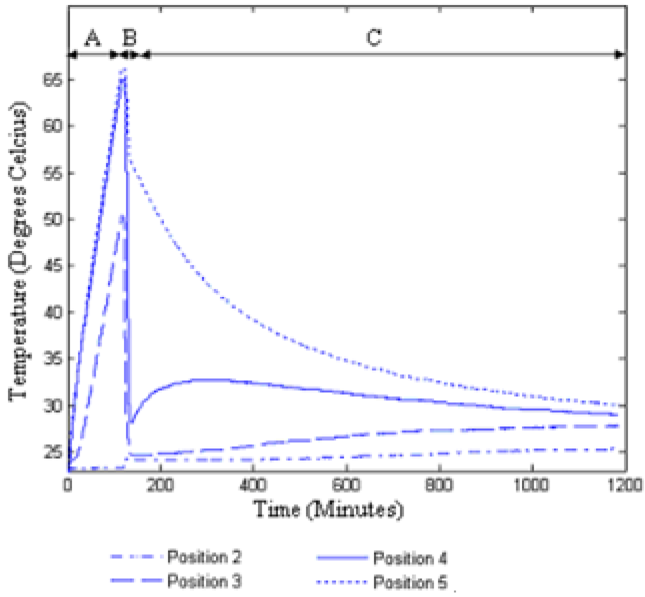


Figure 2.35 Temperatures changes at different positions in a geyser

Figure 2.35 illustrates the different temperate recorded over time at different positions. Region A illustrates that when then the water is heated to 50°C, the upper region heats at 45°C /*l* and the lower rate at 27°C /*l* and that only 20*l* remains to the initial temperature. Region B illustrates the effect of a flow rate. Cold water enters the system at the bottom and sensors 1 to 3 temperatures drops to the cold water temperature, while sensor 5 drops by 10°C. The difference in temperature is because of the boundary between cold and hot water in the geyser. Region C illustrates when the hot and cold water starts to mix over time. The only losses are standby losses consist of air-losses and the effect of stratification [33, pp.1-2].

2.6 *Swimming Pool*

Today it is common knowledge that a HWC or geyser is the most energy intensive appliances in a home. Most research for this reason was done to save cost for a geyser.

According to South Africa's electricity supplier Eskom, swimming pool pumps is the third largest contributor in a house responsible for up to 11% of the monthly bill [34]. They indicate that research have proven that it is sufficient to run the pool for 4 hours per day in winter and 6 hours per day in summer. The rule of thumb is that all the water in a pool should be filtered at least once every 24 hours and twice when the pool is used frequently [34], [27, p.12].

In colder areas, pool pumps are used to circulate the water through solar heating panels or heat pumps to extend the swimming season. Heat loss from water can be due to evaporation and convection losses.

Evaporation composes about 65% - 70% of total heat loss from pools. The amount of evaporated water can be predicted as follows:

$$E = 30.6 + 32.1U \times (Pw - Pa) \times 3600 \quad / \Delta H \tag{2.114}$$

Where:

E : Rate of evaporation at the water surface (kg.h.m^2)

U : Wind speed over surface water (m/s)

Pw : Saturated vapor pressure at surface water temperature (kPa)

Pa : Vapor pressure of the air dew point temperature (kPa)

ΔH : Latent heat of water for specific water temperature (kJ/kg)

Formula 2.114 proves that the evaporation rate is proportional to environmental conditions such as temperature, humidity and wind speed [35, p.10].

The heat loss from the water surface due to convection can be calculated as follows:

$$Q_{conv} = h \times (Tp - Ta) \times A \qquad (2.115)$$

Where:

Q_{conv} : Convection heat transfer rate

h : Convection heat transfer coeffient

Tp : Pool water temperature

Ta : Air temperature over pool.

The heat transfer coefficient (h) for an outdoor pool was found by Czarnecki to be:

$$h = 3.1 + 4.1U \qquad (2.116)$$

Formula 2.115 and 2.116 proves temperatue and wind speed influens convection heat losses [35, p.11].

2.7 *Motion sensor light switches*

Motion sensitive light switches are growing in popularity and reducing energy expenses if implemented in public and corporate spaces.

Motion sensors light switches are expensive and consume little phantom energy. Assuming a motion switch last 15 years and a 60W bulb is switched on for 14 hours and longer a day, every day for a year before it will make economic sense to install a motion sensor. It is more economical to invest in energy efficiency light than motion sensors [36].

2.8 *Sensor communication*

Communication between the controller and sensors is crucial when a smart house controlling system is deployed. When the deployment is at an existing building, cabling is expensive because building alterations are required to install the cabling [37, p.17]. Historically, home networks were used to interconnect computers, printers, entertainment devices and connecting to the internet, but which IoT everything is now interconnected [38, p.10].

Two of the most common communication are over existing electrical wires or wireless communication.

2.8.1 **Power-line communication**

Power-line communications (PLC) transports data over a medium that is simultaneously used for AC electrical power distribution. This is not a new concept and one common application is where the municipalities control household geysers.

In-house PLC uses the same voltage as low-voltage PLC, which is 220 to 400V or 110V in the USA. It is commonly used to interconnect computers, security systems and automatic light control. The PLC modem basically acts as a high pass filter splitting the communication signals 9kHz and higher from the electrical powers 50 to 60Hz [37, p.21]. A main disadvantage is that without any special filters the data from one house are presented at any power outlet in other houses connected to the same distribution transformer.

2.8.2 **Wi-Fi**

Wireless LAN's significantly improved performance and cost over the past years making it a cost effective technology [37, p.22]. It complies with the IEEE 802.11 Wireless LAN standards [38, p.232] and have built in security options. The most common security mechanism used is Protected Access Version 2 (WPA2) with Advance Encryption Standards (AES) which is most efficient for wireless home networks. WPA2 uses encryption keys and a key index that is regularly changed. AES use a private-key block-cipher technique in which plain text is divided in 128, 192 or 256 bits length. Another security feature is the use of the Service Set Identifier (SSID) [38, p.256]. All Wi-Fi devices connect to an Access Point (AP) and if two devices communicate with each other it will be via the AP following a star network topology.

2.9 *Conclusion*

Knowledge of technologies and previous research discussed in chapter 2 guided the study to the development of a prediction model in the following chapters set out in the objectives of the study.

Chapter 3

3 Development of data recorder

Before a prediction model could be developed, a system was needed to be implemented to collect and analyse the environmental data. An off-the-shelf solution was evaluated, but none of those available could retrieve the raw data from the sensors which were essential for this research. An IoT sensor and controller device was developed for the research to retrieve and manipulate the data at a later stage. The following sections explain the use and implementation of hardware implemented for measurement, control and communication.

3.1 *Basic layout*

A decision was made that the only interface to the user or the internet will be via the controller because:

- With a Hub and Spoke architecture, all the data are stored centrally and simplify the research and design of the prediction model;
- The most important is security. Securing one central device is more secure and simplify the design of the sensors and keeping the cost to a minimum because the sensors are not reachable via the internet.

Figure 3.1 shows a basic layout between the controller and the IoT devices. The IoT devices can be a Wi-Fi device with multiple sensors. The device can have any combination of sensors and they do not need to be of the same type. The IoT device can also be internet-based, like weather predictions or a user controlling the setup via a web page. See Appendix B for the IoT schematic and printed circuit board (PCB) diagrams and photos of the device.

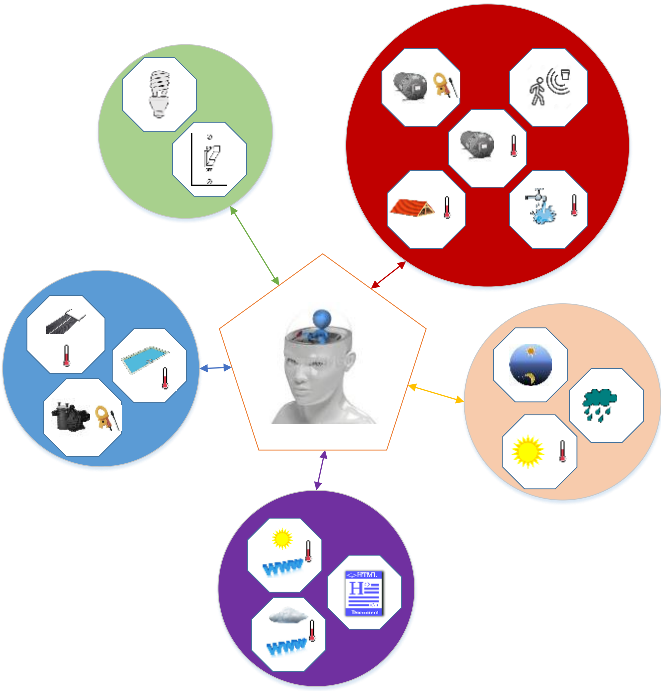


Figure 3.1 Basic controller and devices with sensors layout

Figure 3.2 illustrates a basic Block diagram of the controller that must analyse and record all input data. It must use the recorded data to formulate a prediction module to control sensor endpoints connected to it.

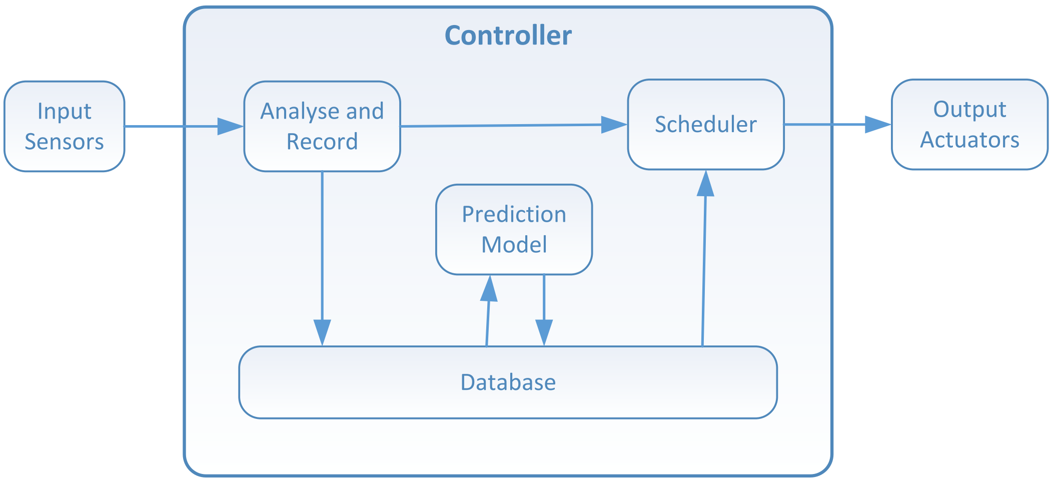


Figure 3.2 Basic controller layout

3.2 *IoT Module*

Different IoT modules were investigated for this purpose such as: the Raspberry Pi, Onion Omega, Arduino and the ESP8266 series. The ESP8266-12E module was decided upon because it met all the requirements needed for the research. The ESP8266 module is manufactured by Espressif Systems and includes the following features:

- Built-in Wi-Fi gateway module, with a full Transport Control Protocol/ Internet Protocol (TCP/IP) protocol stack;
- 32-bit processor;
- 64kB Random Access Memory (RAM) and 512kB of external memory;
- Universal Asynchronous Receiver/Transmitter (UART);
- General Purpose Input/output (GPIO);
- Analog to Digital Converter (ADC);
- Serial Peripheral Interface (SPI);
- Pulse Width Modulation (PWM);
- operating at a supply voltage of 3.3V at 300mA [39, p. 41].

The NodeMCU development board with an ESP8266-12E as illustrated in Figure 3.3 was chosen because of its availability in the area the research was done. It is a ready-made solution for an easy way to connect to the internet.

Overcoming the single analogue limitation, a LM4052N multiplexer was connected that provides four analogue inputs. One LM4052N analogue input is connected to a 30A/1V Non-Invasive clip-on current sensor from Yhdc. The ESP8266-12E monitors the geyser element and pool pump motor with a sealed electromagnetic H105F-1

SPST relay providing a maximum contact current of 40A and average resistive load of 30A@240V AC. Refer to Appendix B for the hardware used.

There are a variety of programming tools and languages available but ESP8266 Basic was chosen ESP8266 Basic for its support for most technologies used to build the IoT sensors for example:

- Building Web Graphical User Interface (GUI);
- USB serial support;
- Timers;
- GIO pins can be input or output and can be set as PWM pin;
- Support DHT11, DHT21 and DHT22 temperature and Humidity sensors;
- OLED, LCD and TFT displays;
- Wi-Fi
- User Datagram Protocol (UDP)

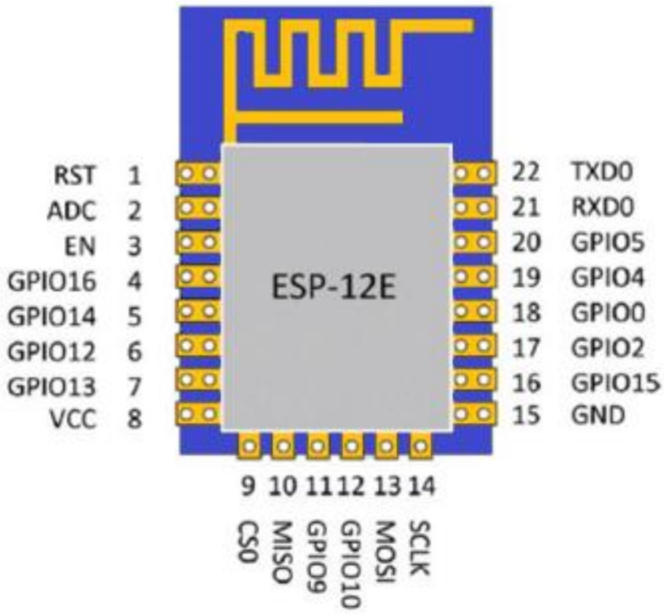


Figure 3.3 *ESP8266-12E*

Appendix A discusses the procedure to flash the ESP8266-12E with ESP8266 Basic firmware and the layout is illustrated in Figure 3.3.

3.3 *Controlling instances basic functions*

Multiple cloud controller solutions are available for IoT applications like Thingspeak, Google Cloud IoT, Autodesk Fusion Control and ThingWorx to name a few but because the decision was made to incorporate a recorder, analyser and prediction model to the controller, an algorithm was developed for the communication processes between the central controller and IoT sensors. Firstly, an algorithm was developed to register the IoT sensors to the controller as explained in Appendix C. Secondly, a process was designed to associate a sensor with a controller instance as explained in Appendix D.

It is important to understand the basic design functions of each instance. It will be used to record the data that will be analysed in Chapter 4. This interface will be modified in Chapter 5 to accommodate the prediction models.

3.3.1 Sensor interface

To simplify the research, it was required to verify sensor readings from the sensors with the recorded data from the controller. The ESP8266 Basic Web GUI simplifies this requirement and Figure 3.4 illustrates the web GUI that was designed for the first IoT device. The following functions can be done from the web interface:

- Switching lights on or off and view status;
- Switch geyser on or off, view status, set the maximum override and view current drawn;

- View geyser, roof and hot water pipe temperatures;
- Switch pool pump on or off, view status and view current drawn;
- View pool, water from solar heating and environmental temperatures;
- View date and time;
- Set controller IP address and device name;
- Connect to controller;
- Display device IP address;
- Display last UDP message.

This interface can now be used to control the devices via a web interface from any IP device. The interface are discussed in more details in the following section.

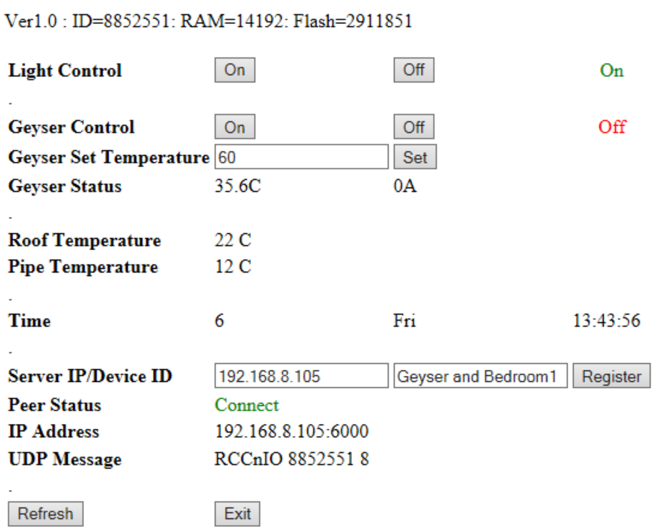


Figure 3.4 *Device web-based interface*

3.3.2 Geyser control

The minimum requirements for the geyser controller are:

- On and Off control via the Controller;
- On and Off control via IoT;
- Geyser relay (Status);

- Set the maximum geyser temperature;
- Read geyser, roof (ambient) and hot water pipe temperatures;
- Read current drawn by the geyser.

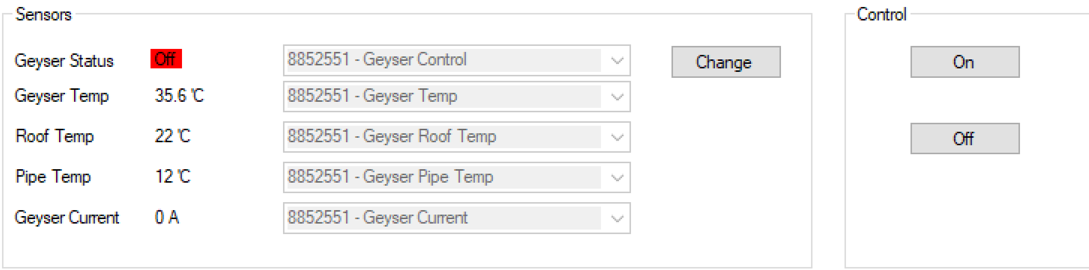


Figure 3.5 Geyser controller's view: Geyser deactivated

Figure 3.5 illustrates the controller's view to control the geyser and displaying all geyser-related information. Five sensors were linked in the same manner as discussed in Appendix D. The sensors are the geyser inside temperature, roof (ambient) temperature, hot water temperature, current drawn by the element and the relay status that controls the geyser element. When the "Off" button is clicked on the controller or the sensor's web interface, the geyser element is deactivated. It is important to notice that there is no variable to set the geyser temperature. This is incorporated into the prediction model discussed in Chapter 5.

Figure 3.6 illustrates the web interface of the sensor when the geyser's element is switched on. A maximum geyser temperature of 60 °C was set on the sensor. Until the prediction model is done, it is the temperature the geyser will heat the water too, but its primary goal is a safety override. When the controller determines the temperatures, the sensor will always deactivate the element when this temperature is reached. This is important if for any reason the controller activates the geyser element, but then loses communication with the sensor and cannot deactivate the element.

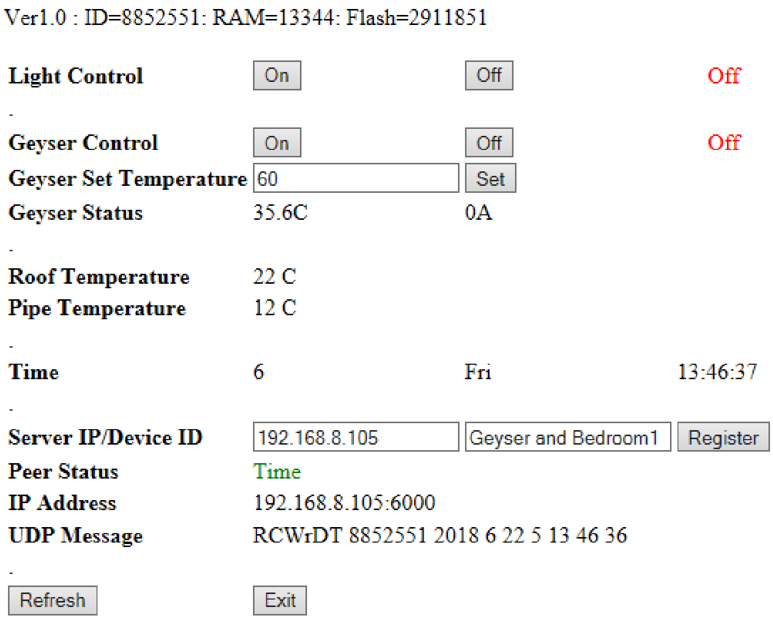


Figure 3.6 Sensor's view: Geyser deactivated

To activate the geyser element, the "On" button can be clicked on the controller or via the sensor's web interface as illustrated in Figure 3.7. The current sensor records that the element is drawing 15.9A and this feedback is used to confirm that the element is switched on. This action will be used by the user if the prediction model fails and the hot water temperature is incorrect. The result will be used to determine the accuracy of the prediction model and to train the model.

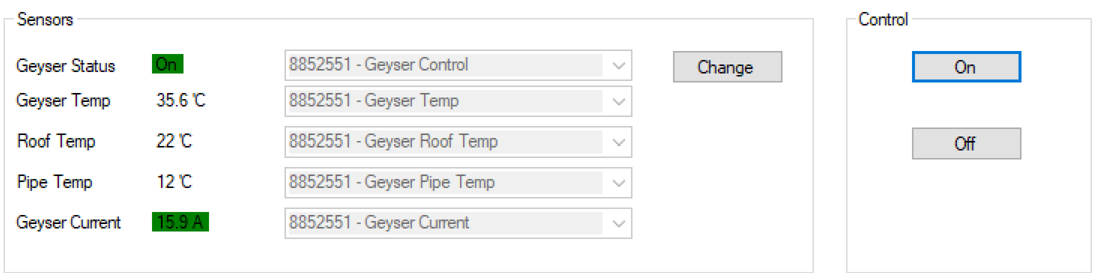


Figure 3.7 Controller's view: Geyser activated

Figure 3.8 illustrates the controller interface after the geyser reached 60 °C. The geyser element deactivates, but the status indicates “Cooldown” instead of “Off”. The current reading confirms the element status. The sensor will only allow the water to cool down to below the maximum temperature and then activate the element again. A variable delta temperature is included in the prediction model discussed in Chapter 5.

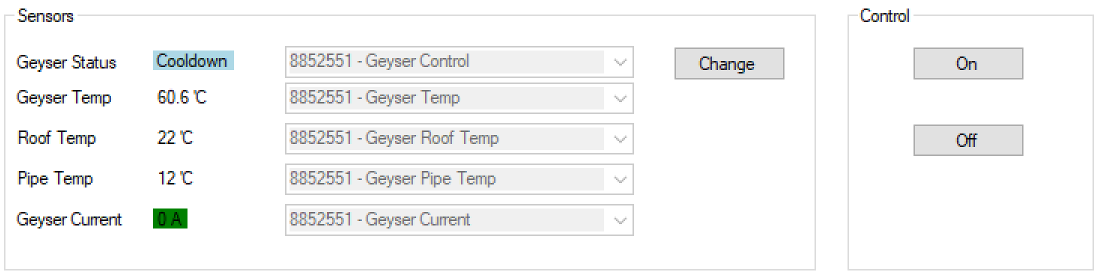


Figure 3.8 Controller’s view: Geyser temperature cooldown

Another built-in protection mechanism that was included in the sensor was to monitor for any electrical problems. If there is an electrical wire or element malfunction, the geyser relay will activate but the element will not draw any current. The sensor will immediately deactivate the element and update the controller with a Current Error status as illustrated in Figure 3.9.

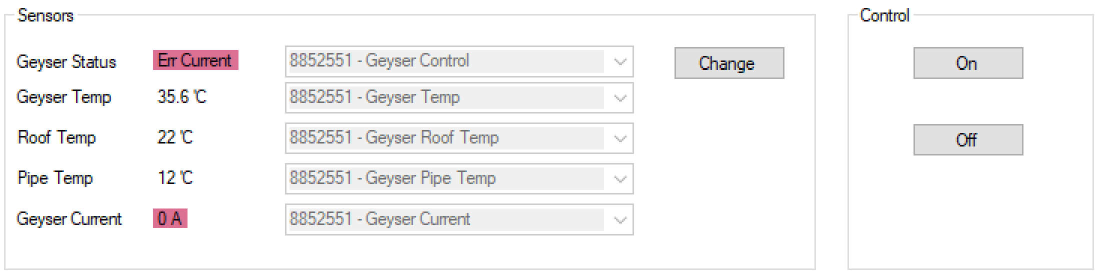


Figure 3.9 Controller’s view: Geyser current error

The above indicates that the geyser relay that normally only have two states, “On” or “Off” now have four conditions. This is why Table C.1 defining the endpoints, indicates that Type ID 100 for RelayGeyser is an analogue and not digital type.

The statuses for the RelayGeyser can be:

0. Off;
1. On;
2. Cooldown;
3. Current Error.

3.3.3 Light control

As discussed in section 2.7, it is uneconomical to use a motion sensor light switch to save energy in a residential house. This solution only makes economical sense in a corporate environment. Occupation will be investigated to be included in the prediction module to control the pool pump motor and the geyser element to save more energy. This technique can now be used to save more energy with minor energy consumers. The alarm's motion sensors were used.

The minimum requirements for the light controller is:

- On and Off control via the Controller;
- On and Off control via the IoT;
- On and Off control via the Light switch;
- Light relay (Status);
- Automatic control via the Controller
- Read movement sensor
- Set maximum movement delay.

Figure 3.10 illustrates the controller's view to control a light and related information. Three sensors were linked, as discussed is Appendix D, light switch, movement sensor and light relay indicating the light status. The movement delay variable is the seconds the light must be switched on after movement was recorded. The modification to the motion sensor light method, will be to allow the prediction model to modify the time delay preventing unreliable deactivation of the light.

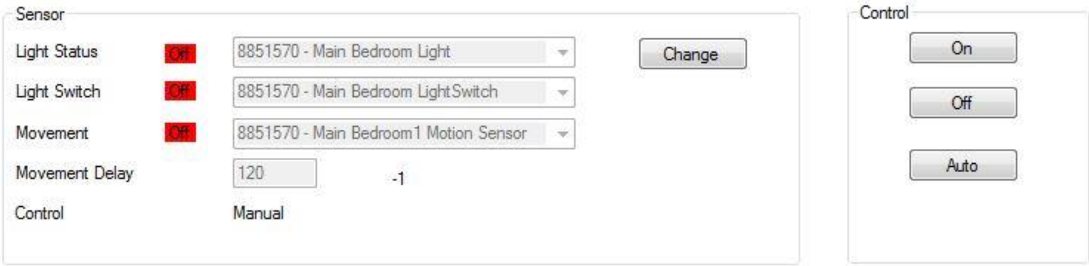


Figure 3.10 *Controller's view: Light deactivated*

Figure 3.11 shown the controller's view, indicating that the wall light switches are switched on. The controller did send a request to the sensor to activate the light. It is important to note that the IoT sensor has no knowledge which sensor is controlled and all the decisions are made by the controller.

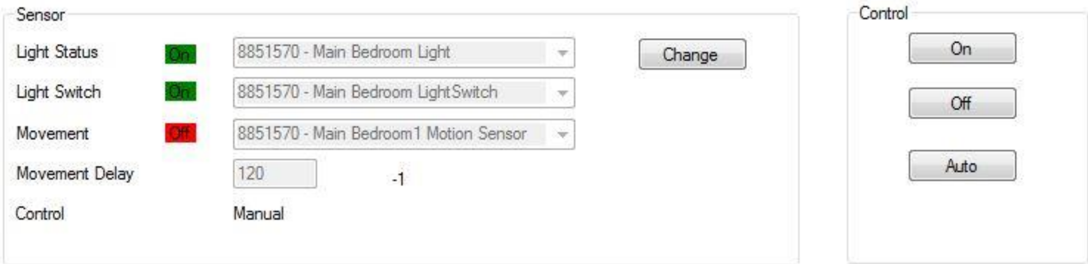


Figure 3.11 *Controller's view: Light switch activated*

In Figure 3.12, the control indication changed from Manual to Auto, indicating that the Auto button on the controller was clicked. The light will now be switched off 120 seconds after last recorded movement.

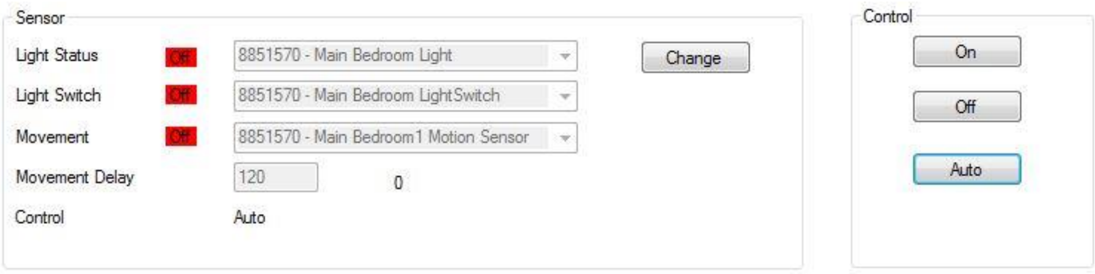


Figure 3.12 Controller's view: Time delay mode

Figure 3.13 illustrates that the movement sensor did sense movement and informed the controller. The controller sent a request to the sensor to active the light relay. The countdown time starteds counting down after no movement was recorded.

Figure 3.13 Controller's view: Movement sensor activates a light

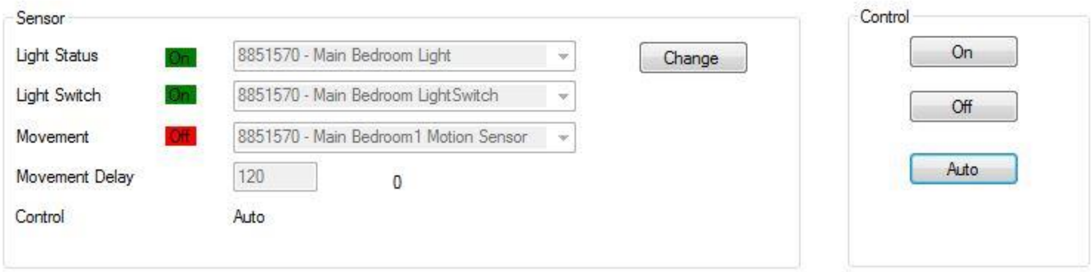


Figure 3.14 Controller's view: Light switch overrides movement sensor

Figure 3.14 illustrated that in auto mode, when the wall light switch was activated that it overrode the time delay and activated the light relay. This action would be used by the user if the prediction model failed and the light activation was incorrect. The result would be used to determine the accuracy of the prediction model and to train the model. Another result could be that the countdown timer expired and was being activated immediately again, indicating that the countdown time was too short.

3.3.4 **Pool control**

The pool IoT operation is identical to the geyser IoT. The sensor names and Endpoint type are different to allow the controller to link correct sensor as discussed in Appendix C and illustrated in Figure 3.15.

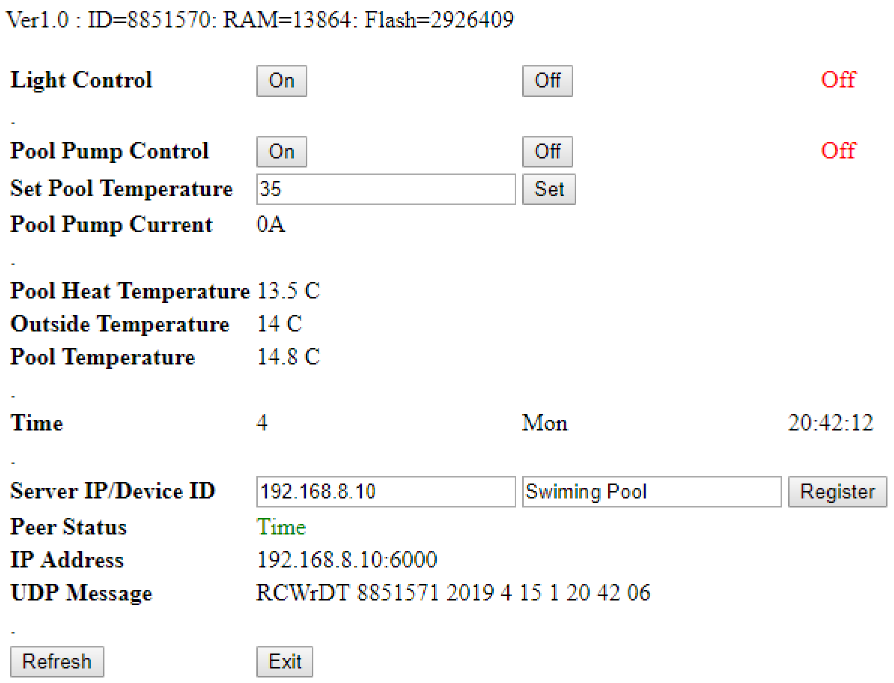


Figure 3.15 Pool sensor web-based interface

The minimum requirements for the swimming pool controller is:

- On and Off control via the Controller;
- On and Off control via the IoT;
- Pool pump relay (Status);
- Set maximum swimming pool temperature;
- Read pool, outside (ambient) and roof heating panel temperatures;
- Read current drawn by pool pump.

It is important to understand that the research does acknowledge that pool pumps have inductive load and that the current lags the voltage. In the research, no power factor was used to calculate the kWh. The current drawn was a verification that the pool pump had been activated. Ensuring that the pool cleaning function is not impacted upon, the controller must ensure the pool pump is enabled for minimum time periods as discussed in section 2.6.

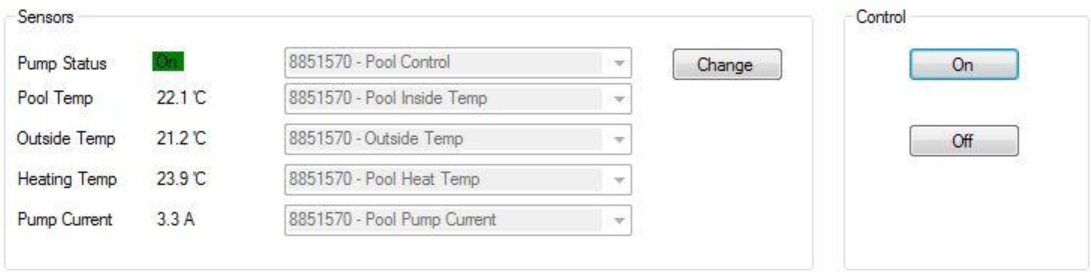


Figure 3.16 Pool controller's view: Pump activated

Figure 3.16 illustrates the controller's view to control the pool pump and displaying all related information. Five sensors were linked in the same manner as discussed in Appendix D. The sensors are the pool temperature, outside (ambient) temperature, hot water temperature from the roof, current drawn by pool pump and the relay status that controls the pool pump. When the Off button is clicked on, the controller or the sensor's web interface, the pool pump is deactivated and activated when the On button is clicked.

3.3.5 Weather prediction

If the system has knowledge about the weather in advance, more accurate prediction can be made for example: If there is a high likelihood that it is going to be overcast and rainy during the day, the geyser water can be heated with electricity from early in the morning because it is predicted that the solar collector will be insufficient. The swimming pool pump does not need to pump water to the solar heaters for the entire day as the changes are slim that anybody will swim that day. The weather prediction was implemented by using a web base rather than redeveloping a weather prediction measuring necessary variable as this is an acceptable method being used?

Weather prediction in xml format can be retrieved for free from openweathermap.org as illustrated in Example 3.1 in xml format. The three-day prediction only shows entries for every three hours. A shortfall with the prediction is that it does not have any information about solar radiation.

Example 3.1

```
<sun rise="2018-01-12T03:28:29" set="2018-01-12T17:05:59"></sun>
<forecast>
  <time from="2018-01-12T09:00:00" to="2018-01-12T12:00:00">
    <symbol number="500" name="light rain" var="10d"></symbol>
    <precipitation unit="3h" value="0.5025" type="rain"></precipitation>
    <windDirection deg="184" code="S" name="South"></windDirection>
    <windSpeed mps="3.25" name="Light breeze"></windSpeed>
    <temperature unit="celsius" value="27.67" min="27.08" max="27.67">
    </temperature>
    <pressure unit="hPa" value="858.47"></pressure>
    <humidity value="53" unit="%"></humidity>
    <clouds value="scattered clouds" all="32" unit="%"></clouds>
  </time>
```

The information is stored in a database to be analysed by the prediction module Figure 3.17 illustrates the graphical interpretation of the xml file retrieved from openweathermap.org.

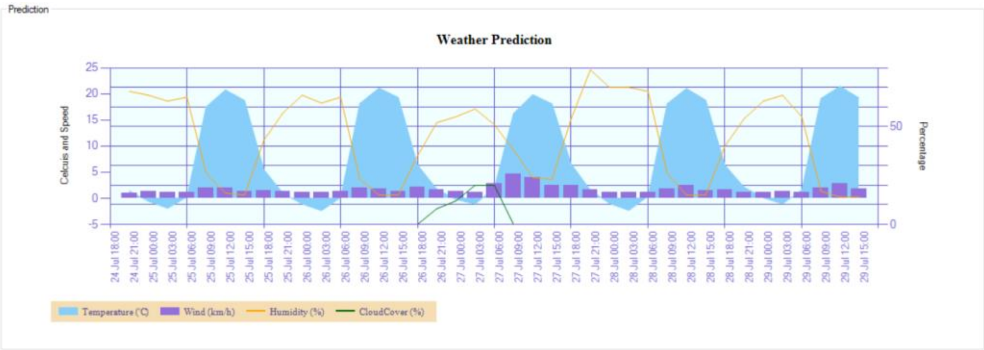


Figure 3.17 *Weather Prediction*

With all of the hardware in place, the capturing of the data could be done for analysing the data.

Chapter 4

4 Analyse recorded data and formulate predictions influences

Knowledge of hot water usage by the residents and weather patterns is essential to the prediction of the power cycles for a backup electrical element for a geyser and pool pump motor. The water usage is analysed in sections 4.1 to 4.3 and the weather patterns against geyser solar heating is section 4.4. Pool heat loss and gain is analysed in section 4.5 and the weather patterns against pool temperatures in section 4.6.

The techniques used to analyse a geyser, connected to solar heating panels, isalso used to analyse a swimming pool pump, with solar heating panels, in Chapter 5.

4.1 *Geyser usage*

The 200-litre geyser that was monitored for this research was also connected to a solar collector with a smart geyser controller controlling the electrical backup. Figure 4.1 illustrates the setup of connecting a solar collector to an existing geyser with an element.

Four people were busy with their daily routines, which include going to work or school for both days that were analysed below. The solution must be able to detect different levels of hot water usage because the property under investigation has a shower and bath. Geyser, pipe and roof temperatures with current recording is illustrated in Figure 4.2, depicting a clear difference between a shower and bath usage.

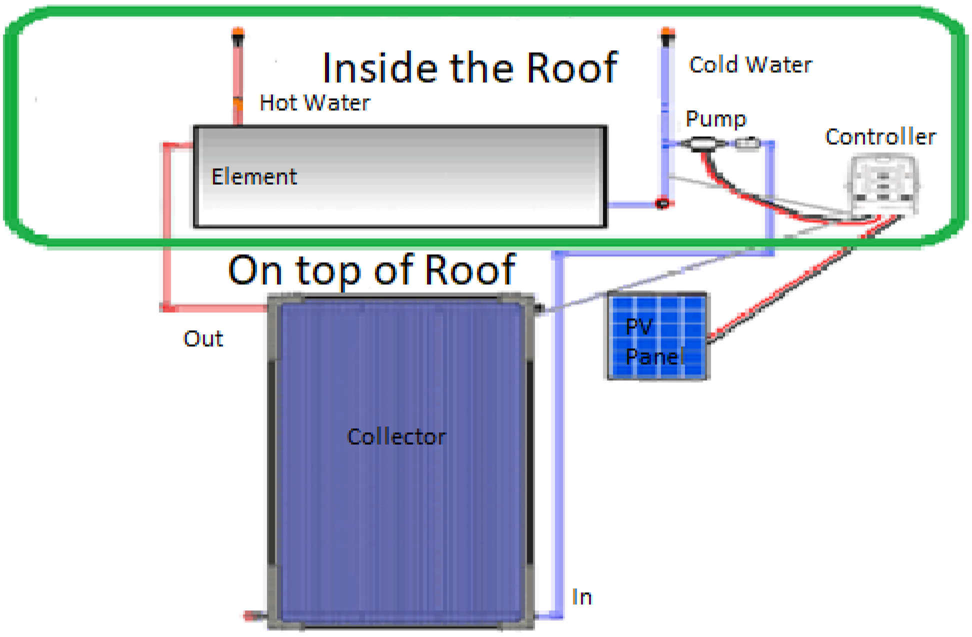


Figure 4.1 *Existing electrical geyser with outside solar collector.*

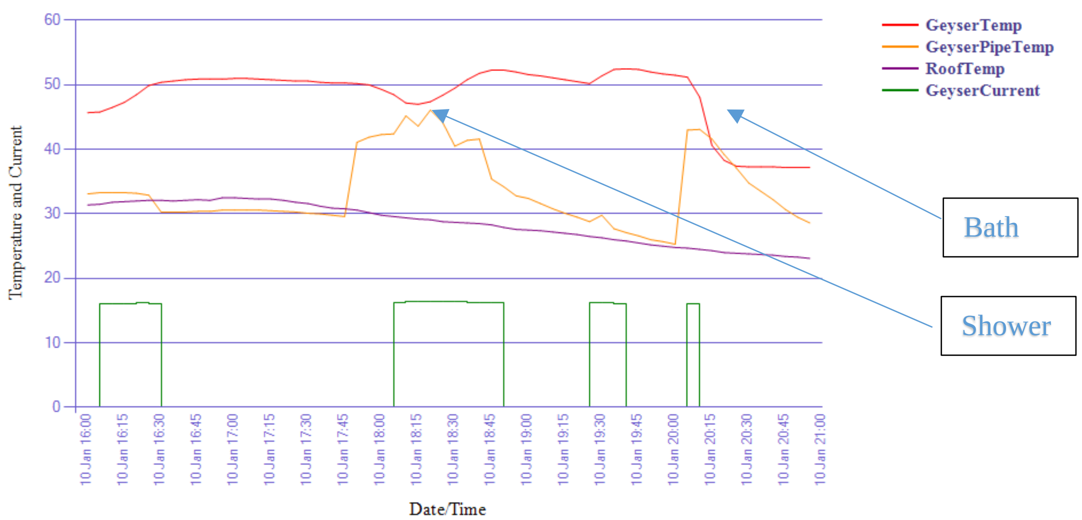


Figure 4.2 *Difference between a shower and a bath*

It was decided to use Fuzzy Logic to group the geyser usage in Fuzzy linguistic terms: “None”, “Small”, “Medium” and “Large”. Sensors were used to record a month of data included in the Fuzzy set through the developer’s knowledge base [6, p.147]. Fuzzylite 4.0, developed by Dr. Juan Rada-Vilea as part of his Ph.D. thesis [40], was used in this research to develop Fuzzy sets before it was implemented in the controller. The delta values for each membership function were taken over five-minute periods and the highest usage per hour were recorded for further predictions.

Only the geyser temperature and hot water pipe were used to determine if the Fuzzy Logic could correctly classify the usage, for the first trial.

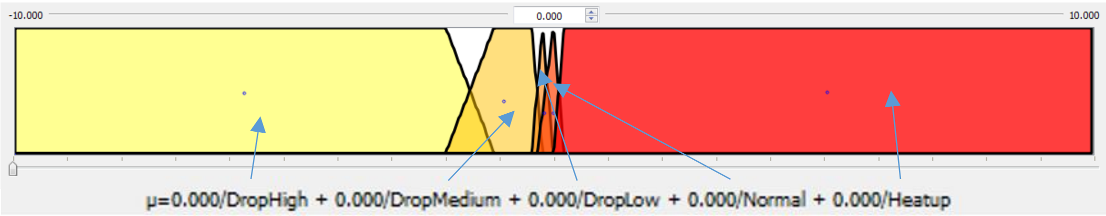


Figure 4.3 *DeltaGeyserTemp Fuzzy set*

The delta geyser temperature Fuzzy set as illustrated in Gigure 4.3, with linguistic membership functions, were specified as:

- DropHigh;

  This membership function has a trapezium shape with *xy* coordinates {(-10, 0), (-10, 1), (-2, 1), (-1.1, 0)}.

- DropMedium;

  This membership function also has a trapezium shape with *xy* coordinates {(-2, 0), (-1.1, 1), (-0.4, 1), (-0.2, 0)}.

- DropLow;

  This membership function has a triangle shape with *xy* coordinates {(-0.4, 0), (-0.2, 1), (0, 0)}.

- Normal;

  This membership function also has a triangle shape with *xy* coordinates

  {(-0.3, 0), (0, 1), (0.2, 0)}.

- Heatup.

  The last membership function has a trapezium shape with *xy* coordinates

  {(0, 0), (0.2, 1), (10, 1), (10, 0)}.

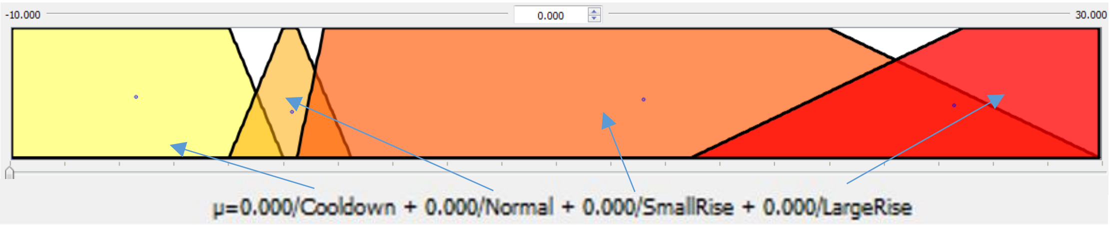


Figure 4.4 DeltaPipeTemp Fuzzy set

The delta hot water pipe temperature Fuzzy set is illustrated in Figure 4.4, with linguistic membership functions specified as:

- Cooldown

  This membership function has a trapezium shape with *xy* coordinates

  {(-10, 0), (-10, 1), (-2, 1), (0, 0)}.

- Normal

  This membership function has a trapezium shape with *xy* coordinates

  {(-2, 0), (0, 1), (0.5, 1), (2.5, 0)}.

- SmallRise

  This membership function also has a trapezium shape with *xy* coordinates

  {(0.5, 0), (1.5, 1), (20, 1), (30, 0)}.

- LargeRise

  The last membership function also has a trapezium shape with *xy* coordinates

  {(15, 0), (25, 1), (30, 1), (30, 0)}.

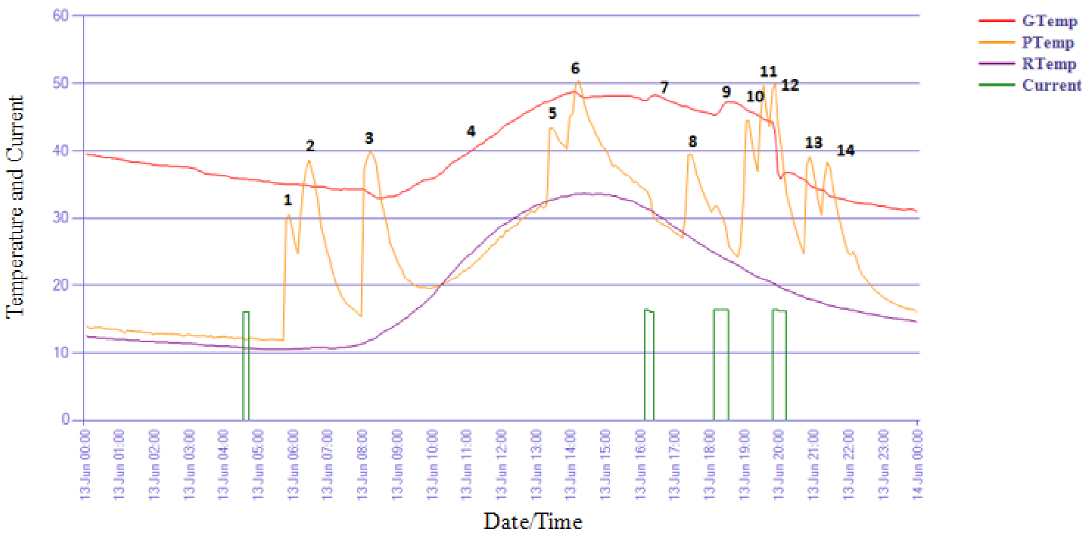


Figure 4.5 Geyser usage: 13 June 2017 for Temperatures and Current

On investigation, Figure 4.5 shows that there is a direct relationship between the delta in roof, pipe and geyser temperature. It illustrates that an increase in the outside temperature will increase the roof temperature and that the minimum pipe temperature will be almost the same as the roof temperature. The solar geyser influence is notable where the water temperature in the geyser increase with the almost the same factor as the roof temperature. It is also noticed that at times the hot water pipe temperature is higher than the geyser temperature. This is because of the stratification of the water at different levels, as discussed in section 2.5.3.

Investigating Table 4.1, the 0.2 degrees drop in water temperature was fuzzified as Use Small during the day and night, but if the decrease in external temperature were taken into account, it should be UseNone during the cooldown period at night.

Table 4.1 Geyser Usage: Roof Temperature influence

Date	Delta Geyser	Delta Pipe	Delta Roof	Expect	Fuzzy 4.0 Simulation
2017/06/13 08:10	-0.2	1.1	0.2	UseSmall	UseSmall
2017/06/13 19:40	-0.2	-2.3	-0.2	UseNone	UseSmall

Roof temperatures must be included in the Fuzzy Logic model to determine the difference between the water pipe temperature increase and cooldown effect from water usage or weather temperature changes. A third Fuzzy set, delta roof temperature was created as illustrated in Figure 4.6 with linguistic membership functions:

- Drop

  This membership function has a trapezium shape with *xy* coordinates

  {(-1, 0), (-1, 1), (-0.2, 1), (0, 0)}.

- Constant

  This membership function has a triangle shape with *xy* coordinates

  {(-0.2, 0), (0, 1), (0.2, 0)}.

- Rise

  This membership function has a trapezium shape with *xy* coordinates

  {(0, 0), (0.2, 1), (1, 1), (1, 0)}.

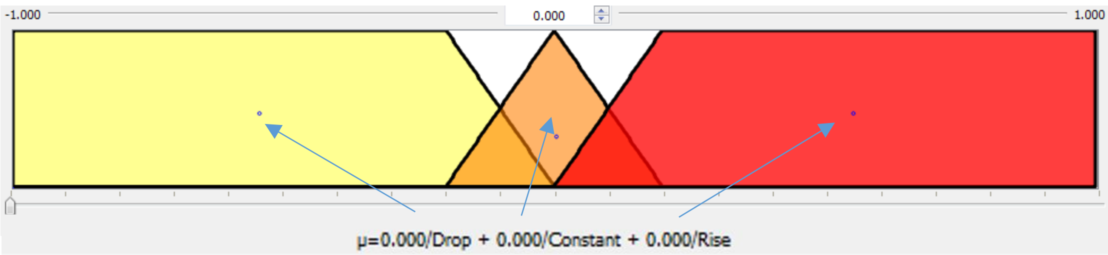


Figure 4.6 DeltaRoofTemp Fuzzy set

Table 4.2 *Fuzzy rules fuzzified roof, geyser and roof temperatures.*

Rule	DeltaRoof	And DeltaGeyser	And DeltaPipe	Usage
1	not Drop	Heatup	Cooldown	UseNone
2	not Drop	Heatup	Normal	UseNone
3	not Drop	Heatup	SmallRise	UseSmall
4	not Drop	Heatup	LargeRise	UseMedium
5	not Drop	Normal	Cooldown	UseNone
6	not Drop	Normal	Normal	UseNone
7	not Drop	Normal	SmallRise	UseSmall
8	not Drop	Normal	LargeRise	UseMedium
9	not Drop	DropLow	Cooldown	UseSmall
10	not Drop	DropLow	Normal	UseSmall
11	not Drop	DropLow	SmallRise	UseSmall
12	not Drop	DropLow	LargeRise	UseMedium
13	not Drop	DropMedium		UseMedium
14	not Drop	Heatup	Cooldown	UseNone
15	not Drop	Heatup	Normal	UseNone
16	not Drop	Heatup	SmallRise	UseSmall
17	not Drop	Heatup	LargeRise	UseMedium
18	Drop	Normal	Cooldown	UseNone
19	Drop	Normal	Normal	UseNone
20	Drop	Normal	SmallRise	UseSmall
21	Drop	Normal	LargeRise	UseMedium
22	Drop	DropLow	Cooldown	UseNone
23	Drop	DropLow	Normal	UseNone
24	Drop	DropLow	SmallRise	UseSmall
25	Drop	DropLow	LargeRise	UseSmall
26	Drop	DropMedium	Cooldown	UseSmall
27	Drop	DropMedium	Normal	UseSmall
28	Drop	DropMedium	SmallRise	UseMedium
29	Drop	DropMedium	LargeRise	UseMedium
30		DropHigh		UseLarge

Table 4.2 shown the thirty rules created from the Fuzzy sets covering all the options satisfactory for this purpose.

With simulations it was determined that Algebraic Product for T-Norm and the Largest of Maximum as S-Norm did produce the best fuzzified outcome.

To test the reliability of the fuzzy sets and rules two days, 13 June 2017 and 28 July 2017, were inspected. For each day, every incident that was noticeable on a graphical illustration was numbered as shown in Figure 4.5 and Figure 4.7.

Table 4.3 shows all the Fuzzy outcomes per five-minute intervals per incident numbered from 1 to 14, for 13 June 2017. Each incident is also discussed from the aspect of what a person would expect to investigate the graphics and raw data.

Table 4.3 *Geyser usage: 13 June 2017*

Date	Geyser Temp	Delta Geyser	Roof Temp	Delta Roof	Pipe Temp	Delta Pipe	Fuzzy	Graph Point	Expect
2017/06/13 05:35	35.1		10.5		11.9			1	Small rise in pipe temperature, geyser temperature stays constant. UseSmall
2017/06/13 05:40	35.1	0	10.5	0	11.8	-0.1	UseNone		
2017/06/13 05:45	35	-0.1	10.5	0	30	18.2	UseSmall		
2017/06/13 05:50	35	0	10.5	0	30.5	0.5	UseNone		
2017/06/13 05:55	35	0	10.6	0.1	28.7	-1.8	UseNone		
2017/06/13 06:00	35	0	10.6	0	26.2	-2.5	UseNone		
2017/06/13 06:05	35	0	10.6	0	24.8	-1.4	UseNone		
2017/06/13 06:10	34.9	-0.1	10.6	0	30.8	6	UseSmall	2	Larger rise in pipe temperature, but small drop in geyser temperature. UseSmall
2017/06/13 06:15	34.9	0	10.7	0.1	35.2	4.4	UseSmall		
2017/06/13 06:20	34.9	0	10.7	0	37.5	2.3	UseSmall		
2017/06/13 06:25	34.8	-0.1	10.7	0	38.6	3.4	UseSmall		
2017/06/13 06:30	34.7	-0.1	10.8	0.1	37	0.4	UseNone		
2017/06/13 06:35	34.7	0	10.8	0	35.1	-1.9	UseNone		
...									
2017/06/13 07:50	34.3		11.2		15.7			3	Large rise in pipe temperature, but notible drop in geyser temperature. UseMeduim
2017/06/13 07:55	34.3	0	11.3	0.1	15.4	-0.3	UseNone		
2017/06/13 08:00	34.3	0	11.4	0.1	37.3	21.9	UseSmall		
2017/06/13 08:05	33.9	-0.4	11.7	0.3	38.8	1.5	UseMeduim		
2017/06/13 08:10	33.7	-0.2	11.9	0.2	39.9	1.1	UseSmall		
2017/06/13 08:15	33.4	-0.3	12	0.1	39.4	-0.5	UseMedium		
2017/06/13 08:20	33.1	-0.3	12.2	0.2	38.3	-1.1	UseMedium		
2017/06/13 08:25	33	-0.1	12.6		35.3	-3	UseNone		
...									
2017/06/13 11:00	39.7		24.5		22.6			4	Hot day , all temperatures are rising. UseNone
2017/06/13 11:05	40	0.3	24.7	0.2	22.8	0.2	UseNone		
2017/06/13 11:10	40.3	0.3	25.2	0.5	23.3	0.5	UseNone		
2017/06/13 11:15	40.7	0.4	25.7	0.5	23.7	0.4	UseNone		
...									

Table 4.3 continuous *Geyser usage: 13 June 2017*

Date	Geyser Temp	Delta Geyser	Roof Temp	Delta Roof	Pipe Temp	Delta Pipe	Fuzzy	Graph Point	Expect
2017/06/13 11:00	39.7		24.5		22.6			4	Hot day , all temperatures are rising. UseNone
2017/06/13 11:05	40	0.3	24.7	0.2	22.8	0.2	UseNone		
2017/06/13 11:10	40.3	0.3	25.2	0.5	23.3	0.5	UseNone		
2017/06/13 11:15	40.7	0.4	25.7	0.5	23.7	0.4	UseNone		
...									
2017/06/13 13:05	46.8		32		31.8			5	Small rise in pipe temperature, geyser temperature is also rising. UseSmall
2017/06/13 13:10	47.2	0.4	32.4	0.4	31.5	-0.3	UseNone		
2017/06/13 13:15	47.2	0	32.4	0	31.9	0.4	UseNone		
2017/06/13 13:20	47.4	0.2	32.7	0.3	43.3	11.4	UseSmall		
2017/06/13 13:25	47.6	0.2	32.7	0	43.3	0	UseNone		
...									ddd
2017/06/13 13:45	48.4		33.2		40.9			6	Large rise in pipe temperature, notible drop in geyser temperature. UseMeduim
2017/06/13 13:50	48.5	0.1	33.2	0	40.3	-0.6	UseNone		
2017/06/13 13:55	48.6	0.1	33.4	0.2	45.1	4.8	UseSmall		
2017/06/13 14:00	48.8	0.2	33.5	0.1	45.5	0.4	UseNone		
2017/06/13 14:05	48.8	0	33.6	0.1	49.7	4.2	UseSmall		
2017/06/13 14:10	48.4	-0.4	33.6	0	50.4	0.7	UseMedium		
2017/06/13 14:15	48.1	-0.3	33.6	0	49.1	-1.3	UseMedium		
2017/06/13 14:20	47.9	-0.2	33.7	0.1	47.2	-1.9	UseSmall		
2017/06/13 14:20	47.9	0	33.6	-0.1	45.6	-1.6	UseNone		
...									
2017/06/13 16:00	47.6		31.7		34.5			7	Geyser temperature rises sharp, element did switched on. UseNone
2017/06/13 16:05	47.5	-0.1	31.6	-0.1	34.3	-0.2	UseNone		
2017/06/13 16:10	47.6	0.1	31.3	-0.3	33.9	-0.4	UseNone		
2017/06/13 16:15	48.1	0.5	31.2	-0.1	32.7	-1.2	UseNone		
2017/06/13 16:20	48.3	0.2	30.9	-0.3	30	-2.7	UseNone		
2017/06/13 16:25	48.3	0	30.5	-0.4	29.7	-0.3	UseNone		
...									
2017/06/13 17:05	46.8		28.2		27.4			8	Outside temperature is dropping, small increase in pipe temperature. UseSmall
2017/06/13 17:10	46.6	-0.2	27.9	-0.3	27.1	-0.3	UseNone		
2017/06/13 17:15	46.6	0	27.6	-0.3	31	3.9	UseSmall		
2017/06/13 17:20	46.5	-0.1	27.4	-0.2	39.4	8.4	UseSmall		
2017/06/13 17:25	46.2	-0.3	27	-0.4	39.5	0.1	UseSmall		
2017/06/13 17:30	46.1	-0.1	26,8	-022	38.2	-1.3	UseNone		
...									

Table 4.3 continuous *Geyser usage: 13 June 2017*

Date	Geyser Temp	Delta Geyser	Roof Temp	Delta Roof	Pipe Temp	Delta Pipe	Fuzzy	Graph Point	Expect
2017/06/13 17:05	46.8		28.2		27.4			8	Outside temperature is dropping, small increase in pipe temperature. UseSmall
2017/06/13 17:10	46.6	-0.2	27.9	-0.3	27.1	-0.3	UseNone		
2017/06/13 17:15	46.6	0	27.6	-0.3	31	3.9	UseSmall		
2017/06/13 17:20	46.5	-0.1	27.4	-0.2	39.4	8.4	UseSmall		
2017/06/13 17:25	46.2	-0.3	27	-0.4	39.5	0.1	UseSmall		
2017/06/13 17:30	46.1	-0.1	26,8	-022	38.2	-1.3	UseNone		
...									
2017/06/13 18:00	45.5		25.1		30.9			9	Geyser temperature. rises sharp, element did switched on. UseNone
2017/06/13 18:05	45.3	-0.2	24.9	-0.2	31.7	0.8	UseNone		
2017/06/13 18:10	45.5	0.2	24.6	-0.3	31.7	0	UseNone		
2017/06/13 18:15	46	0.5	24.4	-0.2	30.7	-1	UseNone		
2017/06/13 18:20	46.7	0.7	24.1	-0.3	30.1	-0.6	UseNone		
2017/06/13 18:25	47.2	0.5	23.9	-0.2	28.7	-1.4	UseNone		
2017/06/13 18:30	47.3	0.1	23.7	-0.2	25.9	-2.8	UseNone		
2017/06/13 18:35	47.2	-0.1	23.5	-0.2	25.3	-0.6	UseNone		
2017/06/13 18:40	47.2	0	23.3	-0.2	24.7	-0.6	UseNone		
2017/06/13 18:45	47	-0.2	23	-0.3	24.2	-0.5	UseNone	10	Small rise in pipe temperature, notible drop in geyser temperature. UseMeduim
2017/06/13 18:50	46.8	-0.2	22.8	-0.2	26	1.8	UseSmall		
2017/06/13 18:55	46.5	-0.3	22.5	-0.3	32.9	6.9	UseMedium		
2017/06/13 19:00	46.1	-0.4	22.2	-0.3	44.5	11.6	UseMedium		
2017/06/13 19:05	45.9	-0.2	22	-0.2	44.5	0	UseNone		
2017/06/13 19:10	45.7	-0.2	21.8	-0.2	41.4	-3.1	UseNone		
2017/06/13 19:15	45.6	-0.1	21.5	-0.3	38.9	-2.5	UseNone		
2017/06/13 19:20	45.3	-0.3	21.3	-0.2	37	-1.9	UseSmall		
2017/06/13 19:25	45.1	-0.2	21.1	-0.2	45	8	UseSmall	11	Large rise in pipe temperature, geyser temperature is droping UseMeduim
2017/06/13 19:30	44.7	-0.4	20.9	-0.2	49.7	4.7	UseMedium		
2017/06/13 19:35	44.6	-0.1	20.8	-0.1	45.9	-3.8	UseNone		
2017/06/13 19:40	44.4	-0.2	20.6	-0.2	43.6	-2.3	UseNone		

Table 4.3 continuous *Geyser usage: 13 June 2017*

Date	Geyser Temp	Delta Geyser	Roof Temp	Delta Roof	Pipe Temp	Delta Pipe	Fuzzy	Graph Point	Expect
2017/06/13 19:45	44.3	-0.1	20.4	-0.2	49	5.4	UseSmall	12	Large drop in geyser temperature and pipe temperature is rising. UseLarge
2017/06/13 19:50	42.7	-1.6	20.2	-0.2	50	1	UseLarge		
2017/06/13 19:55	36.8	-5.9	19.9	-0.3	44.1	-5.9	UseLarge		
2017/06/13 21:00	35.8	-1	19.7	-0.2	41.2	-2.9	UseSmall		
2017/06/13 20:05	36.7	0.9	19.5	-0.2	37.5	-3.7	UseNone		
2017/06/13 20:10	36.8	0.1	19.3	-0.2	33.6	-3.9	UseNone		
2017/06/13 20:15	36.8	0	19.2	-0.1	31.9	-1.7	UseNone		
2017/06/13 20:20	36.7	-0.1	19	-0.2	30.2	-1.7	UseNone		
2017/06/13 20:25	36.5	-0.2	18.8	-0.2	28.5	-1.7	UseNone		
2017/06/13 20:30	36.1	-0.4	18.6	-0.2	27.1	-1.4	UseSmall		
2017/06/13 20:35	35.9	-0.2	18.5	-0.1	25.8	-1.3	UseNone		
2017/06/13 20:40	35.7	-0.2	18.3	-0.2	24.7	-1.1	UseNone		
2017/06/13 20:45	35.5	-0.2	18.1	-0.2	37.9	13.2	UseSmall	13	Small rise in pipe temperature, notible drop in geyser temperature. UseMeduim
2017/06/13 20:50	34.9	-0.6	18	-0.1	39.1	1.2	UseMedium		
2017/06/13 20:55	34.6	-0.3	17.9	-0.1	37.7	-1.4	UseSmall		
2017/06/13 21:00	34.5	-0.1	17.8	-0.1	35.1	-2.6	UseNone		
2017/06/13 21:05	34.3	-0.2	17.6	-0.2	32.5	-2.6	UseNone		
2017/06/13 21:10	34.2	-0.1	17.5	-0.1	30.5	-2	UseNone		
2017/06/13 21:15	34.1	-0.1	17.3	-0.2	36	5.5	UseSmall	14	Small rise in pipe temperature, notible drop in geyser temperature. UseMeduim
2017/06/13 21:20	33.6	-0.5	17.1	-0.2	38.3	2.3	UseMedium		
2017/06/13 21:25	33.2	-0.4	17	-0.1	37.4	-0.9	UseMedium		
2017/06/13 21:30	33.2	0	16.9	-0.1	34.7	-2.7	UseNone		
2017/06/13 21:35	33.1	-0.1	16.8	-0.1	32.5	-2.2	UseNone		
2017/06/13 21:40	33	-0.1	16.7	-0.1	30.3	-2.2	UseNone		
2017/06/13 21:45	32.9	-0.1	16.6	-0.1	28.5	-1.8	UseNone		
2017/06/13 21:50	32.8	-0.1	16.6	0	26.6	-1.9	UseNone		
2017/06/13 21:55	32.6	-0.2	16.5	-0.1	25.1	-1.5	UseNone		
2017/06/13 22:00	32.5	-0.1	16.3	-0.2	24.5	-0.6	UseNone		

After investigating the temperature changes for 13 June 2017 for all geyser and pipe temperature changes, the fuzzy outcome was the same as the expected usage for all fourteen incidents.

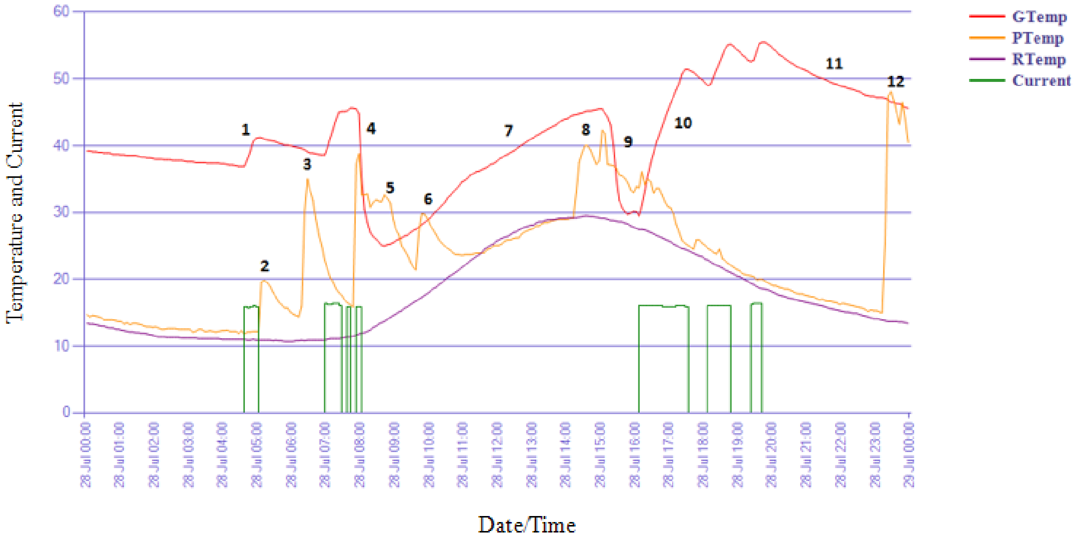


Figure 4.7 Geyser usage: 28 July 2017 for Temperatures and Current

Table 4.4 shows all the fuzzy outcomes per five-minute intervals per event numbered from 1 to 12, for 28 July 2017. Each incident is also discussed from the aspect of what a person would expect to investigate the graphics and raw data.

Table 4.4 Geyser usage: 28 July 2017

Date	Geyser Temp	Delta Geyser	Roof Temp	Delta Roof	Pipe Temp	Delta Pipe	Fuzzy	Graph Point	Expect
2017/07/28 04:30	36.9		11		12.3			1	Sharp rise in Geyser temperature, element did switch on. UseNone
2017/07/28 04:35	36.9	0	11	0	11.8	-0.5	UseNone		
2017/07/28 04:40	37.9	1	10.9	-0.1	12.1	0.3	UseNone		
2017/07/28 04:45	39.1	1.2	10.9	0	12.1	0	UseNone		
2017/07/28 04:50	40.6	1.5	11	0.1	12.2	0.1	UseNone		
2017/07/28 04:55	41.1	0.5	10.9	-0.1	12.2	0	UseNone		
2017/07/28 05:00	41.2	0.1	10.9	0	12.2	0	UseNone		
2017/07/28 05:05	41.2	0	10.9	0	19.6	7.4	UseSmall	2	Only small rise in pipe temperature, geyser temperature is constant. UseSmall
2017/07/28 05:10	41	-0.2	10.9	0	19.8	0.2	UseSmall		
2017/07/28 05:15	40.9	-0.2	10.9	0	19.5	-0.3	UseNone		
2017/07/28 05:20	40.8	-0.1	10.9	0	18.9	-0.6	UseNone		
…									

Table 4.3 continuous *Geyser usage: 28 July 2017*

Date	Geyser Temp	Delta Geyser	Roof Temp	Delta Roof	Pipe Temp	Delta Pipe	Fuzzy	Graph Point	Expect
2017/07/28 06:05	39.8		10.8		15.6			3	Larger rise in pipe temperature, notible drop in geyser temperature. UseMeduim
2017/07/28 06:10	39.7	-0.1	10.8	0	15.4	-0.2	UseNone		
2017/07/28 06:15	39.6	-0.1	10.9	0.1	15.9	1.5	UseSmall		
2017/07/28 06:20	39.4	-0.2	10.8	-0.1	30	14.1	UseSmall		
2017/07/28 06:25	39.1	-0.3	10.9	0.1	35	5	UseMedium		
2017/07/28 06:30	38.9	-0.2	10.9	0	33.3	-1.7	UseSmall		
2017/07/28 06:35	38.9	0	10.9	0	31.8	-1.5	UseNone		
2017/07/28 07:40	45.6		11.4		16.2			4	Geyser element was switch on because water is to cold. Large drop after water reaches temperature. Hot water is used a few times. UseLarge
2017/07/28 07:45	45.6	0	11.5	0.1	15.8	0.4	UseNone		
2017/07/28 07:50	45.5	-0.1	11.6	0.1	37.2	21.4	UseSmall		
2017/07/28 07:55	44.8	-0.7	11.8	0.2	38.8	1.6	UseMedium		
2017/07/28 08:00	35.6	-9.2	11.9	0.1	32.6	-6.2	UseLarge		
2017/07/28 08:05	30.4	-5.2	12	0.1	32.6	0	UseLarge		
2017/07/28 08:10	28.5	-1.9	12.2	0.2	32.8	0.2	UseLarge		
2017/07/28 08:15	27	-1.5	12.4	0.2	30.8	-2	UseMedium		
2017/07/28 08:20	26.4	-0.6	12.8	0.4	31.5	0.4	UseMedium		
2017/07/28 08:25	26	-0.4	13	0.2	31.9	-.4	UseMedium		
2017/07/28 08:30	25.6	-0.4	13.3	0.3	31.7	-0.2	UseMedium		
2017/07/28 08:35	25.1	-0.5	13.5	0.2	31.6	-0.1	UseMedium		
2017/07/28 08:40	25	-0.1	13.7	0.2	32.6	1	UseSmall		
2017/07/28 08:45	25.2	0.2	14	0.3	32.1	-0.5	UseNone		
2017/07/28 08:50	25.3	0.1	14.3	0.3	31.4	-0.7	UseNone		
2017/07/28 08:55	25.5	0.2	14.6	0.3	28.9	-2.5	UseNone	5	Hot water pipe coles down and geyser water temperature Increases. UseNone
2017/07/28 09:00	25.8	0.3	14.8	0.2	27.4	-1.5	UseNone		
2017/07/28 09:05	26	0.2	15.1	0.3	26.7	-0.7	UseNone		
2017/07/28 09:10	26.2	-0.2	15.3	0.2	25	-1.7	UseNone		
2017/07/28 09:15	26.5	0.3	15.7	0.4	24.3	-0.7	UseNone		
2017/07/28 09:20	26.6	0.1	15.9	0.2	23.6	-0.7	UseNone		
2017/07/28 09:25	27	0.4	16.2	0.3	22.6	-1	UseNone		
2017/07/28 09:30	27.3	0.3	16.5	0.3	21.8	-0.8	UseNone		
2017/07/28 09:35	27.5	0.2	16.7	0.2	21.4	-0.4	UseNone		
2017/07/28 09:40	27.8	0.3	17	0.3	27.1	5.7	UseSmall	6	Only small rise in pipe temperature, geyser temperature is constant. UseSmall
2017/07/28 09:45	28.2	0.4	17.3	0.3	29.8	2.7	UseSmall		
2017/07/28 09:50	28.5	0.3	17.6	0.3	29.7	-0.1	UseNone		
2017/07/28 09:55	28.9	0.4	17.9	0.3	29	-0.4	UseNone		
2017/07/28 10:00	29.2	0.3	18.2	0.3	28.1	-0.9	UseNone		
...									

Table 4.4 continuous *Geyser usage: 28 July 2017*

Date	Geyser Temp	Delta Geyser	Roof Temp	Delta Roof	Pipe Temp	Delta Pipe	Fuzzy	Graph Point	Expect
2017/07/28 12:00	38		25.9		25.1			7	Outside, pipe and geyser temperature is increasing. UseNone
2017/07/28 12:05	38.2	0.2	26.1	0.2	25.3	0.2	UseNone		
2017/07/28 12:10	38.5	0.3	26.3	0.2	25.7	0.4	UseNone		
2017/07/28 12:15	38.7	0.2	26.4	0.1	25.9	0.2	UseNone		
2017/07/28 12:20	39	0.3	26.7	0.3	25.9	0	UseNone		
2017/07/28 12:25	39.2	0.2	27	0.3	26.1	0.2	UseNone		
2017/07/28 12:30	39.5	0.3	27.1	0.1	26.2	0.1	UseNone		
2017/07/28 12:35	39.8	0.3	27.4	0.3	26.2	0	UseNone		
2017/07/28 12:40	40.1	0.3	27.6	0.2	26.8	0.6	UseNone		
2017/07/28 12:45	40.4	0.3	27.8	0.2	27.1	0.3	UseNone		
2017/07/28 12:50	40.7	0.3	27.9	0.1	27.3	0.2	UseNone		
2017/07/28 12:55	41	0.3	28.1	0.2	27.5	0.2	UseNone		
...									
2017/07/28 14:00	44.1		29.2		29			8	Notible rise in pipe temperature, geyser temperature starts slowly to drop. UseMeduim
2017/07/28 14:05	44.4	0.3	29.2	0	29.1	0.1	UseNone		
2017/07/28 14:10	44.6	0.2	29.2	0	29.1	0	UseNone		
2017/07/28 14:15	44.7	0.1	29.2	0	33	3.9	UseSmall		
2017/07/28 14:20	44.8	0.1	29.3	0.1	37.5	4.5	UseSmall		
2017/07/28 14:25	44.9	0.1	29.4	0.1	39	1.5	UseSmall		
2017/07/28 14:30	45.1	0.2	29.5	0.1	40	1	UseNone		
2017/07/28 14:35	45.2	0.1	29.5	0	40	0	UseNone		
2017/07/28 14:40	45.2	0	29.4	-0.1	39.4	-0.6	UseNone		
2017/07/28 14:45	45.3	0.1	29.4	0	38.3	-1.1	UseNone	8	
2017/07/28 14:50	45.4	0.1	29.3	-0.1	37.2	-1.1	UseNone		
2017/07/28 14:55	45.5	0.1	29.2	-0.1	37.7	0.5	UseNone		
2017/07/28 15:00	45.5	0	29.1	-0.1	42.3	4.6	UseSmall		
2017/07/28 15:05	44.9	-0.6	29.1	0	41.9	-0.4	UseMedium		
2017/07/28 15:10	44.3	-0.6	29	-0.1	37.2	-4.7	UseMedium		
2017/07/28 15:15	43.1	-1.2	28.8	-0.2	37.1	-0.1	UseSmall	9	Sharp drop in gerser temperature. UseLarge
2017/07/28 15:20	40.1	-3	28.8	0	37	-0.1	UseLarge		
2017/07/28 15:25	34.9	-5.2	28.7	-0.1	36.6	-0.4	UseLarge		
2017/07/28 15:30	31.8	-3.1	28.6	-0.1	35.6	-1	UseLarge		
2017/07/28 15:35	30.8	-1	28.6	0	35.5	-0.1	UseMedium		
2017/07/28 15:40	30.1	-0.7	28.4	-0.2	35.1	0.4	UseSmall		
...									

Table 4.4 continuous *Geyser usage: 28 July 2017*

Date	Geyser Temp	Delta Geyser	Roof Temp	Delta Roof	Pipe Temp	Delta Pipe	Fuzzy	Graph Point	Expect
2017/07/28 17:00	46.4		25.6		30.7			10	Small increase in pipe temperature, geyser temperature stays constant. UseSmall
2017/07/28 17:05	47.4	1	25.3	-0.3	29.8	-0.9	UseNone		
2017/07/28 17:10	48.4	1	25.1	-0.2	28.2	-1.6	UseNone		
2017/07/28 17:15	49.4	1	24.8	-0.3	27	-1.2	UseNone		
2017/07/28 17:20	50.8	1.4	24.6	-0.2	25.8	-1.2	UseNone		
2017/07/28 17:25	51.4	0.6	24.5	-0.1	25.4	-0.4	UseNone		
2017/07/28 17:30	51.4	0	24.3	-0.2	25.1	-0.3	UseNone		
2017/07/28 17:35	51.2	-0.2	24.1	-0.2	24.8	-0.3	UseNone		
2017/07/28 17:40	51	-0.2	23.9	-0.2	24.5	-0.3	UseNone		
2017/07/28 17:45	50.6	-0.4	23.7	-0.2	25.9	1.4	UseMeduim		
2017/07/28 17:50	50.2	-0.4	23.5	-0.2	25.9	0	UseSmall		
2017/07/28 17:55	49.9	-0.3	23.4	-0.1	25.4	-0.5	UseSmall		
2017/07/28 18:00	49.4	-0.5	23	-0.4	25.1	-0.3	UseSmall		
...									
2017/07/28 22:30	14.8		15.4		16.4		UseNone	11	Outside, pipe and geyser temperature is decreasing. UseNone
2017/07/28 22:35	14.7	-0.1	14.9	-0.5	16.2	-0.4	UseNone		
2017/07/28 22:40	47.6	-0.1	14.5	-0.4	15.5	-0.7	UseNone		
2017/07/28 22:45	47.5	-0.1	14.4	-0.1	15.2	-0.3	UseNone		
2017/07/28 22:50	47.4	-0.1	14.2	-0.2	15.4	0.2	UseNone		
2017/07/28 22:55	47.4	0	14.1	-0.1	15.3	-0.1	UseNone		
2017/07/28 23:00	47.2	-0.2	14.1	0	15.3	0	UseSmall	12	Notible increase in pipe temp. UseMeduim
2017/07/28 23:05	47.2	0	14	-0.1	15.1	-0.2	UseNone		
2017/07/28 23:10	47.2	0	13.9	-0.1	15	-0.1	UseNone		
2017/07/28 23:15	47.1	-0.1	13.8	-0.1	25.9	10.9	UseSmall		
2017/07/28 23:20	46.9	-0.2	13.7	-0.1	47.4	21.5	UseSmall		
2017/07/28 23:25	46.5	-0.4	13.7	0	48.1	0.7	UseMedium		
2017/07/28 23:30	46.5	0	13.7	0	46.7	-1.4	UseNone		
2017/07/28 23:35	46.3	-0.2	13.7	0	45	-1.7	UseSmall		
2017/07/28 23:40	46.3	0	13.6	-0.1	43.2	-1.8	UseNone		
2017/07/28 23:45	46	-0.3	13.6	0	46.5	3.3	UseMedium		
2017/07/28 23:50	45.7	-0.3	13.5	-0.1	43.9	-2.6	UseSmall		

After investigating the temperature changes for 28 July 2017 for all geyser and pipe temperature changes, the fuzzy outcome was as expected for eleven of the twelfth events.

After the successful simulations, the modules were built into the home controller. Figure 4.8 illustrates the flowchart to convert the membership function shapes in straight lines to be used when the delta temperatures are fuzzified.

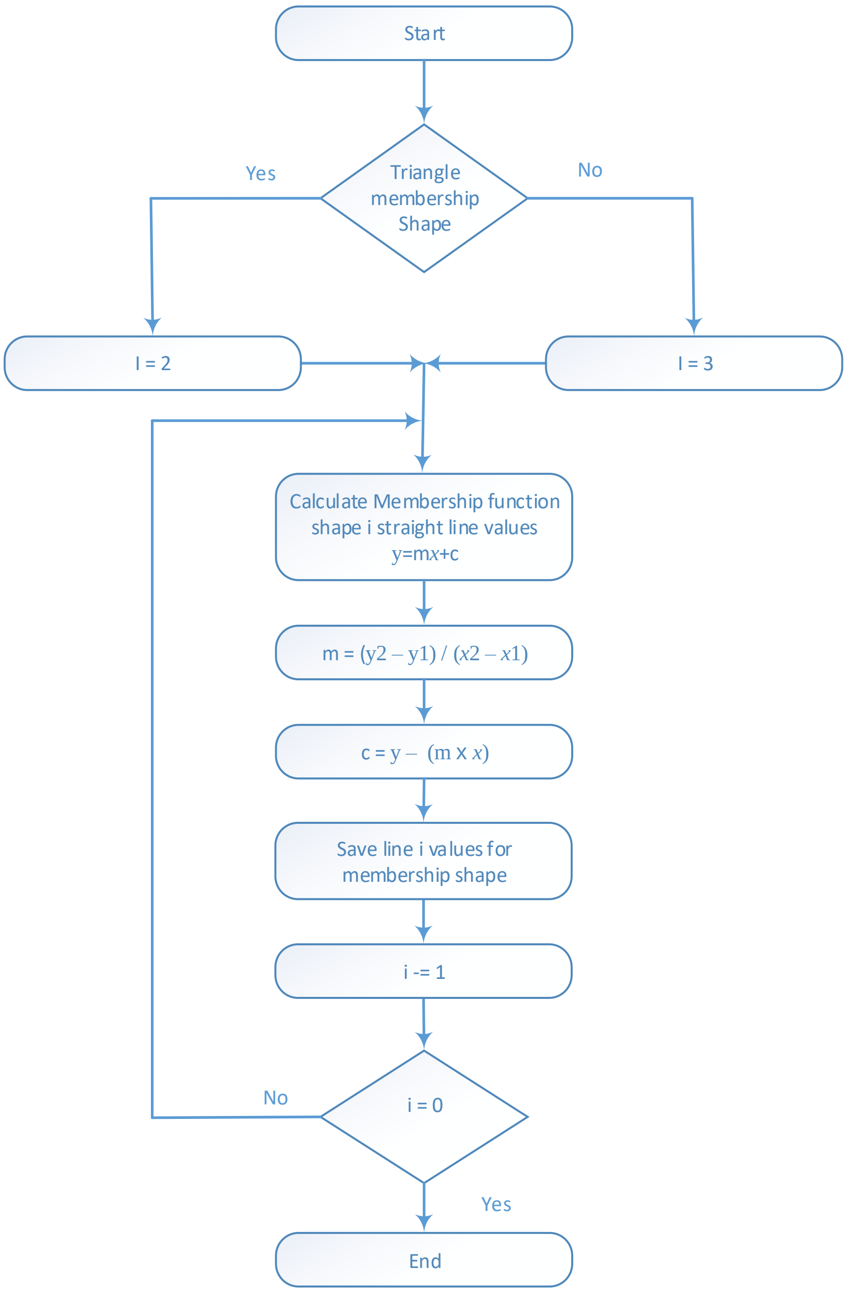


Figure 4.8 Calculate gradient of each line of membership function shape

Figure 4.9 illustrates the flowchart to fuzzify, inference and defuzzify the roof, geyser and pipe temperature at each five-minute interval, to give a *None*, *Low*, *Medium* or *Large* output.

Before fuzzification could start, the previous hour;s recorded geyser, hot water pipe, roof temperatures and the membership function values illustrated in Figure 4.3, Figure 4.4 and Figure 4.6 were loaded. The difference between each five-minute interval is recorded for each temperature sensor. To fuzzify each delta temperature the y value was calculated for the fuzzy set for each membership function.

At the inference stage the T-norm, also known as Intersection, as discussed in section 2.1.3.3 was used. For the application it was decided to investigate the output between the Algebraic Product and the Bounded Product as illustrated in Equation 2.22 and 2.23 and in Example 4.1.

For the defuzzification at the S-norm stage, also known as Union, the Largest of Maximum was used, as discussed in section 2.1.3.2. It was then decided that the Largest of Maximum was the best suited defuzzification strategy because in this design the NoUsage value was 0 and the highest usage 1. The highest usage wouldl be the output of the defuzzification process as shown in Example 4.1. Lastly, the controller wouldl record the largest five-minute usage per hour as that hourly usage

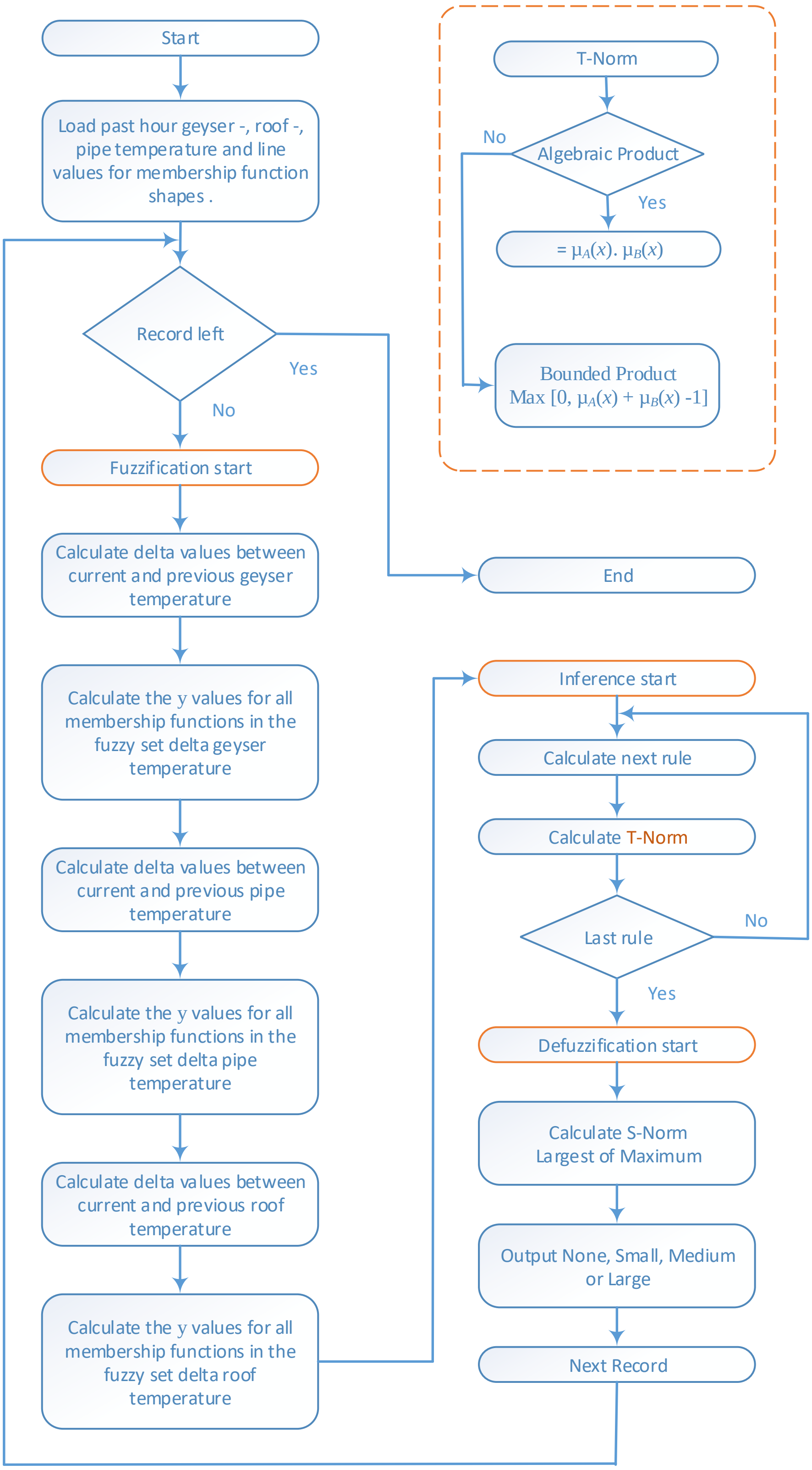


Figure 4.9 *Fuzzified geyser, roof and pipe temperatures*

The membership functions were recreated from the Fuzzylite simulations. Figure 4.10 shows the membership function per Fuzzy set as built on the home controller. The window illustrates the parameters graphically in triangle or trapezium shapes. The user can alter the membership shape parameters, click on the zoom button to zoom in or out of the diagrams. This was done to get a better view where the lines are to near each other, for example the Drop Low and Normal membership functions which ranges are much smaller, between -0.3 and 0.2 and -0.4 and 0, than the other in the Geyser Temperature Fuzzy set as shown in Figure 4.11.

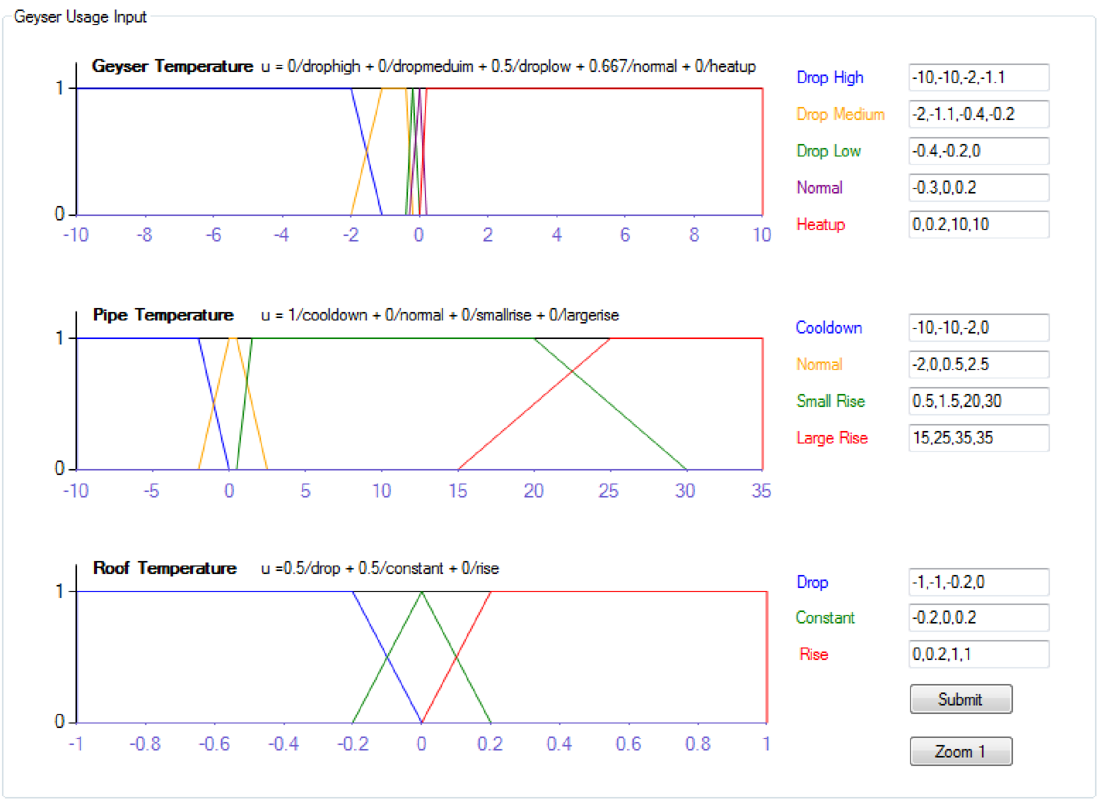


Figure 4.10 Membership functions for geyser-, pipe- and roof temperature fuzzy sets

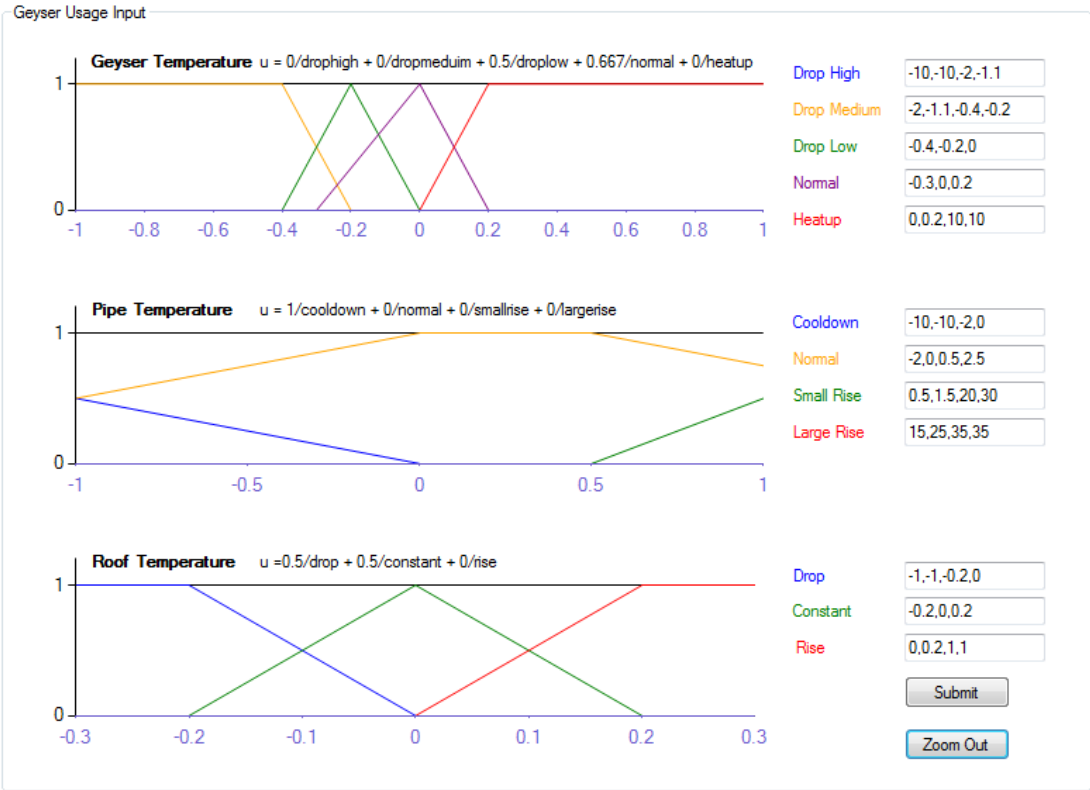


Figure 4.11 Zoomed membership functions for geyser used fuzzy sets

Example 4.1 illustrates the mathematical process to retrieve a crisp value from the Fuzzification of the geyser-related temperatures.

Example 4.1

On 13 June 2017 at 21h00 the difference in temperatures where:

Delta Geyser = -0.1°C, (4.1)

Delta Pipe = -2.6°C, (4.2)

Delta Roof = -0.1°C as retrieved from Table 4.3. (4.3)

The Algebraic Product was chosen for the T-Norm and the Largest of Maximum for the S-Norm parameter.

The **first step** the controller executess after the membership function coordinates is submitted, is to calculate the equation for each line. For this example, it will only be done for the lines where the input values cross that line.

Fuzzified Delta Geyser Temperature (4.1)

-0.1 crosses membership *Drop Low* line with *my* coordinates (-0.2,1) and (0,0) and membership function *Normal* line with coordinates (-0.3, 0) and (0,1).

Drop Low is calculated as:

$$m = (y2 - y1) / (x2 - x1)$$

$$= (0 - 1) / (0 - (-0.2))$$

$$= -5 \tag{4.4}$$

$$c = y - (m \times x)$$

$$= 1 - (-5 \times -0.2)$$

$$= 0 \tag{4.5}$$

Normal is calculated the same as in 4.4 and 4.5 and is:

$$m = 3.334 \tag{4.6}$$

$$c = 1 \tag{4.7}$$

μGeyserTemperature is calculated as follows

Insert (4.4) to (4.7) into (4.8) for (4.1)

$$\mu GeyserTemperature = 0/drophigh + 0/dropmeduim + (mx+c)/droplow + (mx+c)/normal + 0/heatup \quad (4.8)$$

$$= 0/drophigh + 0/dropmeduim + (-5)(-0.1)+0/droplow + (3.334)(-0.1)+1/normal + 0/heatup$$

$$= 0/drophigh + 0/dropmeduim + 0.5/droplow + 0.666/normal + 0/heatup \quad (4.9)$$

The answer to (4.9) is the same as μGeyserTemperature calculated by the controller as illustrated in Figure 4.11. The same processes can be repeated to calculate μPipeTemperature (4.10) for (4.2) and μRoofTemperature (4.11) for (4.3) as shown in Figure 4.11 [41, p.1310].

$$\mu PipeTemperature = 1/cooldown + 0/normal + 0/smallrise + 0/largerise \quad (4.10)$$

$$\mu RoofTemperature = 0.5/drop + 0.5/constant + 0/rise \quad (4.11)$$

The **second step** is the inference phase, where the controller calculates the T-Norm using the algebraic product for each rule. For simplicity the rules were written using the first character of the Fuzzy set in small letters and the membership function in capital letters, for example g=N means μGeyserTemperature = Normal.

Figure 4.12 shown that the controller did calculate values bigger than zero for rule 5, 9, 18 and 22. To better understand the example, only the calculation for rule 9 in (4.12) and rule 14 in (4.13) will be shown [41, p.1311].

Rule 9: r<>D and g=L and p=C then S

The summarised rule is written as:

if DeltaRoofTemp is not Drop and DeltaGeyserTemp is DropLow and DeltaPipeTemp is Cooldown then Usage is UseSmall

$$\text{T-Norm}_9 = \mu_{DeltaRoofTempNotDrop}(1-0.5).\mu_{DeltaGeyserTempDropLow}(0.5).\mu_{DeltaPipeTempCooldown}(1)$$

$$= 0.25 \tag{4.12}$$

Rule 14: r = D and g=H and p=C then N

The summarised rule is written as:

if DeltaRoofTemp is Drop and DeltaGeyserTemp is Heatup and DeltaPipeTemp is Cooldown then Usage is None

$$\text{T-Norm}_{14} = \mu_{DeltaRoofTempDrop}(0.5).\mu_{DeltaGeyserTempHeatup}(0).\mu_{DeltaPipeTempCooldown}(1)$$

$$= 0 \tag{4.13}$$

The **last step** is the defuzzification of the hot water usage. The selected S-Norm is the Largest of maximum where the sum of all usages is calculated [41, p.1312].

$$\text{S-Norm}_{None} = 0.3335 + 0.3335 + 0.25 = 0.917 \tag{4.14}$$

$$\text{S-Norm}_{Small} = 0.25 \tag{4.15}$$

$$\text{S-Norm}_{Meduim} = 0 \tag{4.16}$$

$$\text{S-Norm}_{Large} = 0 \tag{4.17}$$

The largest usage is (4.14) and the final usage is None. The calculation done between (4.12) and (4.17) can be verified by the controller shown in Figure 4.12.

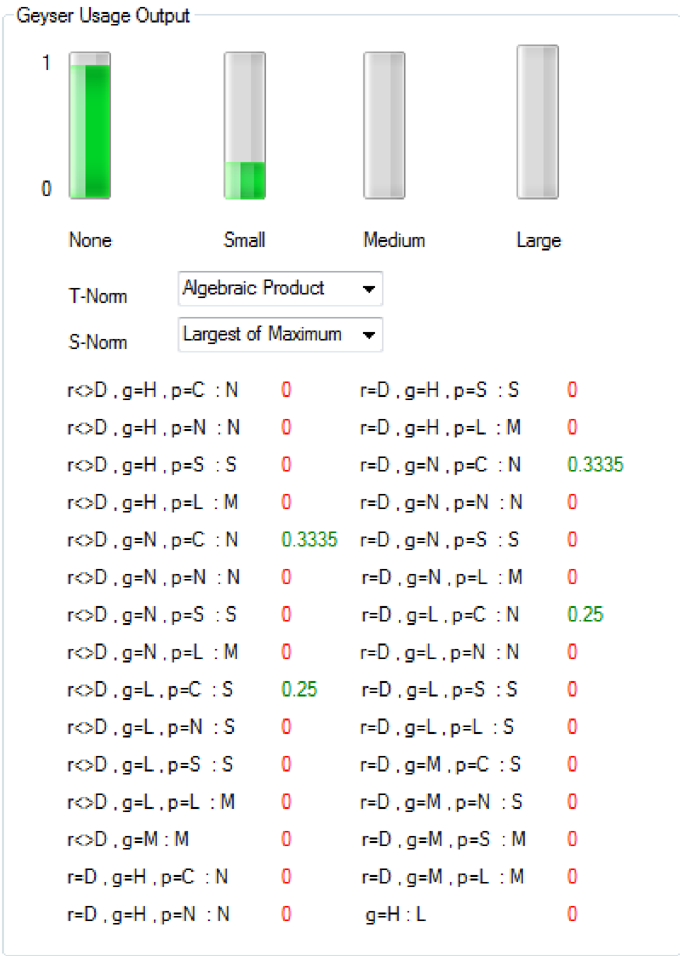


Figure 4.12 Geyser used fuzzy rules and defuzzification

The controller will calculate the highest five minute Fuzzification usage per hour. The highest usage was recorded between 20h46 and 20h50 as *Meduim. Meduim usage will be recorded* for that hour as illustrated in Figure 4.13.

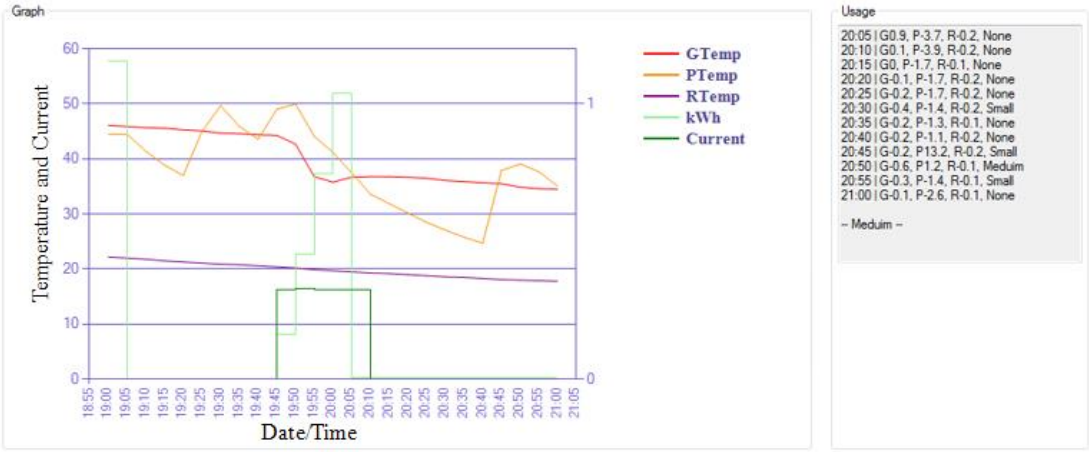


Figure 4.13 *Fuzzification Usage of 13 June 2017 between 20h05 and 21h00*

Figure 4.14 illustrates the overlay of the fuzzified usage against the geyser, pipe and roof temperatures. To illustrate the fuzzified usage graphical a numeric value of 0 for None, 0.334 for Small, 0.667 for Medium and 1 for Large usage

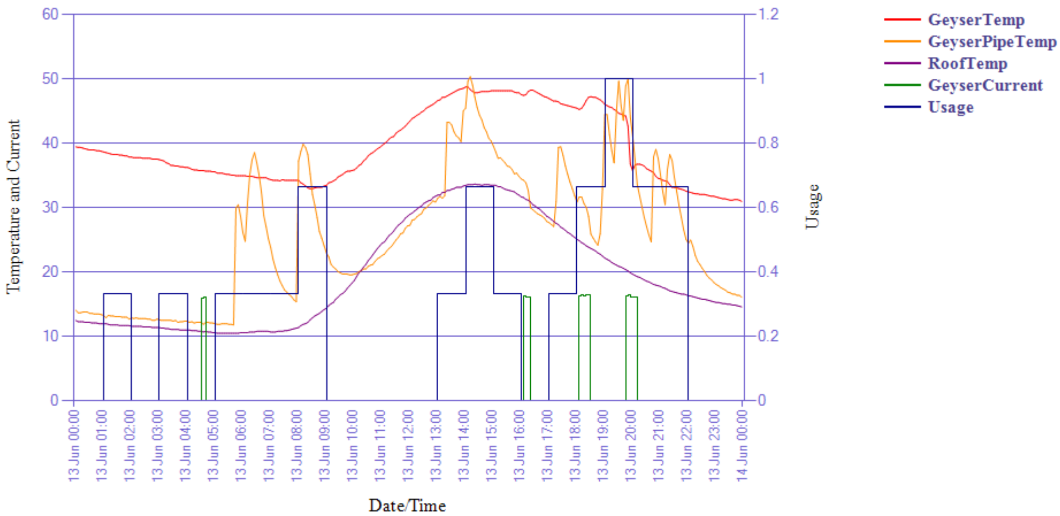


Figure 4.14 *Geyser usage: 13 June 2017 for Temperatures and Fuzzification*

4.2 *Geyser heat loses and gain*

Figure 4.5 illustrates that between 0h00 to 4h00 and 22h00 to 23h59 the water temperature decreased with the absence of any water usage. The same results can be seen in Figure 4.7 between 0h00 to 4h00 and 20h00 to 23h59. The raw data is available in Appendix F and Appendix G.

To predict how to control a geyser more accurately, the first important factor is to determine when the hot water temperature will be at a certain time if no energy is added to the system.

The equation to calculate the losses in a geyser is [42, p.3]:

$$q_{loses} = \frac{T_h - T_{ambient}}{\Delta x / k + 1/h} \qquad (4.18)$$

where:

q_{loses} : Heat loss in W/m^2

T_h : Water temperature inside the HWC in °C

$T_{ambient}$: Air temperature outside the HWC in °C

Δx : Thickness of insulation in m

k : Termal conductivity in W/m.K

h : Surface heat transfer coefficient in W/m^2.K

For all calculations pipe losses are excluded. On the assumption that Δx, k and h is not known, it is important for further calculations that a replacement constantis calculated from recorded temperature decreases. To calculate the losses one needs to know the outside area of the geyser that is in contact with the outside air. Most geysers have a cylindrical form with the following dimensions as taken from the Kwikot Geysers pamphlet [43, p.3]:

Table 4.5 *Kwikot geyser specifications.*

Size (l)	**Raduis** (mm)	**Length** (mm)	**Rating** (kW)	**Area** (m^2)
100	240	990	2	1.885
150	269	1035	3	2.205
200	269	1325	4	2.695
250	269	1645	4	4.236

The surface area was calculated as:

Area of a cylinder = 2 x side panels + Round Area

$$= 2\pi r^2 + 2\pi rl \qquad (4.19)$$

Calculate Joules energy stored in a geyser [42, p.3]:

$$Q = mc\Delta T \qquad (4.20)$$

where

Q : Energy in Joules

m : Mass in liters of water

c : Specific heat of water which is 4180 Joules per liter (kg) per °C [44, p.143]

ΔT : Change in temperature in °C

To eliminate the drop in ambient temperature, samples were taken where the roof temperature did not change for multiple five-minute samples.

Between 00:30 to 03:30 on 24 March 2018 the roof temperature were 17.2 °C and the 200-litre geyser temperatures decrease from 47.3 °C to 46.5°C.

Q_{3hr} = 200kg * 4180 J/kg.K * (47.3°C – 46.5°C) substitute 4.20

Q_{1hr} = 668800 k / 3 = 222933J (4.21)

Q_{loses} = Heat loss * Area * Hour Substitute 4.21 in 4.18

$222933\ J = (47.3\ °C - 17.2\ °C) / (\Delta x / k + 1/h) * 2.695\ m^2$

$(\Delta x / k + 1/h) = 3.63873\ x10^{-4}\ °C.m^2/J$ (4.22)

$Q_{losses\ 1h} = (T_h - T_{ambient}) / (\Delta x / k + 1/h) * 2.695\ m^2 * 1$ substitude 4.22 in 4.18

$= (47.3°C - 17.2\ °C) / 3.63873\ x10^{-4}\ °C.m^2/J * 2.695\ m^2$

$= 222933J$ (4.23)

$Q = mc\Delta T$ Substitude 4.23 in 4.20

$222933J = 200kg * 4180J/kg.K * (47.3\ °C - T_{new})$

$T_{new} = 47.0\ °C$ (4.24)

The recorded geyser temperature one hour later at 01:30 were 47.2 °C and not the calculated 47.0°C.

If the calculations are repeated for a four-hour period, the following is achieved:

$Q_{losses\ 4h} = (47.3\ °C - 17.2\ °C) / 3.63873\ x10^{-4}\ °C.m^2/J * 2.695\ m^2 * 4$

$= 891726J$ (4.25)

$Q = mc\Delta T$ Substitude 4.25 in 4.20

$891726J = 200kg * 4180\ J/kg.K * (47.3\ °C - T_{new})$

$T_{new} = 46.2\ °C$ (4.26)

The recorded geyser temperature was recorded at 46.2 °C. Calculations are more accurate over a longer timespan.

Previous mathematical calculations proved to predict the water temperature successfully, but the system needs to incorporate the ambient temperature changes.

The time period to heat the water from *x* °C to *y* °C of a specific geyser is also important.

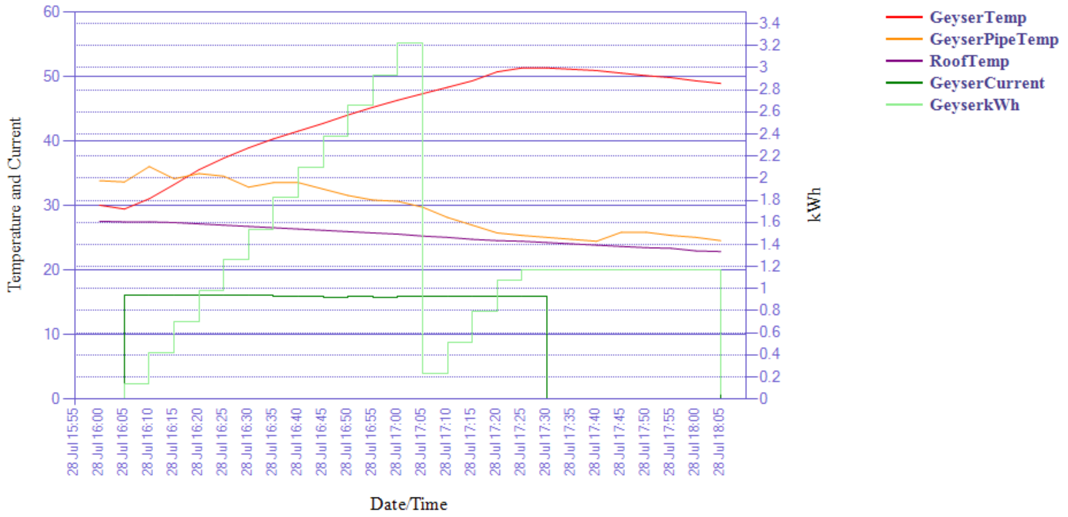


Figure 4.15　　*Geyser electrical activation: 28 July 2017 at 16:00*

Table 4.6　　*Geyser electrical activation: 28 July 2017 at 16:00*

Date Time	Geyser Temp	Roof Temp	Geyser Current	Geyser kWh
2017/07/28 16:05	29.5	27.5	16.1	0.13
2017/07/28 16:10	31.1	27.5	16.1	0.42
2017/07/28 16:15	33.3	27.5	16.1	0.7
2017/07/28 17:00	46.4	25.6	15.9	3.22
2017/07/28 17:20	50.8	24.6	16.0	1.07
2017/07/28 17:25	51.4	24.5	15.9	1.16

Figure 4.15 and Table 4.6 illustrates where the geyser switched on at 16h00 and off at 17h30 It should be noted that the entries indicate the past five minutes’ readings from the timestamp.

Table 4.6 illustrates that the geyser drawn 3.22 kWh the first hour and 1.16kWh the last hour to heat the 200*l* from 29.5°C to 51.4°C. To solve the problem Equation 4.20 was used:

$Q = mc\Delta T$

$Q = 200kg \times 4180\ J/kg.K \times (51.4°C - 29.5°C)$

$$Q = 18\ 308\ 400\ \text{Joules} \tag{4.27}$$

In theory the 4kW element can contribute 4000 Joules per second, but from measurements it can be seen that the element under investigation did deliver 3.22kWh. If the delta kWh per five minutes were calculated, it is observed that the geyser used 0.28 kWh, except when the geyser did switch on, it used 0.13kWh and 0.09kWh in the cycle when it switched off.

4.3 *Factors to take in account when predicting hot water requirements*

The predicting algorithm also needs to take the following factors into account:

4.3.1 Timeslot changing for hot water usage

Another energy loss is noted in Figure 4.7 between 16h00 and 20h00. The geyser element switches on three times, but there is no large hot water usage. That specific day, it has already been consumed at 08:00 and 15:20. In the above scenario all energy applied to the system was lost by heat losses and was never used.

4.3.2 **Solar thermal energy**

Using solar thermal energy to heat water introduces unknown factors. The energy inserted into the system is not a constant energy source as with an electrical element as shown in equation 4.20. When a domestic solar water heating system, with an electric backup element, is controlled with a time-based controller, it is unclear to which temperature it must heat the water. If the water is not heated enough in the morning, the hot water will be too cold for the entire day, but if the cloud cover is limited and it is a sunny day, the solar geyser will heat the water and the electrical energy injected into the solar geyser system would be unnecessary and wasted.

When the data was recorded, the temperature was set for:

- 4h30 – 6h00 to 40°C
- 16h00 – 18h00 to 50°C
- 18h00 – 20h00 to 55°C

as illustrated in Figure 4.16. Some overshooting of the temperature value was noted.

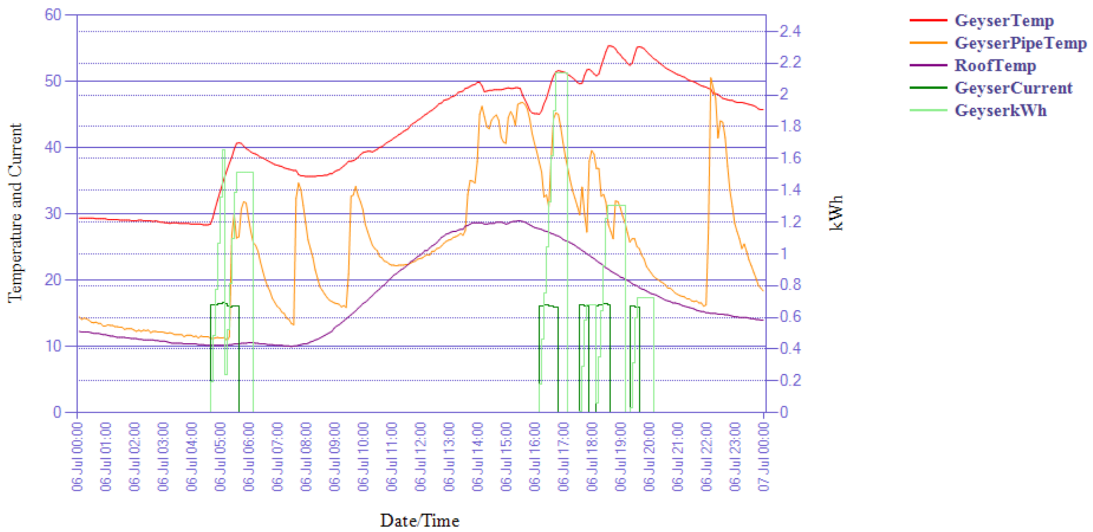


Figure 4.16 Geyserwise times and temperatures

The research done by Dintchev, described in section 2.5. indicated that in the summer months solar heating is efficient during the day and inefficient in winter months or cloudy and rainy days. This confirmed that if a controller has knowledge of the weather prediction, a more accurate decision can be made by the controller.

Weather prediction can be retrieved for free from openweathermap.org, as discussed in section 3.3.5, but this does not include solar radiation.

Research by Sauer, discussed in section 2.5, indicated that solar radiation can be calculated using Clustering and HMM from historical weather recordings and radiations. Figure 4.17, Figure 4.18 and Table 4.7 illustrated the divergence in element usages, ambient and geyser-related temporaries between the two Sundays in winter and summer. In both diagrams the recorded sensor data were joined with the three-day weather prediction.

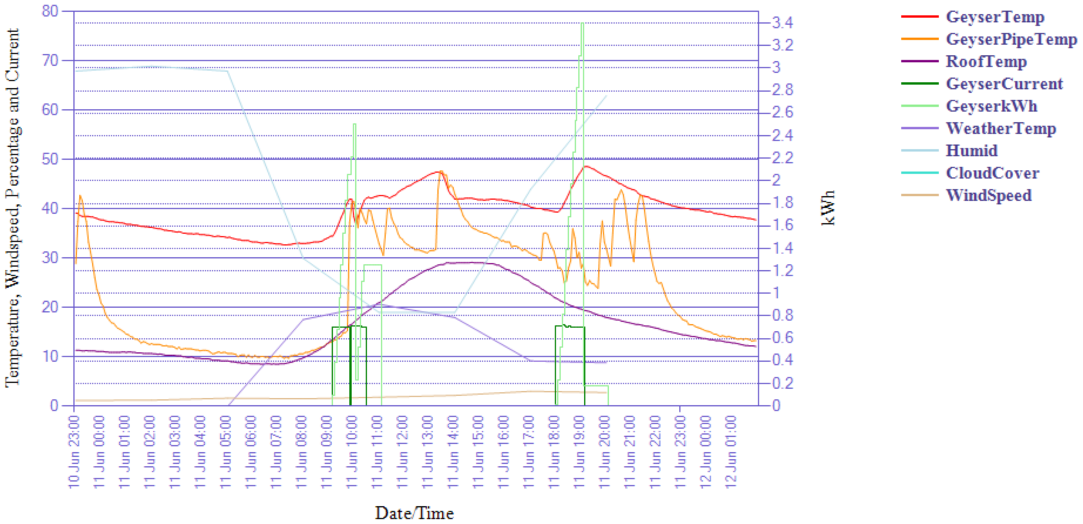


Figure 4.17   Geyser and surrounding temperatures in winter

The first notable difference is that the electrical element switched on three times in winter using 7.3 kWh and only once in the summer using 3.1 kWh.

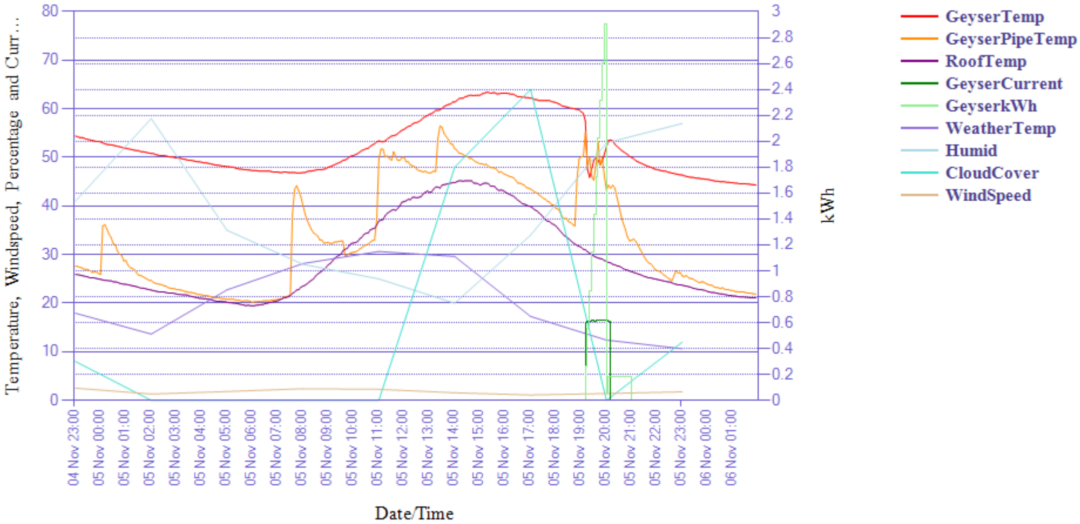


Figure 4.18 Geyser and surrounding temperatures in summer

Solar radiation heated the water to 47.5 °C with a 0% cloud cover and a maximum outside temperature of 21°C in winter. In summer the water was heated to 62.8°C with a 64% cloud cover and a maximum outside temperature of 32 °C. It is also important to note that the cloud cover increased to 50% at 12h00, but did not have any influence on the solar heating because the water was already heated to 62 °C.

Table 4.7 Geyser and surrounding environmental conditions for winter and summer

Season	**Geyser Temp** (°C)	**Predicte Day Temp** (°C)	**Hot water Pipe Spikes**	**Cloud Cover** (%)	**Geyser** (kWh)
Winter	47.5	21	10	0	7.3
Summer	62.8	32	8	64	3.1

The hotwater pipe temperature spikes were higher in winter, ten in contrast to eight in summer.

4.3.3 **Occupation Changes**

- Sleep over visitors

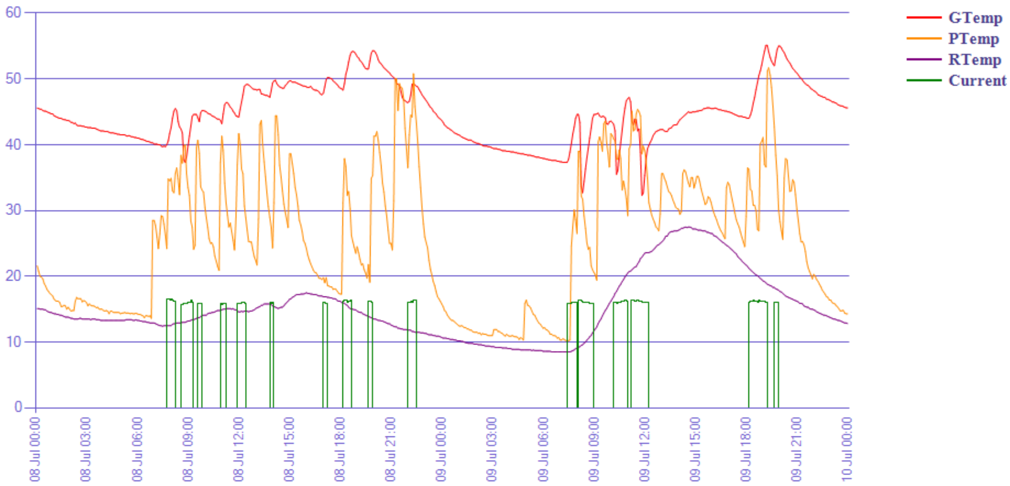


Figure 4.19 Occupation changes

Analysing the geyser recorder data in Figure 4.19, it is noticed that the hotwater usage is outside the normal behaviour for the weekend of 8-9 July 2017 compared to Figure 4.17. Another factor is that the element switched on outside the controller's predefined activation times, indicating that the element was switched on manually.

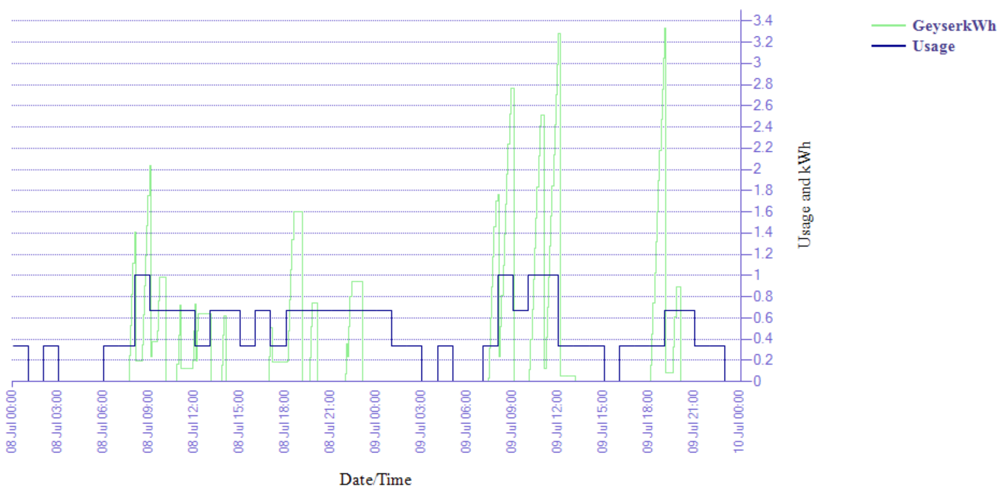


Figure 4.20 Fuzzy usage and element activation for 8-9 July 2017

Figure 4.20 and Figure 4.21 illustrates through the Fuzzy geyser usage and element kWh usage, the sudden difference in usages of hot water. The reason for the change in the profile is that the residents under investigation had sleep-over guests for the weekend of 8-9 July 2017.

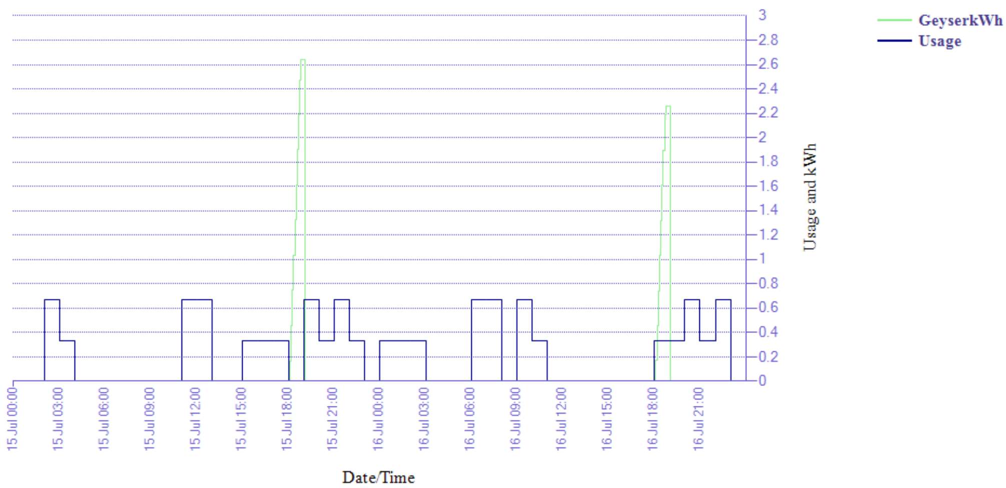


Figure 4.21 Fuzzy usages and element activation for 15-16 July 2017

- No occupation

Another problem would be if the water was heated, but there was nobody home that day to use the hot water. For the controller to inspect occupation, a movement sensor was installed and connected to the controller to record the household activities. As discussed in section 2.5.1, it is not a major factor during the summer months as the solar geyser operates at optimal capacity, as illustrated in Figure 4.22.

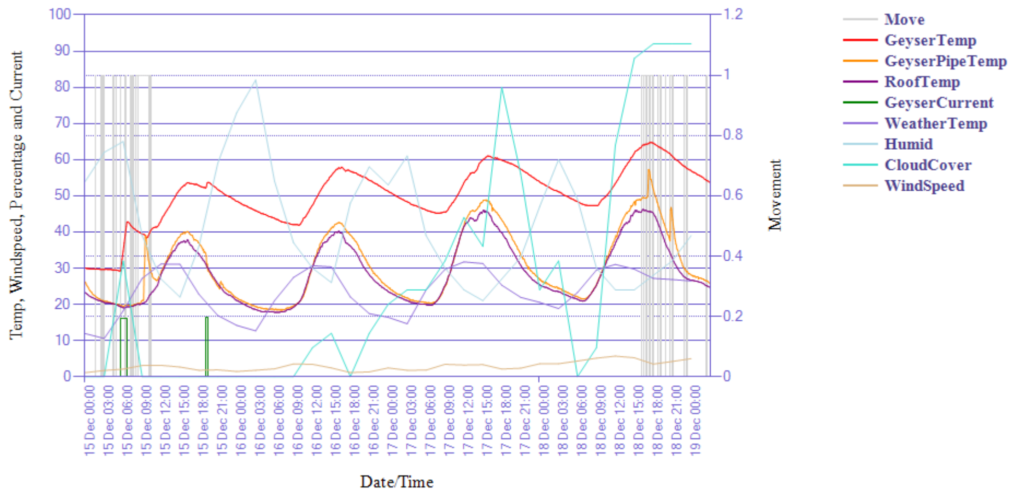


Figure 4.22 *No occupation on hot summer day with clear skies*

It is shown that on 15 December there was no movement recorded after 9h00. That day the geyser element was switched on from 18h00 until 18h20, consuming 0.688 kWh. The following two days there was no movement or hot water used, indicating that the premises were unoccupied. The three-day weather prediction predicted 12%, 44% and 64% and cloud cover of 30.35°C, 31.72°C and 31.02°C between 9h00 and 12h00 between 16 and 18 December when the solar geyser operates optimally. The first movement was recorded on 18 December at 15h10 and hot water usage at 16h15.

This becomes a problem when it is cloudy or in the winter months when the ambient temperature is lower.

4.4 *Weather patterns and Solar Geyser*

The geyser analysed during the study was equipped with a solar collector. Building a prediction model for a solar geyser, it is important to have historical knowledge about the hotwater usage and the weather patterns. One of the goals of the prediction model is to ensure that there is always hot water when needed. It is also useful for predicting the hot water temperature at a specific time and to calculate if solar energy is sufficient or if electrical backup power will be required.

The first weather recordings were made in March 2017, but the first cloud cover recordings started on August 2018 because it was believed that the temperature and humidity recording would be sufficient to predict the solar radiation. There are incomplete weather data recordings until October 2017 because the newly implemented system that had till then not been fully developed.

Analysing the roof and solar temperatures in Figure 4.5 and Figure 4.7 indicates knowing the weather conditions during the warm-up cycle which is between 9h00 and 14h00 is sufficient to predict the hot water temperature. Weather conditions outside the above times have a negligible impact on the efficiency of the solar collector. The three-day weather prediction from openweathermap, as discussed in section 3.3.5, retrieved data are in three-hour intervals starting at 0h00 GMT. South Africa's time zone is GMT+2, meaning that the data retrieved must between 8h00 and 14h00.

The data retrieved from the weather table and stored in a new table called solarheat are:

- Maximum temperature;
- Minimum temperature;
- Average cloud cover, and
- Average humidity

as shown in Figure 4.23.

```
SQL = “SELECT
       Max(Weather_Temprature) as Max_Weather_Temperature,
       Min(Weather_Temprature) as Min_Weather_Temperature,
       Avg(Weather_Clouds) as Max_Weather_Clouds,
       Avg(Weather_Humidity) as Max_Weather_Humidity
       FROM Weather
       WHERE Weather_Date >=  '" & StartDate & "' and Weather_Date < =
       '" & EndDate & "' and DATEPART(HOUR, Weather_Date) >= '8' and
       DATEPART(HOUR, Weather_Date) <= '14'”
```

Figure 4.23 SQL expression to retrieve weather conditions for partition clustering

The average cloud cover and humidity were taken because it was a better representation of the occurrence during the time span. Cloud cover changes the entire day. The maximum cloud cover for both days illustrated in Figure 4.24 and Figure 4.25 between 8h00 and 14h00 is 24%. On the entire morning of 6 March 2018, important for solar heating, the cloud cover was 24%, but on 9 March 2018 it became cloudy when solar heating was not a factor. The daily maximum cloud cover was inadequate to predict the solar collector’s sufficiency. The average cloud cover between 8h00 and 14h00 on 6 March was 16% and on 9 March, 8% which is a more accurate representation of the cloud cover.

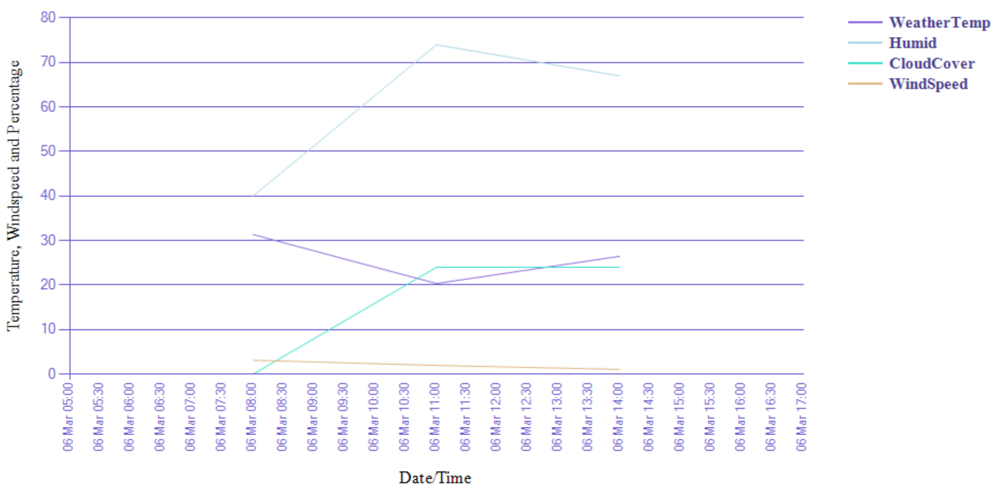


Figure 4.24 Cloud Cover 6 March 2018

Figure 4.25 Cloud Cover 9 March 2018

Temperature always has a linear trend from minimum to maximum as the day progresses, except if the weather condition changes, for example when it starts to rain. The average temperature trend can be calculated from the minimum and maximum if needed.

Partition Around Medoid clustering, as discussed in section 2.2.2, was used to analyse the summarised daily values with regard to heating water with solar radiation. Three medoids were chosen to retrieve a minimum, intermediate and maximum value [45, p.53]. To speed up calculations, the CLARANS algorithm with a count of 31 was used because it is an acceptable trade-off between cost and effectiveness [46, pp.475-478]. The interface for the clustering controller is illustrated in Figure 4.26.

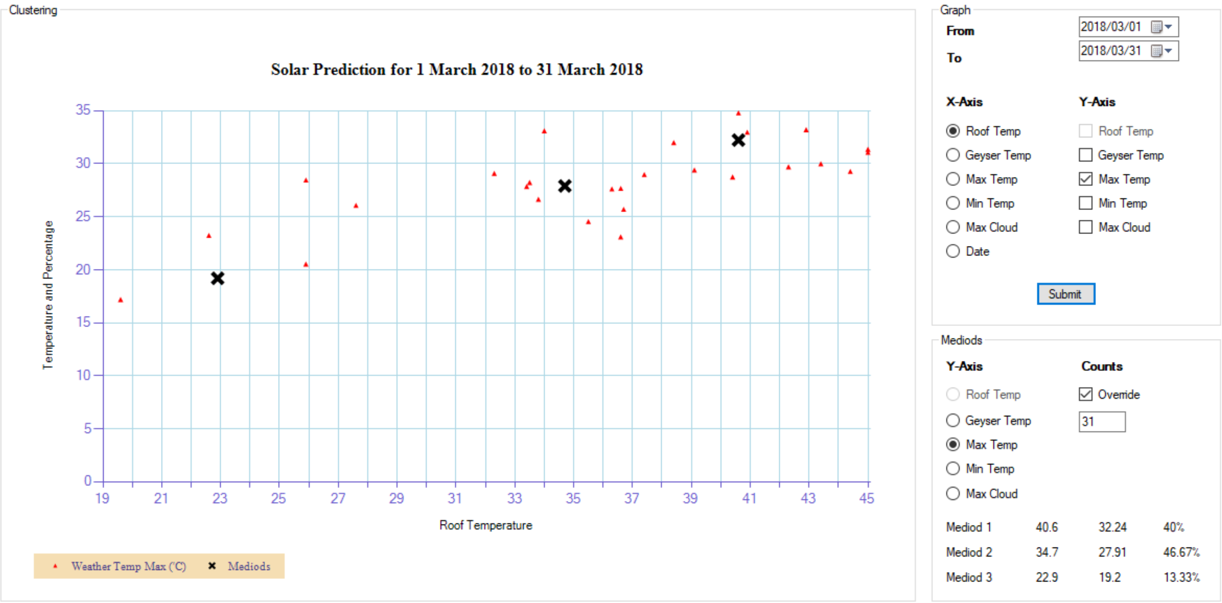


Figure 4.26 Controller's partial clustering interface

For monthly calculations the CLARANS algorithm is not required. It is only applicable if seasonal, quarterly or yearly calculations are required.

The medoids for the Roof and Maximum temperature between 1 January 2018 and 31 March 2018 without CLARANS are(43.7, 30.9), (35.6, 27.4) and (24.0, 23.1) and the calculation time was 3.25 minutes. With 31 random points of CLARANS the medoids are (43.7, 30.9), (36.9, 28.0) and (30.8, 26.2), but the calculation time only took 8 seconds. The first medoid is the same, the second medoid is out by less than 1°C, but the third medoid is out by 6°C.

Partial clustering was used to determine the relationship between the roof temperature and weather predictions for maximum temperature, humidity and cloud cover in sections 4.4.1 to 4.4.3. In section 4.4.4 the relation between roof temperature and geyser temperature was investigated.

4.4.1 Roof vs Maximum outside temperatures

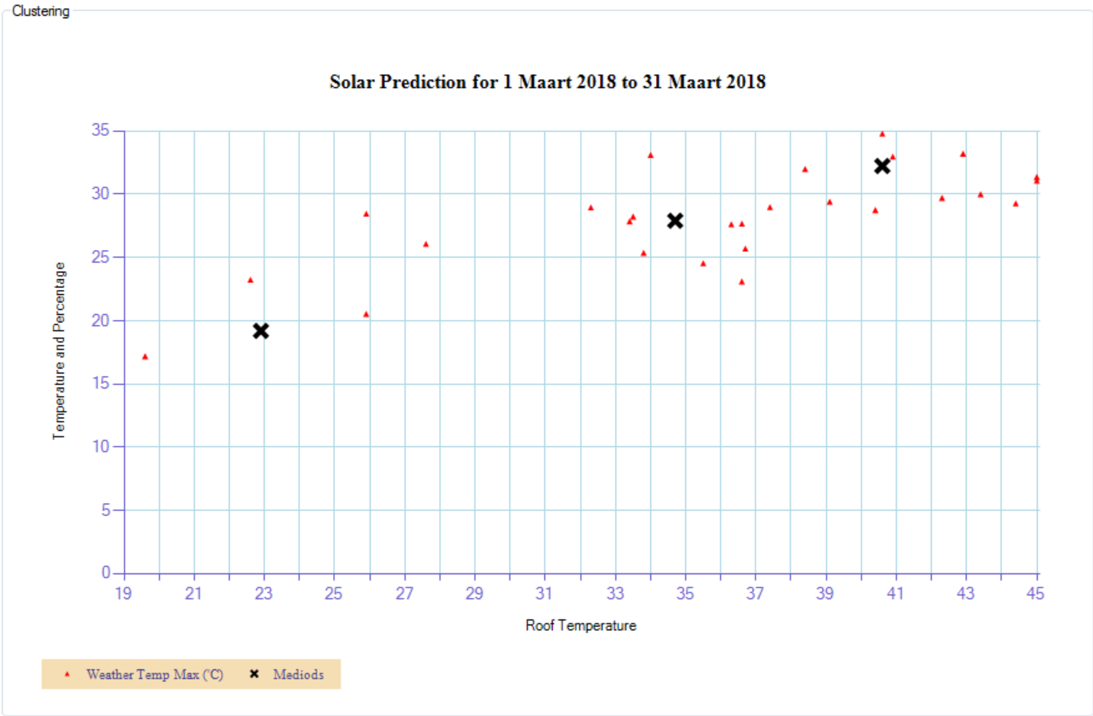


Figure 4.27 Partition clustering roof temperatures vs maximum outside temperatures

The roof temperature was used instead of the geyser temperature because it can be incorrect if energy is inserted into the system via electrical backup energy during the time span investigated. A relationship between the roof and geyser temperatures was investigated in section 4.4.4.

Figure 4.27 illustrates the graphical plotting for the maximum roof and outside temperature for each day during March 2018. Each medoid calculated and the weight of each medoid from March 2017 are displayed in Table 4.8.

Table 4.8 Monthly roof temperatures vs maximum outside temperatures

Month	Medoid1			Medoid2			Medoid3		
	Roof Temperature	Outside Temperature	Persentage	Roof Temperature	Outside Temperature	Persentage	Roof Temperature	Outside Temperature	Persentage
Jan 2018	32.5	25.2	20.7	42.6	27.4	37.9	46.5	33.0	41.4
Feb 2018	24.0	23.1	22.2	36.8	28.6	25.9	42.5	30.2	51.9
Mar 2017	33.3	28.7	35.0	40.0	0.0	10.0	41.7	30.0	55.0
Mar 2018	22.9	19.2	13.3	34.7	27.9	46.7	40.6	32.2	40.0
Apr 2017	29.2	23.9	51.9	35.7	25.9	41.7	37.4	0.0	7.4
Apr 2018	29.0	23.8	44.8	35.1	28.5	31.0	38.6	32.3	24.1
May 2017	27.0	27.6	27.6	29.0	0.0	10.3	32.7	24.6	62.1
May 2018	20.4	28.8	16.7	22.8	16.3	23.3	A 33.3	29.4	60.0
Jun 2017	27.0	18.5	38.5	29.2	0.0	3.85	30.1	22.8	57.7
Jun 2018	11.8	30.3	7.0	27.7	24.2	34.5	29.8	30.2	58.6
Jul 2017	28.6	20.5	50.0	28.3	0.0	20.0	31.0	23.9	30.0
Jul 2018	23.5	20.5	40.0	27.5	25.1	43.3	32.6	30.7	16.7
Aug 2017	29.7	22.3	53.3	31.3	0.0	3.3	34.7	25.0	43.3
Aug 2018	21.3	18.2	10.0	29.9	25.3	33.3	33.4	31.2	56.7
Sep 2017	29.7	28.3	31.0	39.1	30.0	41.4	40.2	0.0	27.6
Sep 2018	31.9	24.1	24.1	36.5	30.6	34.5	41.1	33.8	41.4
Oct 2017	27.5	22.4	33.3	36.0	27.1	26.7	41.2	29.5	40.0
Oct 2018	15.0	26.5	13.3	35.3	25.7	30.0	43.8	32.2	56.7
Nov 2017	18.5	15.5	3.5	33.4	28.2	31.0	43.5	31.9	65.5
Nov 2018	27.8	28.0	17.2	38.4	27.7	34.5	47.1	33.2	48.3
Dec 2017	23.3	29.6	15.0	31.2	27.7	35.0	40.0	30.7	50.0
Dec 2018	22.9	24.7	10.7	37.2	27.8	25.0	47.2	32.8	64.3

Investigating the table, the following were noted:

- A medoid with a zero must be excluded from the prediction model as it represents data that was not recorded.
- Comparing yearly medoids between the same months, no clear pattern is detected. This is proof that additional factors need to be included to define more accurate medoids. For example, one year the season can be dry and another it can be wet.

- Investigating the roof temperature independently, a pattern is noted where the maximum percentage from the previous year is a good indication for the next year. For example, in May 2017 there was a 62% chance the roof temperature would be 32.7°C and in 2018 there was a 60% chance for it to be to be 33.3°C as shown at point A. In March 2017 chances were 55% to be 41.7°C and in 2018 is was 40% to be 40.6°C, as shown at point B. The summer rainfall influence needs to be investigated for the differences in increase in percentage during the rainy months.
- No patterns were noted when comparing months in the same season, except for those expected as it is hotter in the summer months.

Investigating the weekly medoids for outside and roof temperatures as illustrated in Table 4.9, the following were noted:

- The most weighted medoids during the rainy season in March were between 50% and 77.8%, but the roof temperature difference was 8.7 °C but during the dry winter month June, the most weighted medoid weight was 50% and the roof temperature only differed by 2.6 °C.
- The second weighted medoid in March varied between 11% and 33% and the roof temperature also fluctuated by 18.3 °C, but in June the weight was 33% for all four weeks and the roof temperature differed by 2.6 °C.

Table 4.9 *Weekly roof temperatures vs maximum outside temperatures*

Week	Medoid1			Medoid2			Medoid3		
Mar 2018	Roof Temperature	Outside Temperature	Persentage	Roof Temperature	Outside Temperature	Persentage	Roof Temperature	Outside Temperature	Persentage
1 – 7	25.9	28.5	16.7	40.6	32.2	16.7	45.0	31.1	66.7
8 – 14	34.7	27.9	33.3	39.1	29.4	50.0	43.4	30.0	16.7
15 – 21	22.6	23.2	33.3	37.4	29.0	50.0	40.6	34.8	16.7
22 – 31	22.9	19.2	11.1	36.3	27.6	77.8	40.9	33.0	11.1
Jun 2018	Roof Temperature	Outside Temperature	Persentage	Roof Temperature	Outside Temperature	Persentage	Roof Temperature	Outside Temperature	Persentage
1 – 7	28.1	21.0	16.7	28.4	26.1	50.0	30.0	29.8	33.3
8 – 14	11.8	30.3	16.7	27.7	24.2	50.0	30.0	30.5	33.3
15 – 21	26.0	22.0	16.7	30.3	27.6	50.0	30.6	32.5	33.3
22 – 31	23.4	18.2	12.5	28.0	28.0	37.5	29.6	31.0	50.0

- The maximum roof temperature between the two above medoids for June is almost the same value implying that a predicted maximum roof temperature for the previous week could be predicted with an 83% change for June 2018. In the first week it was 28 °C for 66.7% of the time and for 33.3% it was 30 °C. In the second week it was 27.7 °C for 50% of the time and 30 °C for 33.3% of the time.
- During May, the rainy season, the previous week's maximum roof temperatures could not be used for the next week's predictions, indicating that another factor must be included in the model.
- When the medoid with the most weight's maximum outside temperature prediction was higher or lower, the next week's recorded roof temperature medoids followed the same trend. In March 2018 it was 45.0 °C formaximum roof temperature and 31.1 °C for maximum outside temperature. The second

week the temperatures dropped to 39.1 °C and 29.4 °C, the third week to 37.4 °C and 29.0 °C and again dropped to 36.3 °C and 27.6 °C.

The above findings indicate previous weeks' temperatures leading to more accurate prediction values than comparing equivalent months between the years.

Table 4.10 Daily roof temperatures vs maximum outside temperatures

Mar2018	Roof Temperature	Outside Temperature	Jun 2018	Roof Temperature	Outside Temperature
1	39.6	31.1	1	26.7	21.8
2	42.3	29.7	2	30.0	29.7
3	44.4	29.3	3	31.3	32.8
4	45.0	31.1	4	28.1	21.0
5	40.6	32.2	5	28.4	23.9
6	45.0	31.4	6	28.4	26.0
7	25.9	28.5	7	27.6	27.7

Investigating the daily medoids for outside and roof temperatures as illustrated in Table 4.10, the following are noted:

- During the rainy summer month March, the maximum predicted temperature differs by 2.6 °C, but the roof temperature differs by 19.1 °C. A higher outside temperate does not indicate a higher roof temperature.
- During the dry winter month June, the outside temperature varies with 11.8 °C, but the roof temperature fluctuates with only 3.7 °C. A higher outside temperature does indicate a higher roof temperature.

Inspecting monthly, weekly and daily data proved that previous weekly and daily data were more accurate to predict the following week's maximum roof temperatures. During the dry winter months, the values can be used, but during the rainy season, values fluctuate, indicating that the algorithm requires additional parameters to incorporate the rainy weather conditions.

Cloud cover and humidity record data from the weather predictions and its influence will be investigated in section 4.4.2 and 4.4.3.

4.4.2 Roof temperatures vs Cloud cover

Figure 4.28 illustrates the graphical plotting for the maximum roof and cloud cover for each day during March 2018.

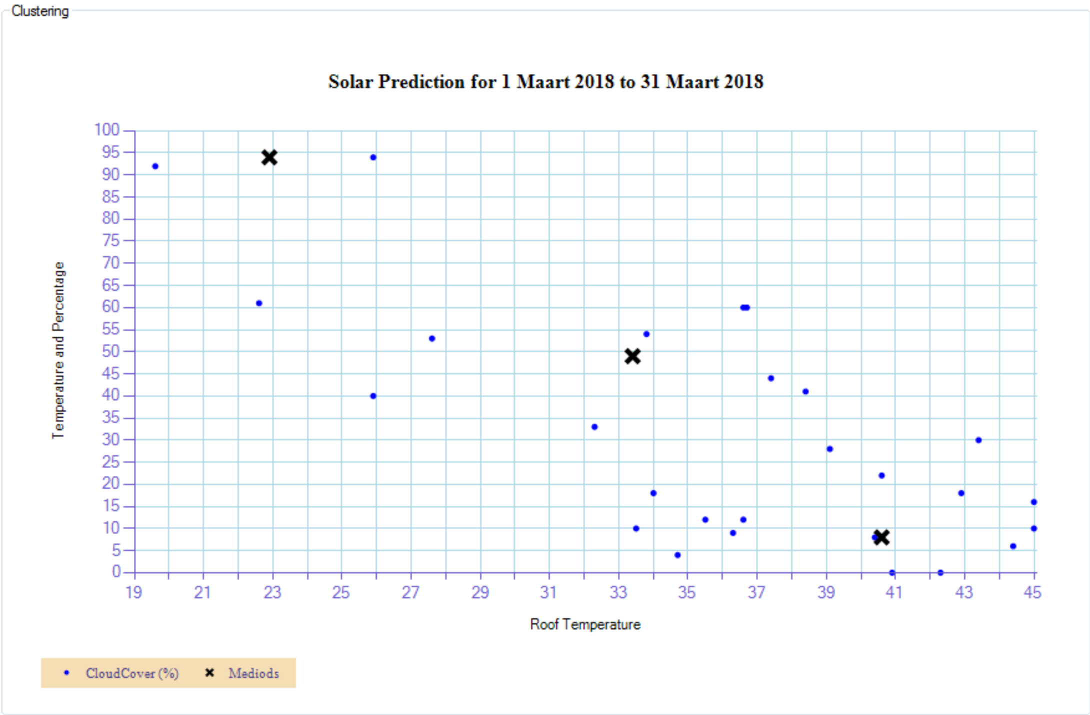


Figure 4.28 Partition clustering roof temperature vs cloud cover

Each medoid calculated and the weight of each medoid from August 2017 are displayed in Table 4.11; the following was noted:

- From May to October the most weighted medoid cloud cover was 0%. This proves why during this period accurate roof temperature can be predicted from only the maximum outside temperatures.

- During the summer months the cloud cover percentage and weight vary per month.
- A higher cloud cover does indicate a lower roof temperature.

Table 4.11 Monthly roof temperatures vs cloud cover

Month	Medoid1			Medoid2			Medoid3		
	Roof Temperature	Cloud Cover	Persentage	Roof Temperature	Cloud Cover	Persentage	Roof Temperature	Cloud Cover	Persentage
Jan 2018	44.1	0.0	20.7	43.7	21.0	31.0	40.3	48.0	48.3
Feb 2018	42.9	24.0	51.9	27.5	92.0	14.8	37.0	56.0	33.3
Mar 2018	40.6	8.0	56.7	33.4	49.0	33.3	22.9	98.0	10.0
Apr 2018	30.7	50.0	41.4	31.7	69.0	27.6	37.9	0.0	31.0
May 2018	17.0	88.0	6.7	33.4	0.0	73.3	26.0	38.0	20.0
Jun 2018	11.8	0.0	6.9	30.1	0.0	37.9	28.0	0.0	55.2
Jul 2018	21.1	85.0	3.3	22.8	29.0	16.7	27.5	0.0	80.0
Aug 2017	31.3	65.0	3.3	31.6	0.0	80.0	31.9	24.0	16.7
Aug 2018	29.9	33.0	26.7	32.0	0.0	46.7	32.7	10.0	26.7
Sep 2017	29.4	56.0	6.9	39.1	0.0	75.9	32.2	20.0	17.2
Sep 2018	34.2	26.0	10.3	38.3	0.0	72.4	43.0	49.0	17.2
Oct 2017	26.5	72.0	20.0	37.3	0.0	43.3	43.0	29.0	36.7
Oct 2018	35.9	69.0	26.7	41.7	0.0	46.7	42.2	37.0	26.7
Nov 2017	41.7	0.0	34.5	34.3	50.0	24.1	41.0	22.0	41.4
Nov 2018	41.4	0.0	37.9	47.8	62.0	37.9	42.9	24.0	24.1
Dec 2017	35.6	74.0	35.0	40.2	6.0	30.0	31.2	34.0	35.0
Dec 2018	46.3	17.0	60.7	45.2	58.0	25.0	30.1	86.0	14.3

Investigating the weekly cloud cover medoids as illustrated in Table 4.12, the following was noted:

- Between May and October, the medoids cloud cover where 0% with a weight of between 72.4% and 100%, except Aug 2018, indicating to have no influence on the roof temperatures during the winter months and the maximum outside temperatures is the only factor influencing the roof temperatures.

- During the summer months the cloud cover percentage and weight varies per week.
- A higher cloud cover does indicate a lower roof temperature as recorded during February and March 2018 with and cloud cover of 92% and 98% and a lower roof temperature of 27.5°C and 22.9°C. The same trend is seen during December 2018 where the cloud cover is 86% and the maximum roof temperature dropped to 30.1°C.

Table 4.12 Weekly roof temperatures vs cloud cover

Week	Medoid1			Medoid2			Medoid3		
Mar 2018	Roof Temperature	Cloud Cover	Persentage	Roof Temperature	Cloud Cover	Persentage	Roof Temperature	Cloud Cover	Persentage
1 – 7	25.9	40.0	16.7	42.3	0.0	16.7	45.0	10.0	66.7
8 – 14	33.4	49.0	33.3	39.1	28.0	33.3	40.6	8.0	33.3
15 – 21 A	22.6	61.0	50.0	37.4	44.0	16.7	40.6	22.0	33.3
22 – 31	22.9	94.0	11.1	36.3	9.0	55.6	33.8	54.0	33.3
June 2018	Roof Temperature	Cloud Cover	Persentage	Roof Temperature	Cloud Cover	Persentage	Roof Temperature	Cloud Cover	Persentage
1 – 7	28.1	0.0	66.7	30.0	0.0	16.7	31.3	0.0	16.7
8 – 14	11.8	0.0	16.7	26.6	0.0	66.7	29.6	2.0	16.7
15 – 21	25.9	0.0	33.3	30.6	0.0	50.0	33.2	0.0	16.7
22 – 31	23.4	2.0	12.5	28.3	0.0	50.0	29.6	0.0	37.5

Table 4.13 illustrates one week during the rainy season as already proven that cloud cover does not impact the roof temperature during the winter months.

- The previous day cloud cover is no indication of the following day's cloud cover.

- A lower cloud cover does not always indicate a higher roof temperature. The weight and percentage cloud cover together indicate to have an influence on the roof temperature. A high cloud cover with a low weight does not indicate to have any influence on the roof temperatures as shown at point A. This proves that the maximum outside temperature must be included when predicting the roof temperature.

Including the maximum temperature to Table 4.12 are shown in Table 4.13. It is noted that above statement is true except for the 15st and 20th March for similar cloud cover and maximum outside temperatures the difference in roof temperature recorded where 8.9°C. Investigating the previous day's recordings, it is noted that on 19 March the cloud cover was 60%, indicating that the previous day's weather does influence the current days roof temperatures. If this is true, the minimum temperature will be lower, but it is higher than the 15th, proving the above statement is incorrect.

Table 4.13 Daily roof and outside maximum temperatures vs cloud cover

March 2018	Roof Temperature	Cloud Cover	Outside Maximum Temperature	Outside Minimum Temperature
14	43.4	30	30.0	27.5
15	42.9	18	33.2	16.2
16	40.6	22	34.8	15.7
17	22.6	61	23.3	20.0
18	27.6	53	26.1	20.7
19	36.7	60	25.7	21.3
20	34.0	18	33.1	23.9
21	37.4	44	29.0	22.4

Investigating the original recorded dated illustrated in Figure 4.29 and Figure 4.30 it is noted that both days the humidity was almost 100%, but the average between 8h00 and 14h00 on 15 March was 54.3% and on 20 March 71.6%, proving it must be included into the prediction model.

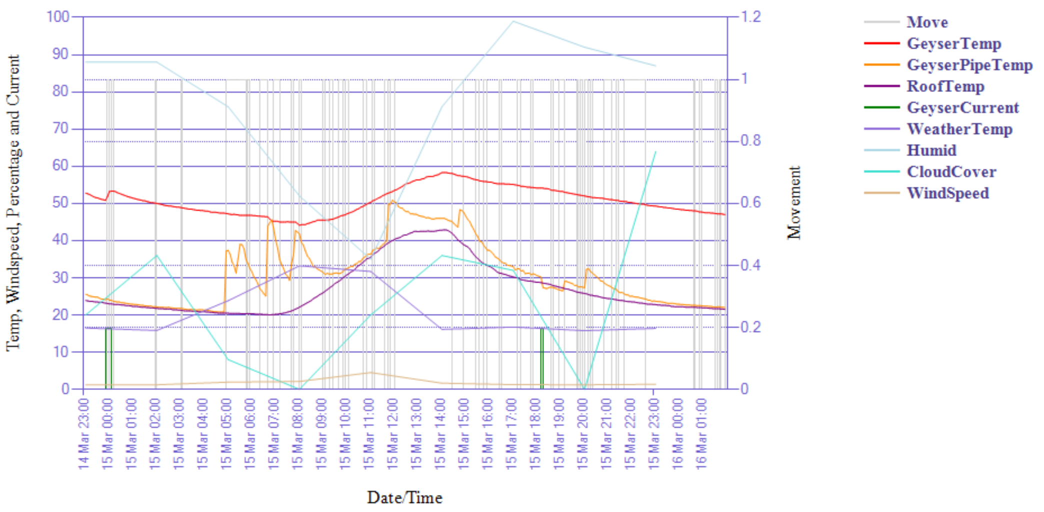


Figure 4.29 *15 March Geyser related data*

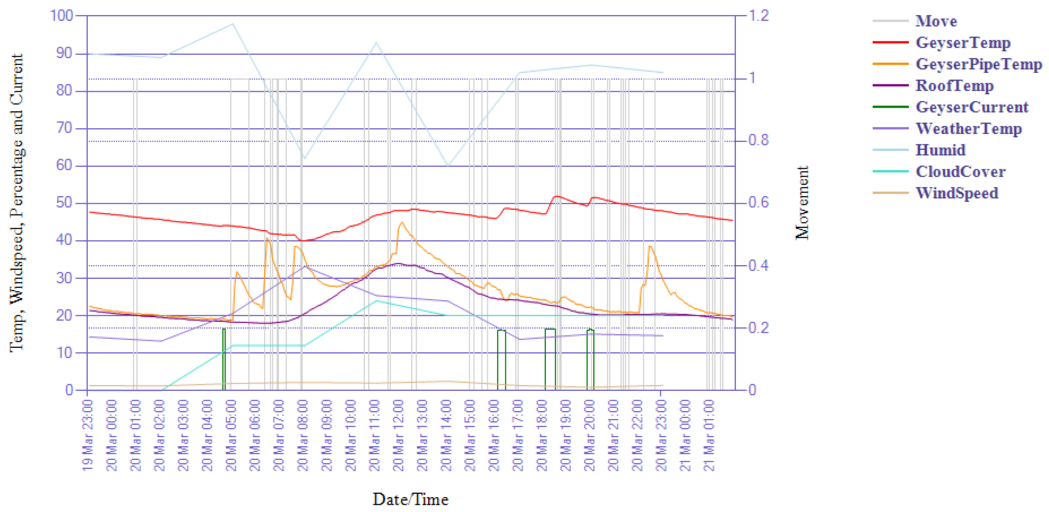


Figure 4.30 *20 March Geyser related data*

4.4.3 Roof temperatures vs Humidity

Figure 4.31 illustrates the medoids for the partition clustering of roof temperature and humidity during a rainy month.

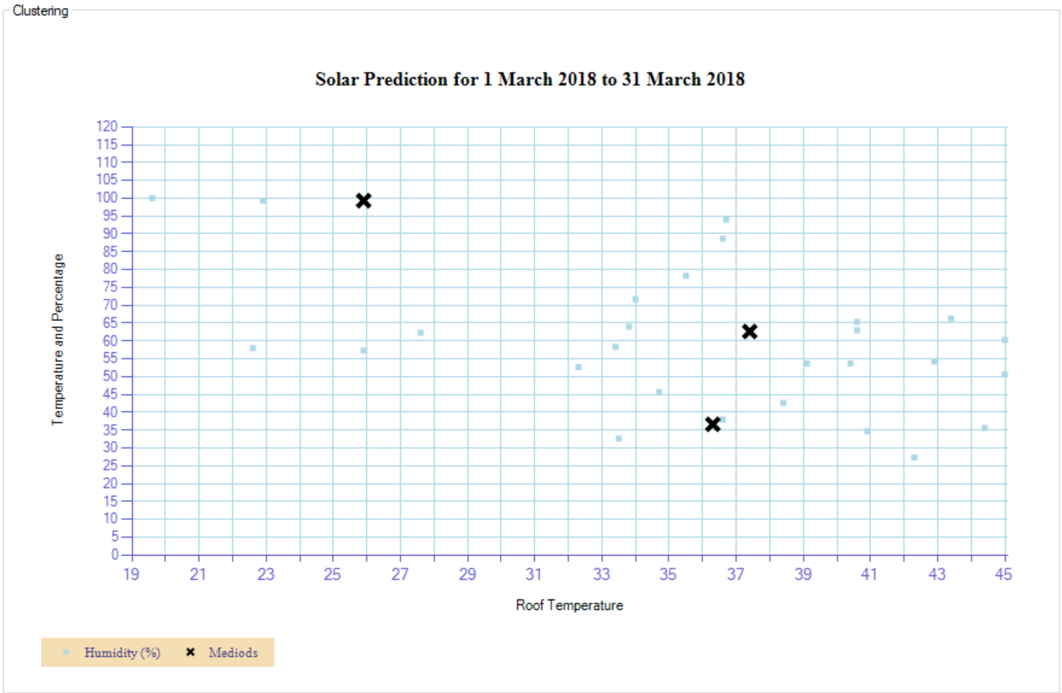


Figure 4.31 Partition clustering roof temperatures vs humidity

Table 4.14 Monthly roof temperatures vs humidity

Month	Medoid1			Medoid2			Medoid3		
	Roof Temperature	Humidity	Persentage	Roof Temperature	Humidity	Persentage	Roof Temperature	Humidity	Persentage
Jan 2018	38.6	63.7	34.5	43.3	37.3	34.5	48.4	22.0	31.0
Feb 2018	24.1	67.3	22.2	42.5	67.0	55.6	42.6	48.3	22.2
Mar 2017	25.4	0.0	10.0	35.1	46.7	30.0	40.6	27.0	60.0
Mar 2018	25.9	99.3	16.7	36.3	36.7	26.6	37.4	62.7	56.7
Apr 2017	30.4	39.3	55.6	34.4	67.3	33.3	36.2	0.0	11.1
Apr 2018	30.7	81.7	24.1	30.8	57.7	30.0	37.9	26.3	44.8
May 2017	29.0	0.0	10.3	30.0	33.3	48.3	33.5	19.3	41.4
May 2018	17.0	99.0	6.7	23.0	52.3	36.7	34.5	19.3	56.7

Table 4.14 cont. *Monthly roof temperatures vs humidity*

Jun 2017	27.0	40.7	26.9	28.4	29.3	26.9	30.1	19.3	46.2
Jun 2018	27.7	37.3	24.1	28.8	23.0	55.2	30.6	17.7	20.7
Jul 2017	27.5	39.3	30.0	28.3	0.0	20.0	31.0	20.0	50.0
Jul 2018	22.8	52.7	18.2	26.0	25.0	36.4	26.1	35.0	45.5
Aug 2017	29.5	39.7	16.7	30.5	28.0	46.7	33.1	12.3	36.7
Aug 2018	27.4	38.7	13.3	31.2	23.7	33.3	33.4	16.0	53.3
Sep 2017	32.8	31.7	37.9	40.2	0.0	27.6	40.6	16.3	34.5
Sep 2018	33.2	31.0	24.1	33.6	16.0	34.5	41.3	7.3	41.4
Oct 2017	26.2	79.0	26.7	38.5	16.3	46.7	38.9	42.7	26.7
Oct 2018	24.5	86.0	10.0	34.6	17.3	33.3	43.0	14.7	56.7
Nov 2017	37.1	64.3	24.1	42.9	39.0	34.5	45.0	20.0	41.4
Nov 2018	37.3	29.7	31.0	39.5	41.0	24.1	47.0	15.7	44.8
Dec 2017	29.9	99.7	15.0	35.6	61.0	35.0	40.2	31.0	50.0
Dec 2018	37.2	77.7	21.4	44.1	14.3	21.4	46.4	33.3	57.1

Each medoid was calculated and the weight of each medoid is displayed in Table 4.14; the following was noted:

- During the dry autumn and winter months from May to September, the most weighted medoid for humidity varied between 16% and 52.7% with a weight between 45% and 56%, except in May 2018 when the highest humidity recorded was 99% with a weight of 6.7%. This proof that during this period accurate roof temperature can be predicted from only the maximum outside temperature.
- During the summer months the humidity percentage varies each month and the most weighted medoid can be lower than in the winter, but at least one medoid per month varies between 63.7% and 99.7%, as highlighted in red, with the exception in March 2017, highlighted in purple, when the highest humidity was 46.7%.
- In most months with a higher humidity, lower roof temperature were recorded.

Table 4.15 Weekly roof temperatures vs humidity

Week	Medoid1			Medoid2			Medoid3		
Mar 2018	Roof Temperature	Humidity	Persentage	Roof Temperature	Humidity	Persentage	Roof Temperature	Humidity	Persentage
1 – 7	25.9	57.3	16.7	42.3	27.3	33.3	45.0	60.3	50.0
8 – 14	34.7	45.7	33.3	39.1	53.7	50.0	43.4	66.3	43.4
15 – 21	22.6	58.0	33.3	37.4	62.7	50.0	36.7	94.0	16.7
22 – 31	22.9	99.3	11.1	35.5	78.3	33.3	36.3	36.7	55.6
June 2018	Roof Temperature	Humidity	Persentage	Roof Temperature	Humidity	Persentage	Roof Temperature	Humidity	Persentage
1 – 7	28.1	44.7	33.3	28.4	23.0	50.0	31.3	14.7	16.7
8 – 14	11.8	18.3	16.7	27.7	37.3	50.0	29.6	21.3	33.3
15 – 21	25.9	37.0	33.3	30.3	30.0	16.7	30.6	17.7	50.0
22 – 31	23.4	28.7	12.5	28.0	17.7	37.5	28.7	25.0	50.0

Table 4.15 illustrates the weekly medoids between roof temperatures and humidity for one rainy and one dry month. The following was noted:

- During the first three weeks in March, the most weighted humidity varied between 53.7% to 62.7% and 36% for the fourth week as highlighted in red. The highest recorded humidity varied between 60.3% and 99.3%.
- Medoid with the highest humidity does not have the lowest roof temperature.
- During the winter month the humidity varied between 17% and 37% with a weight of 50%. The highest recorded humidity varied between 28.7% and 44.7%.
- The previous week's recordings were a more accurate reference for the current week's weather patterns than the previous year.

Table 4.16 illustrates the daily humidity for a week during summer and winter and the following were noted:

- The humidity in winter is lower than in the summer.
- The humidity can vary up to 30% from the previous day.
- Investigating humidity alone does not indicate any trend in predicting the roof temperature.

Table 4.16 Daily roof temperatures vs humidity

Mar2018	Roof Temperature	Humidity	Jun 2018	Roof Temperature	Humidity
15	42.9	54.3	1	26.7	57.3
16	40.6	65.3	2	30.0	23.6
17	22.6	58.0	3	31.3	14.6
18	27.6	62.3	4	28.1	14.6
19	36.7	94.0	5	28.4	36.0
20	34.0	71.6	6	28.4	23.0
21	37.4	62.7	7	27.6	24.0

Figure 4.32 demonstrates the plotted maximum temperature, cloud cover and humidity on the same diagram against the roof temperature during March 2018 during a rainy month.

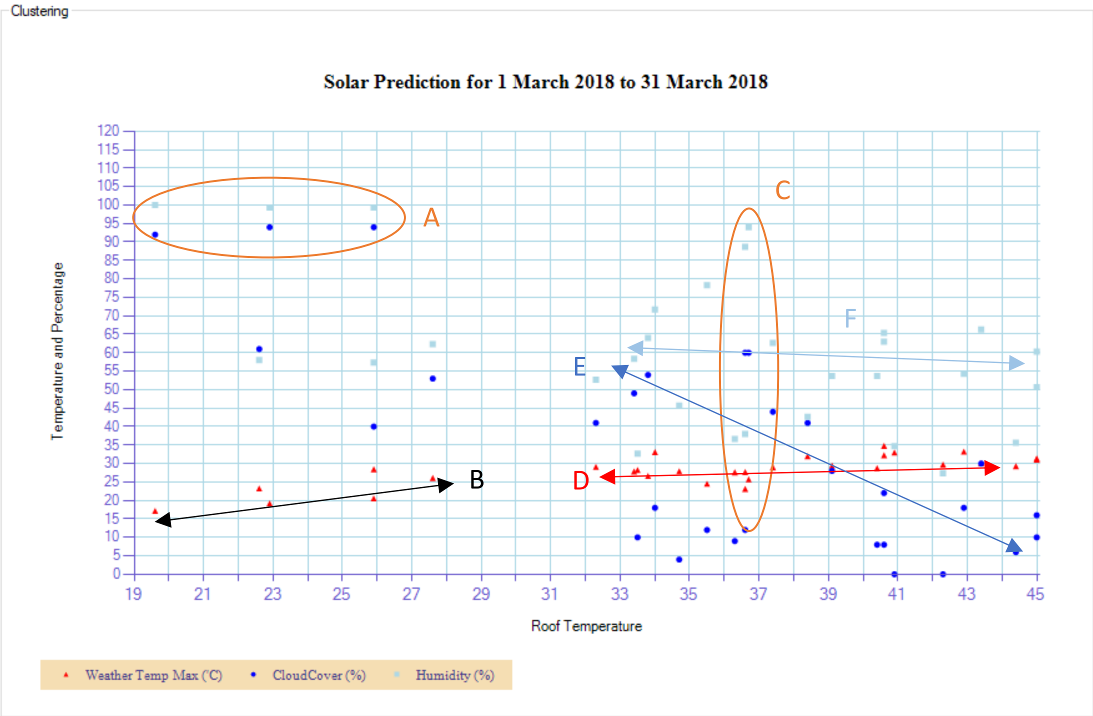


Figure 4.32 Partition clustering roof temp. vs humidity, cloud cover and maximum temp.

Investigating Figure 4.32 it is noted:

- The orange ellipse A illustrates that with a humidity and cloud cover above 90% the roof and maximum weather temperature is lower than other days as illustrated by line B.
- The orange eclipse C, highlights that with a humidity of 90% and a lower cloud cover of 60% a higher roof temperature is recorded but at the lower half of the recorded roof temperature average.
- Line D illustrates that with an ambient temperature that varies between 25 °C and 30°C that the roof temperature did increase with a decrease in cloud cover, as illustrated by line E, while the humidity varied between 55% and 65% as illustrated by line F.
- This proof, that the prediction model must include temperature, cloud cover and humidity to predict what the roof temperature will be.

4.4.4 **Geyser vs Roof temperatures**

The roof temperature was used as reference result from the weather predictions in section 4.4.1 to 4.4.3 and not the geyser temperature because electrical energy can be added at any time making the prediction module inaccurate. A relationship needs to be determined between the roof and geyser temperature. As discussed in section 2.2.2, the K-Medoid in Partitioning Methods is the most robust against noise. In this case, the noise is when the geyser's electrical backup element switches on, giving a temperature reading that was not only dependent on the solar collector and when hot water is drawn by the system. Another important fact to remember is that the higher daily geyser temperature was not used as it will be the same each day, because electrical energy were used after 16:00 to ensure the water is hot enough for the user's afternoon requirements. It was decided to record the geyser temperature at the time the roof temperature was the highest.

Figure 4.33 illustrates the relation between the recorded geyser and roof temperature with three medoids during the summer. A straight line, y = mx + c can be used to determine the relation. Four points marked A to D where identified as noise and being filtered out by the K-Medoid algorithm.

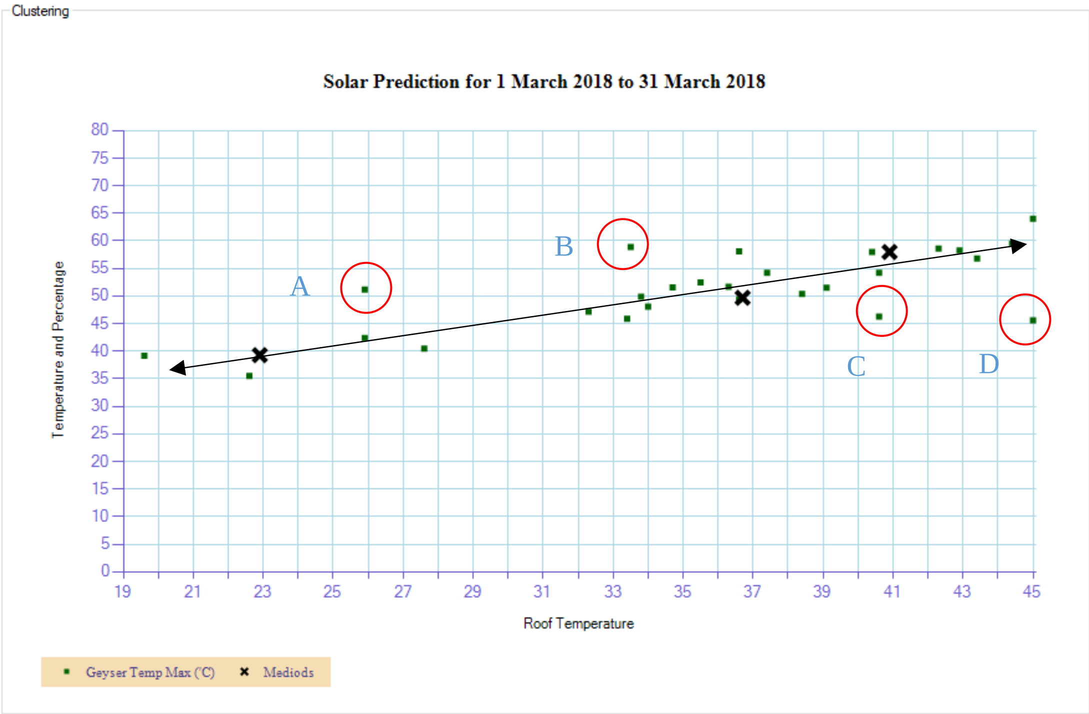


Figure 4.33 Partition clustering roof temperature vs geyser temperatures March 2018

The best fit, least square method was used to calculate the straight-line values. In example 4.2 all the values was used in example 4.3 only the three medoids are used.

Example 4.2

$$x_{\text{Avg}} = (\textstyle\sum_{i=1}^{n} x_i) / n = 35.74 \tag{4.28}$$

$$y_{\text{Avg}} = (\textstyle\sum_{i=1}^{n} y_i) / n = 51.05 \tag{4.29}$$

$$m = (\sum_{i=1}^{n} (x_i - x_{\text{Avg}})(y_i - y_{\text{Avg}})) \ / \ \sum_{i=1}^{n} (x_i - x_{\text{Avg}})^2 = 0.78 \tag{4.30}$$

$$c = y_{\text{Avg}} - m \, x_{\text{Avg}} = 23.24 \tag{4.31}$$

$$y = mx + c = 0.78 + 23.24 \tag{4.32}$$

Example 4.3

$x_{\text{Avg}} = (\sum_{i=1}^{n} x_i) / n = 33.5$

$y_{\text{Avg}} = (\sum_{i=1}^{n} y_i) / n = 49$

$m = (\sum_{i=1}^{n} (x_i - x_{\text{Avg}})(y_i - y_{\text{Avg}})) / \sum_{i=1}^{n} (x_i - x_{\text{Avg}})^2 = 0.97$

$c = y_{\text{Avg}} - m\, x_{\text{Avg}} = 16.58$

$$y = mx + c = 0.97 + 16.58 \qquad (4.33)$$

Comparing equation 4.32 with 4.33 there is no relation between the gradient and the y-axis interception. An important fact to remember is that equation 4.3 with the medoids having filtered out any noise, that is point A-D., Equation 4.4 all the values are used again excluding the incorrect values point A-D.

Example 4.4

$x_{\text{Avg}} = (\sum_{i=1}^{n} x_i) / n = 35.6$

$y_{\text{Avg}} = (\sum_{i=1}^{n} y_i) / n = 51.1$

$m = (\sum_{i=1}^{n} (x_i - x_{\text{Avg}})(y_i - y_{\text{Avg}})) / \sum_{i=1}^{n} (x_i - x_{\text{Avg}})^2 = 0.99$

$c = y_{\text{Avg}} - m\, x_{\text{Avg}} = 15.8$

$$y = mx + c = 0.99 + 15.8 \qquad (4.34)$$

Comparing equation 4.33 and 4.34 it is proven than the medoid can be used to calculate the straight line instead of using all the points, with the added benefit that incorrect data is filtered out. The straight-line $y = 0.99x + 15.8$ is included in Figure 4.33.

Figure 4.34 illustrates the relation between the recorded geyser and roof temperature with three medoids during the winter. Again, a straight line, $y = mx + c$ is used to determine the relation. Two points marked A and B were identified as noise and were filtered out by the K-Medoid algorithm. Point A was indicated where a bath was taken late one morning, leaving no time left to be heated again by solar energy and point B where the geyser was switched on early the morning without using it.

With Figure 4.33 is can be seen that the medoids is in a straight line, but this is not the case for the medoids in Figure 4.34. Calculate the expected geyser temperature from the roof temperature will be calculated using the line of best fit least square method where example 4.5 all the values, excluding point A and B, where used and in example 4.6 only the three medoids where used.

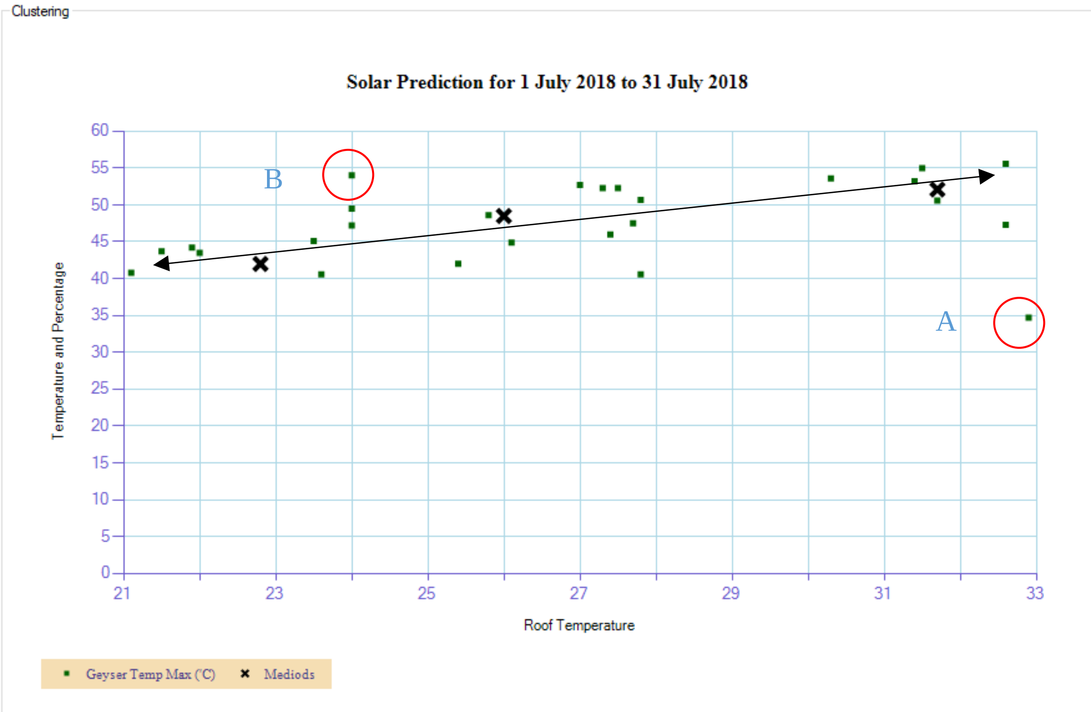


Figure 4.34 Partition clustering roof temperature vs geyser temperatures June 2018

Example 4.5

$x_{\text{Avg}} = (\sum_{i=1}^{n} x_i) / n = 26.9$

$y_{\text{Avg}} = (\sum_{i=1}^{n} y_i) / n = 48.2$

$m = (\sum_{i=1}^{n} (x_i - x_{\text{Avg}})(y_{\text{i}} - y_{\text{Avg}})) \ / \ \sum_{i=1}^{n} (x_i - x_{\text{Avg}})^2 = 0.97$

$c = y_{\text{Avg}} - m\, x_{\text{Avg}} = 21.9$

$$y = mx + c = 0.97 + 21.9 \qquad (4.35)$$

Example 4.6

$x_{\text{Avg}} = (\sum_{i=1}^{n} x_i) / n = 26.8$

$y_{\text{Avg}} = (\sum_{i=1}^{n} y_i) / n = 47.5$

$m = (\sum_{i=1}^{n} (x_i - x_{\text{Avg}})(y_{\text{i}} - y_{\text{Avg}})) \ / \ \sum_{i=1}^{n} (x_i - x_{\text{Avg}})^2 = 1.07$

$c = y_{\text{Avg}} - m\, x_{\text{Avg}} = 18.7$

$$y = mx + c = 1.07x + 18.7 \qquad (4.36)$$

Comparing equation 4.33 and 4.34 proofs than the medoid can be used to calculate the straight line instead of using all the points, with the added benefit that incorrect data is filtered out. The straight-line $y = 1.07x + 18.7$ is included to Figure 4.34.

Table 4.17 illustrates each month's medoids for Roof and Geyser temperatures, including the weight starting March 2017 sorted from lowest to highest roof temperatures.

Table 4.17 Monthly roof temperatures vs geyser temperatures

Month	Medoid1			Medoid2			Medoid3			y = mx + c	
	Geyser Temperature	Roof Temperature	Persentage	Geyser Temperature	Roof Temperature	Persentage	Geyser Temperature	Roof Temperature	Persentage	m	c
Jan 2018	50.9	32.5	24.1	54.8	43.2	34.5	60.0	46.9	41.4	0.6	31.8
Feb 2018	46.0	24.6	22.2	50.6	37.0	37.0	57.8	42.9	40.7	0.6	30.4
Mar 2017	51.0	25.4	10.0	47.2	35.6	30.0	56.3	42.5	60.0	0.3	42.6
Mar 2018	39.3	22.9	16.7	49.7	36.7	50.0	58.0	40.9	33.3	1.0	15.8
Apr 2017	48.5	21.3	11.1	44.7	29.2	37.0	52.0	36.2	51.9	0.2	42.1
Apr 2018	45.7	28.6	34.5	51.2	35.0	34.5	56.2	37.9	31.0	1.1	14.3
May 2017	42.6	27.5	34.5	46.0	31.5	27.6	49.3	33.4	39.9	1.1	12.4
May 2018	39.4	18.8	30.0	46.5	30.3	23.3	52.0	34.2	46.7	0.8	24.5
Jun 2017	46.7	26.7	34.6	39.3	28.9	34.6	48.4	30.5	30.8	0.2	38.6
Jun 2018	43.8	11.8	6.9	47.3	28.3	55.2	52.8	30.1	37.9	0.4	39.1
Jul 2017	35.6	21.0	3.3	47.5	27.9	46.7	51.4	31.0	50.0	1.6	2.1
Jul 2018	42.0	22.8	33.3	48.5	26.0	33.3	52.1	31.7	33.3	1.1	18.7
Aug 2017	47.8	26.4	20.0	52.2	32.0	60.0	55.7	35.7	20.0	0.8	25.4
Aug 2018	33.4	54.1	40.0	25.1	42.9	10.0	31.1	49.8	50.0	1.3	9.8
Sep 2017	47.7	29.4	24.1	53.9	35.0	24.1	58.0	40.6	51.7	0.9	21.0
Sep 2018	36.4	55.5	24.1	34.1	51.8	27.6	40.7	57.0	48.3	0.7	27.6
Oct 2017	47.3	26.5	33.3	56.4	37.3	33.3	56.1	41.2	33.3	0.7	30.5
Oct 2018	43.8	57.6	56.7	17.7	40.9	13.3	35.3	54.0	30.0	0.7	29.7
Nov 2017	49.8	32.2	24.1	52.6	40.6	27.6	58.9	44.9	48.3	0.6	27.7
Nov 2018	45.9	28.8	17.2	54.3	37.3	31.0	58.5	46.3	51.7	0.7	25.2
Dec 2017	44.9	29.9	35.0	51.3	38.2	45.0	60.1	46.0	20.0	0.9	16.2
Dec 2018	41.6	22.1	10.7	55.1	43.1	50.0	60.0	47.4	39.3	0.7	25.5

Investigating Table 4.17 the following is noted:

- There is a trend for similar climate conditions where the gradient is similar per year. Previous year gradient value cannot be used for the current year, but the pattern can be used. For example, in March 2017 and April 2017 the gradient is almost the same at 0.2 and 0.3. The same trend is seen in 2018 but now the gradient is 1.0 and 1.1.

- As expected, higher geyser temperatures were recorded during the summer month. It is important to remember that the temperatures recorded by default are outside the time zones when the standby electrical element is switched on.
- The gradient varies between 0.6 and 0.9 for the hotter months between August and February.

Table 4.18 Weekly roof temperatures vs geyser temperatures

Week	Medoid1			Medoid2			Medoid3			*y* = mx + c	
Mar 2018	Geyser Temperature	Roof Temperature	Persentage	Geyser Temperature	Roof Temperature	Persentage	Geyser Temperature	Roof Temperature	Persentage	m	c
1 - 7	51.2	25.9	16.7	46.3	40.6	33.3	59.5	44.4	50.0	0.2	43.8
8 - 14	45.9	33.4	16.7	58.0	40.4	33.3	51.5	39.1	50.0	1.5	-4.8
15 - 21	35.5	22.6	33.3	49.7	36.7	33.3	54.2	40.6	33.3	1.0	12.2
22 - 31	39.3	22.9	11.1	58.1	36.3	33.3	49.9	33.8	55.5	1.2	9.6
Jun 2018	Geyser Temperature	Roof Temperature	Persentage	Roof Temperature	Outside Temperature	Persentage	Roof Temperature	Outside Temperature	Persentage	m	c (Red Line)
1 - 7	51.9	30.0	16.7	45.7	28.1	33.3	49.2	28.4	50.0	2	-30.1
8 - 14	43.8	11.8	16.7	42.7	27.7	33.3	51.6	28.8	50.0	0.2	40.8
15 - 21	54.5	33.2	16.7	47.4	25.9	33.3	53.0	30.6	50.0	0.9	21.8
22 - 31	46.8	23.4	25.0	52.8	30.1	25.0	47.3	28.3	50.0	0.7	29.1

Table 4.18 illustrates the weekly medoids for two months in the same year between the roof and geyser temperatures. On investigating the table, the following were noted:

- There is no gradient trend between concurrent weeks.
- No trend was noted between the geyser temperature and the roof temperature. An example is when the maximum geyser temperature where 51°C the maximum roof temperatures varied between 25°C and 39°C.
- There is a trend in the weight for each medoid per week at 16.7%, 33.3% and 50%.

Table 4.19 illustrates the daily gradients between the current and previous day for one week in March and June between the roof and geyser temperatures. On investigating the table, it was noted that there is no usable pattern visible.

Table 4.19 Daily roof temperatures vs geyser temperatures

Mar 2018	Geyser Temperature	Roof Temperature	m	c
1	55.9	39.6		
2	58.6	42.3	1.0	16.3
3	59.5	44.4	0.4	40.5
4	64.0	45.0	7.5	-273.5
5	46.3	40.6	4.0	-117.0
6	45.6	45.0	-0.2	52.8
7	51.2	25.9	-0.3	58.8
8	42.4	25.9	Undefined	Undefined
Jun 2018	**Geyser Temperature**	**Roof Temperature**	**m**	**c (Red Line)**
1	48.9	26.7		
2	51.9	30.0	1.0	24.6
3	48.4	31.3	-2.7	132.7
4	45.7	28.1	0.8	22
5	49.2	28.4	11.6	.282.1
6	43.9	28.4	Undefined	Undefined
7	49.3	27.6	-6.8	235.6
8	53.6	29.3	2.5	-20.5

With no solution found for a relation between the roof and geyser temperature, a new approach was taken to investigate the relation if the past seven days is always taken as shown in Table 4.20.

Table 4.20 Seven days roof temperatures vs geyser temperatures

Days	Medoid1			Medoid2			Medoid3			*y* = mx + c	
Mar 2018	Geyser Temperature	Roof Temperature	Persentage	Geyser Temperature	Roof Temperature	Persentage	Geyser Temperature	Roof Temperature	Persentage	m	c
1 - 7	59.5	44.4	50.0	46.3	40.6	33.3	51.2	25.9	16.7	0.2	43.8
2 - 8	59.5	44.4	33.3	46.3	40.6	33.3	51.2	25.9	33.3	0.2	43.8
3 -9	64.0	45.0	33.3	46.3	40.6	33.3	51.2	25.9	33.3	0.4	38.3
4 - 10	58.0	40.4	16.6	46.3	40.6	33.3	51.2	25.9	50.0	0.1	49.8
5 - 11	58.0	40.4	16.6	45.9	33.4	50.0	51.2	25.9	33.3	0.4	36.5
6 - 12	28.8	40.4	33.3	27.9	34.7	33.3	28.5	25.9	33.3	0.0	28.0
7 - 13	29.4	39.1	50.0	27.9	34.7	33.3	20.6	25.9	16.7	0.6	3.0
8 - 14	45.9	33.4	16.7	58.0	40.4	33.3	51.5	39.1	50.0	1.5	-4.8

On investing the table, the following was noted:

- With some entries the three medoids is the same except for the weight as with the 1st to 7th and the 2nd to 8th indicating that weight or probability must be included in the prediction model.
- Between entries in most cases, only one medoid changes.
- The gradient and *y*-axis cut values are more stable and indicate if temperature is raising or declining.

4.5 *Swimming pool heat loses and gain*

Figure 4.35 illustrates the 12000 litre swimming pool with 0.75kW pump motor and four solar panels that were monitored. The monitor system must be able to detect the difference in temperature between the pool water and the heated water from the solar panels. The ambient sensor [Arrow 1] was placed next to the cold water sensor, but without touching any water pipes. Measuring the pool water temperature [Arrow 2] a temperature sensor was mounted against the water pipe before the bypass valve. Measuring water temperature in this manner where the cheapest, safest and easiest, because no existing plumbing needed to change and no electrical wires contacting water. The heated pool water [Arrow 3] was measured just before the bypass valve from the roof.

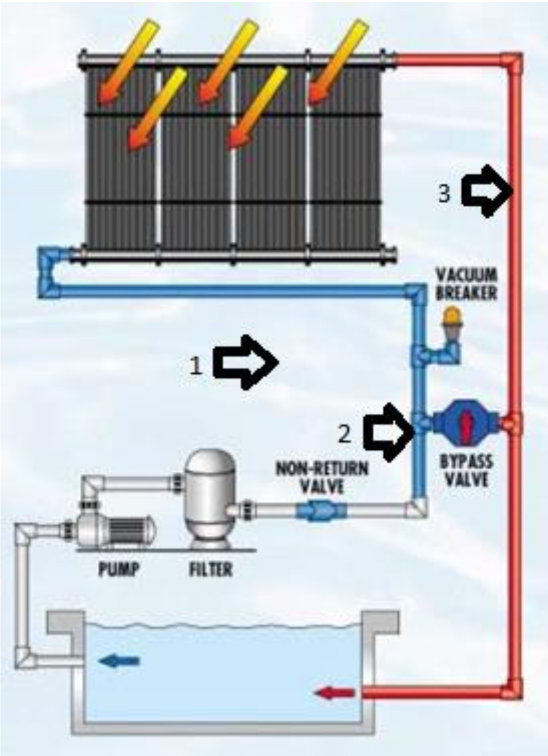


Figure 4.35 Existing swimming layout with solar heating.

There was no need implementing fuzzy logic to determine different levels of hot water usage as the only heat loss from the pool is thermal loses to the surrounding atmosphere. The prediction module must include ambient temperature, cloud cover, humidity and wind speed, as it does influence outdoor water temperatures as discussed in section 2.6.

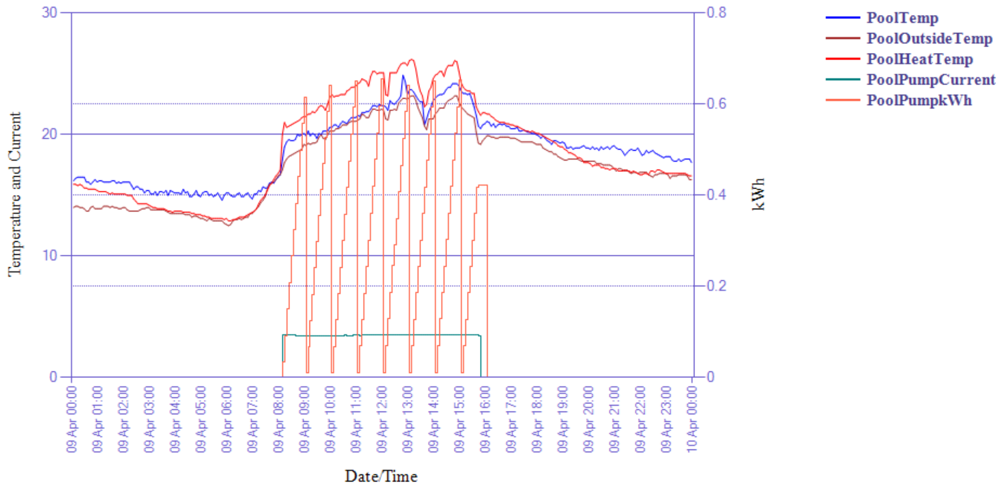


Figure 4.36 Swimming pool reading 9 April 2019

Figure 4.36 is a graphical illustration of all the swimming pool related readings from 9 April 2019. The diagram shows the pool pump drawn current between 08:00 and 15:40 consuming 4.93kW @R1.18 per kW.

As discussed in section 2.6, the pool pump needs to run for four hours during the winter and six hours during the summer. The diagram shown that the pump did run for eight hours. Investigate the diagram it is noted that the outside temperature dropped between 13h00 and 14h00 and again after 15h00. Switching off the pump when a drop in temperature was detected, would have saved 1.5 kW and R1.77 that day without impacting the heating of the pool water.

.

During March 2019 South Africa experienced power load shedding impact manual timers to be out of sync activating the pool pump at 9h40 and deactivated at 18h00 as shown in Figure 4.37. The heated water temperature had a steep decline after 16h00 and the pump could have been deactivated at 16h00 as it did already run for six hours, saving 1.5kW.

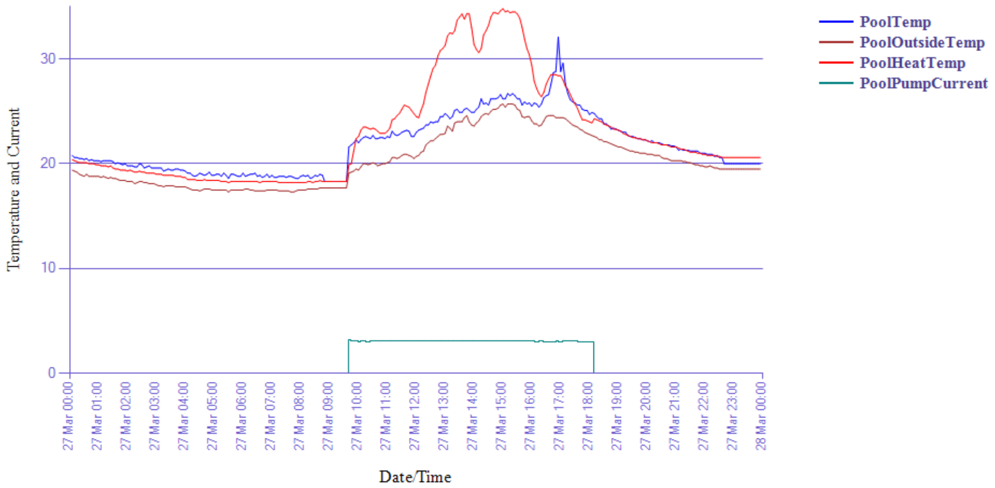


Figure 4.37 Swimming pool reading timer out of sync

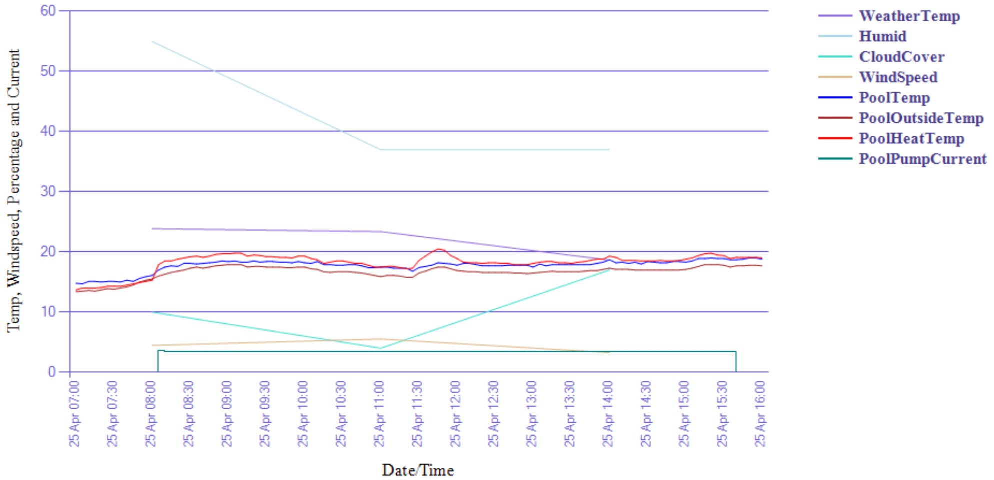


Figure 4.38 Swimming pool reading with moderate wind speed

Figure 4.38 illustrates the one of the highest predicted wind speed during April was just over 5.5 km/h. It can be seen by a decrease in humidity and cloud cover and an increase in wind speed that the pool water temperature did decrease. A suitable wind speed value must be identified for acceptable swimming conditions.

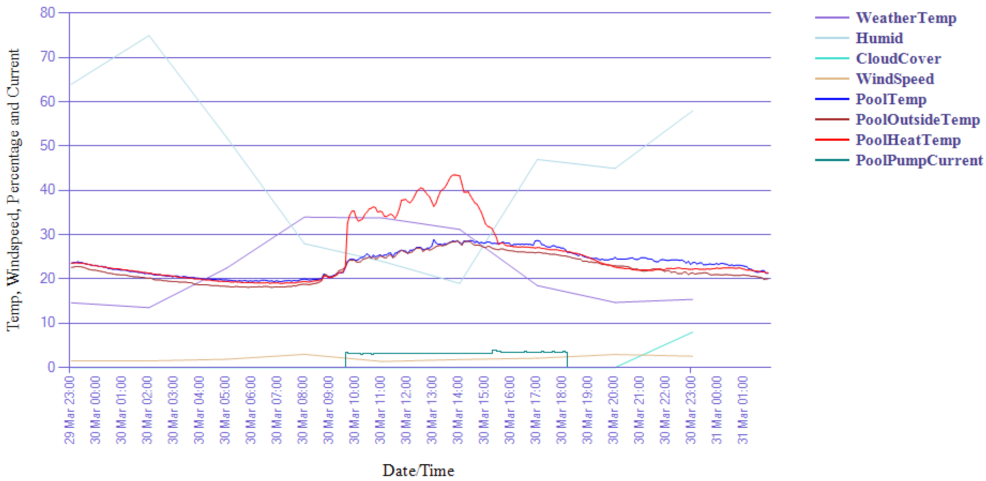


Figure 4.39 *Perfect sunny swimming conditions*

Figure 4.39 illustrates recorded values for 30 March indicating it was a perfect sunny swimming day because between 10h00 and 18h00:

- the heated water temperate where on average 10°C higher than the pool temperature;
- Cloud Cover 0%;
- Humidity is low (This will be different at coastal areas);
- Wind speed is low and varied between 1.37 km/h and 2.11 km/h and
- Predicted maximum temperature was 33°C.

The raw data sensor data is available in Appendix H.

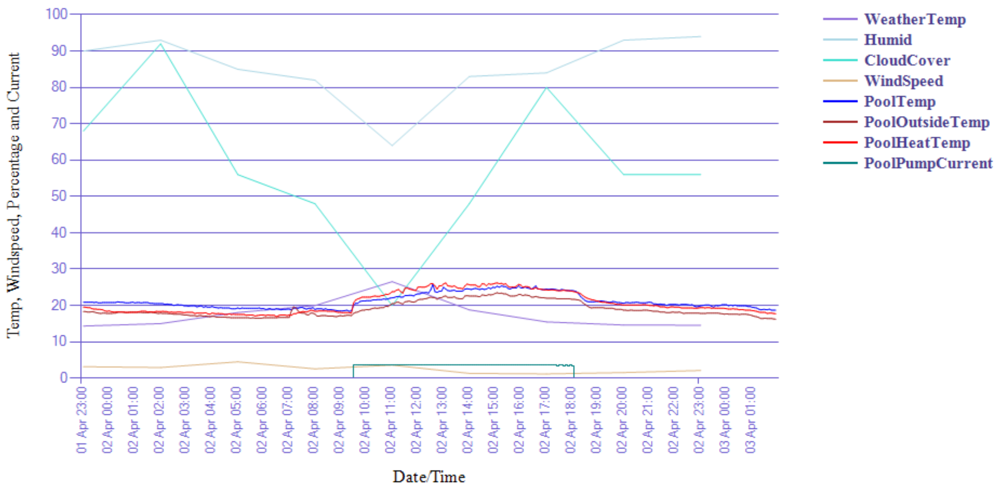


Figure 4.40 *Rainy unsuitable swimming conditions*

Figure 4.40 illustrates recorded values for 2 April indicating can be a rainy day unsuitable for swimming because between 10h00 and 18h00:

- the heated water temperate where on average 2°C higher than the pool temperature;
- Cloud cover varied between 20% and 80%;
- The humidity was high and varied between 64% and 95%;
- Wind speed is low and varied between 1.16 km/h and 3.56km/h and
- Predicted maximum temperature was 26°C.

The raw sensor data is available in Appendix I.

Figure 4.41 illustrates a spike in temperature from the water coming from the solar panels in the morning and a spike in the swimming pool temperature in the afternoon. After investigating it was found that the spike did occur on some days when the pump switched on or off. In the mornings the water in the panels started to be heated without any flow. After the pump was activated, the heated water passed the solar sensor with water much hotter. In the afternoon after the pump was deactivated, the water was still heated on the roof and the water slowly flowed back to the pool passed the pool temperature sensor. These scenarios indicating that the pool pump could be activated earlier and later the day heating the water.

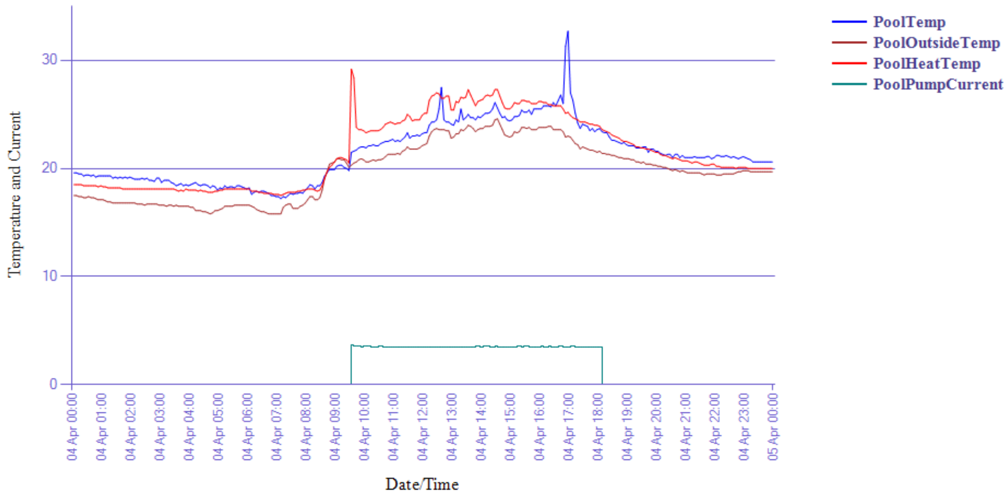


Figure 4.41 Pool and Solar Temperature spikes

4.6 *Weather patterns and swimming pool temperatures*

Building a prediction model for a swimming pool, it is important for a historical knowledge about the solar heating behaviour combined with weather prediction patterns. The goals of the prediction model are to ensure that the pump is activated at the correct times for ultimate performance, heating the pool water.

Analysing the pool related temperatures in Figure 4.39 and Figure 4.40 indicates knowing the weather conditions during the hot up cycle which is between 9h00 and 14h00 is sufficient to predict the hot water temperature. The time duration is the same as for the solar geyser as discussed in section 4.4 Weather conditions outside above time zone have negligible impact on the heating of the pool water.

As discussed in section 2.6, wind speed does influence the pool temperature and was include in the solar heat database as shown in Figure 4.42.

```
SQL = "SELECT
Max(Weather_Temprature) as Max_Weather_Temperature,
Min(Weather_Temprature) as Min_Weather_Temperature,
Avg(Weather_Clouds) as Max_Weather_Clouds,
Avg(Weather_Humidity) as Max_Weather_Humidity
Max(Weather_Windspeed) as Max_Weather_Windspeed
FROM Weather
       WHERE Weather_Date >=  '" & StartDate & "' and Weather_Date

< = '" & EndDate & "' and DATEPART(HOUR, Weather_Date) >= '8' and

DATEPART(HOUR, Weather_Date) <= '14'"
```

Figure 4.42 Updated SQL expression retrieve weather conditions for partition clustering

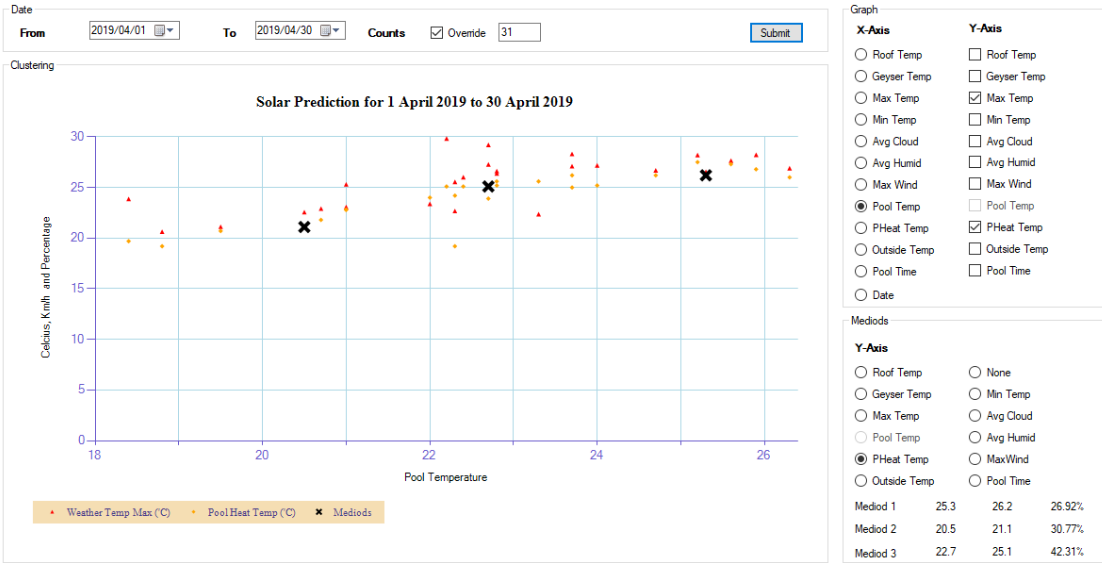


Figure 4.43 Updated Controller's partial clustering interface

Partition Around Medoid clustering, as discussed in section 2.2.2, was used to analyse the summarised daily values with regard to heating the pool water with solar radiation. Three medoids were chosen the retrieve a minimum, intermediate and maximum values. Speeding up calculations, CLARANS algorithm with a count of 31 was used because it is an acceptable trade-off between cost and effectiveness. The updated interface for pool related recordings for the controller's partial cluster interface is illustrated in Figure 4.43.

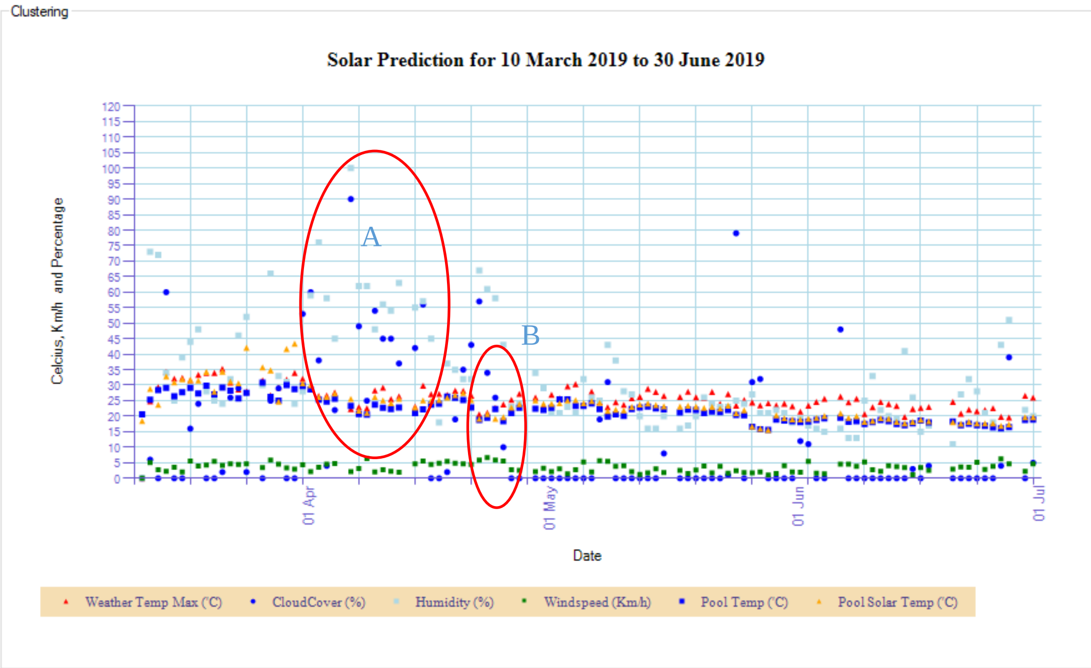


Figure 4.44 Daily recordings for weather predictions with pool and pool solar temperatures from March to June 2019

Figure 4.44 illustrated the daily weather predictions plotted with the pool and pool solar temperatures together from March to Jun 2019 and Figure 4.45 focus on April 2019. The following was noted:

- March maximum weather and pool water temperatures where the highest with low humidity and cloud cover, indicating perfect swimming conditions
- Point A indicates between 6th and 12th of April it was cloudy, high humidity with a drop in temperature indicating that it could be overcast and rainy.
- Point B indicates that between 22nd and 25th of April there was a drop in temperature, increase in humidity and cloud cover with an increase in wind speed, indicating unsuitable swimming conditions.
- From May to July a decline in temperature is seen, with the pool temperature that is almost equal to the heated water temperature, indicating unsuitable swimming conditions and poor results from the solar heating system. As

these months are during autumn and winter, it proofs that the pool motor can run for four hours as described in section 2.6.

,

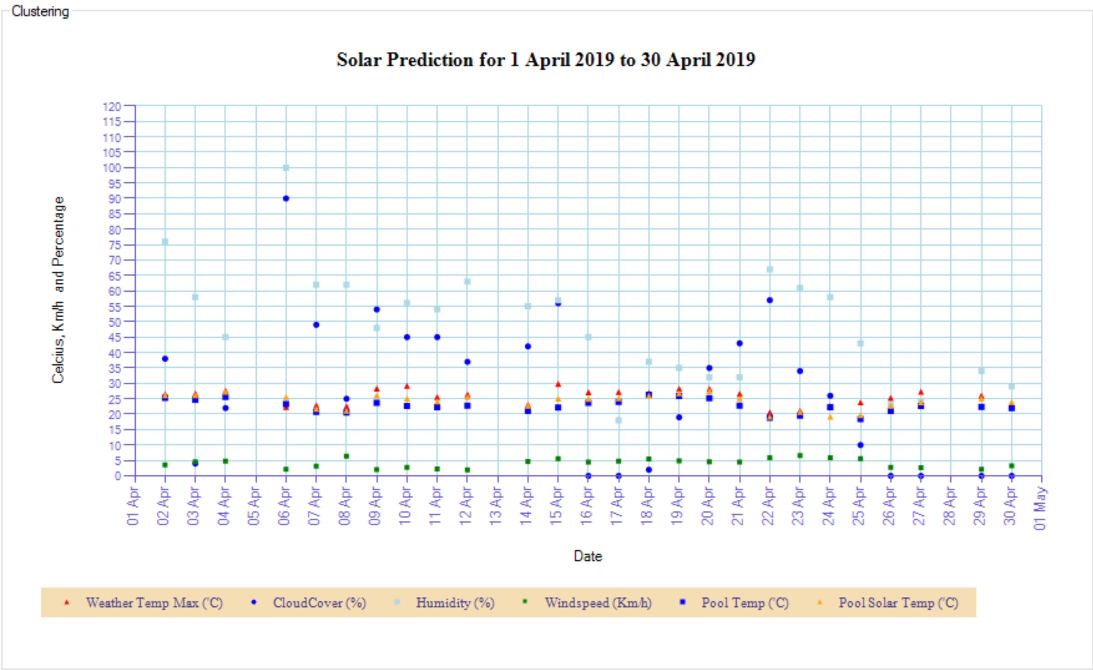


Figure 4.45 Daily recordings for weather predictions with pool and pool solar temperatures from April 2019

Partial clustering where used to determine the relationship between the heated solar water temperatures and weather predictions for maximum temperature, humidity, cloud cover and wind speed in section 4.6.1 to 4.6.4. Section 4.6.5 weather predicted values are compared with the difference in temperature between the pool water and the heated solar water.

4.6.1 Pool solar temperature vs Outside temperature

A comparison is done between the predicted maximum heated water and recorded outside temperature near the swimming pool with partial clustering as indicated in Figure 4.46 for April 2019.

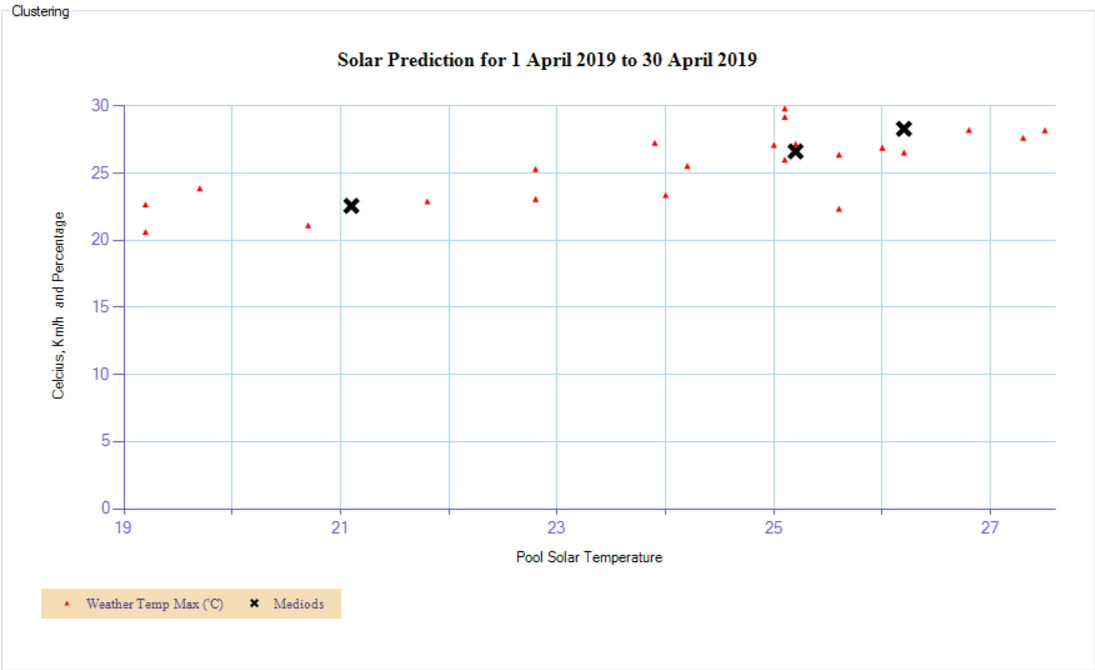


Figure 4.46 Partition clustering pool heated vs predicted maximum outside temperatures

Each medoid calculated and the weight of each medoid from March 2019 are displayed in Table 4.21. If only temperature influence the water heating, it is suspected that the heated water will always be higher than the predicted maximum temperature.

Table 4.21 Monthly pool heated vs predicted maximum outside temperatures

Month	Medoid1			Medoid2			Medoid3		
	Pool Solar Temperature	Outside Temperature	Persentage	Pool Solar Temperature	Outside Temperature	Persentage	Pool Solar Temperature	Outside Temperature	Persentage
Mar 2019	32.0	32.4	72.2	41.7	31.8	16.7	23.7	29.4	11.1
Apr 2019	25.2	26.6	46.2	21.1	22.6	30.8	26.2	28.3	23.1
May 2019	23.4	27.2	55.6	21.9	23.7	35.7	15.6	24.2	10.7
Jun 2019	19.5	24.6	44.0	18.2	22.4	36.0	17.7	19.8	20.0
Jul 2019	17.1	24.6	42.3	15.2	20.2	34.6	19.3	26.8	23.1
Aug 2019	22.3	27.3	56.7	19.3	23.5	23.3	23.4	29.3	20.0
Sep 2019	27.7	30.9	53.9	21.3	26.7	30.8	20.1	21.3	15.4
Oct 2019	28.4	33.4	56.0	31.9	35.3	28.0	23.3	25.5	16.0
Nov 2019	31.5	30.5	35.0	30.0	32.1	30.0	29.1	28.1	35.0
Dec 2019	32.0	32.9	38.9	27.6	29.5	33.3	18.6	18.7	27.8
Jan 2020	29.2	30.6	40.7	26.9	26.8	25.9	32.1	31.2	33.3

Investigating the table, the following were noted:

- During March the temperatures were 72.2% equal, 11.1% the heated water was 5°C lower and only 16.7% it where higher. If only temperature is used as indicated, only 3 days it was required to run the pump motor for longer than required to heat the pool water.
- During April all three medoids indicate that the heated water was lower than the outside temperature and indicate that it was not needed to run the pool motor for longer periods to heat the pool water.
- During May to Aug it where autumn and winter and the Medoids confirm the drop in temperatures and it is only needed to run the pool pump for four hours.

- During September to October during spring, an incline in temperatures was recorded, but pool temperatures still lower than outside temperatures. During October pleasant pool temperatures were recorded.
- From November higher pool temperatures than outside predicted temperature were recorded indicating pool pump must start operating maximum required hours.

Above statements appearances incorrect. Figure 4.47 proofs that the recorded outside temperature was between 2°C to 5°C lower, as the weather maximum predicted temperatures. The same trend was noted between predicted and recorded temperatures as to an increase or decrease in temperature from the previous day. During the second half of October a heat wave hit Gauteng as reported by the Weather services [47] explaining the abnormal high recorded temperatures and perfect swimming conditions.

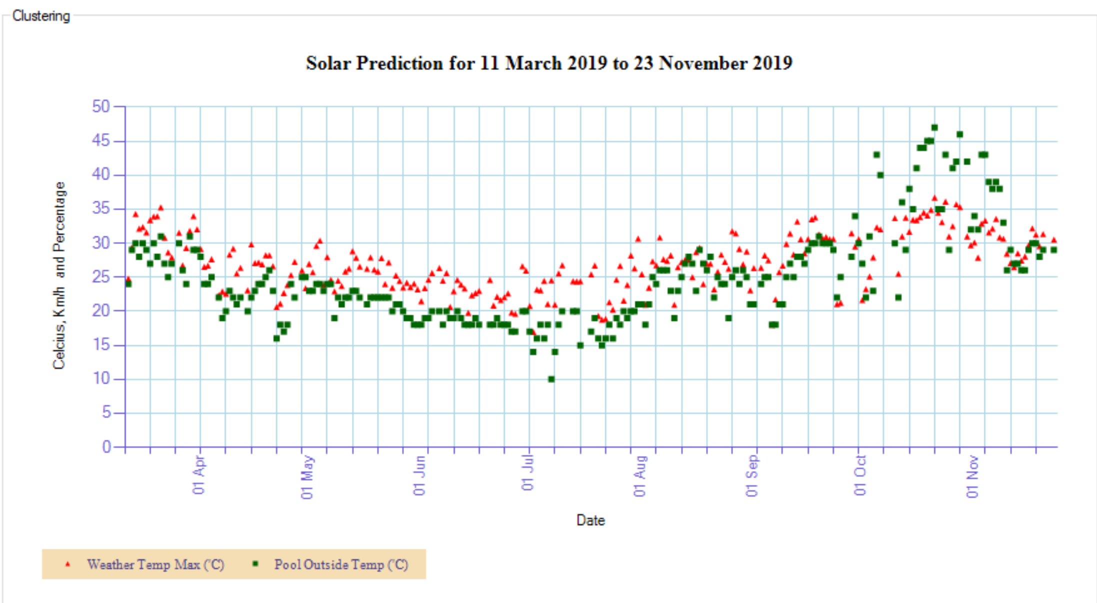


Figure 4.47 Comparison between recorded and predicted maximum temperatures.

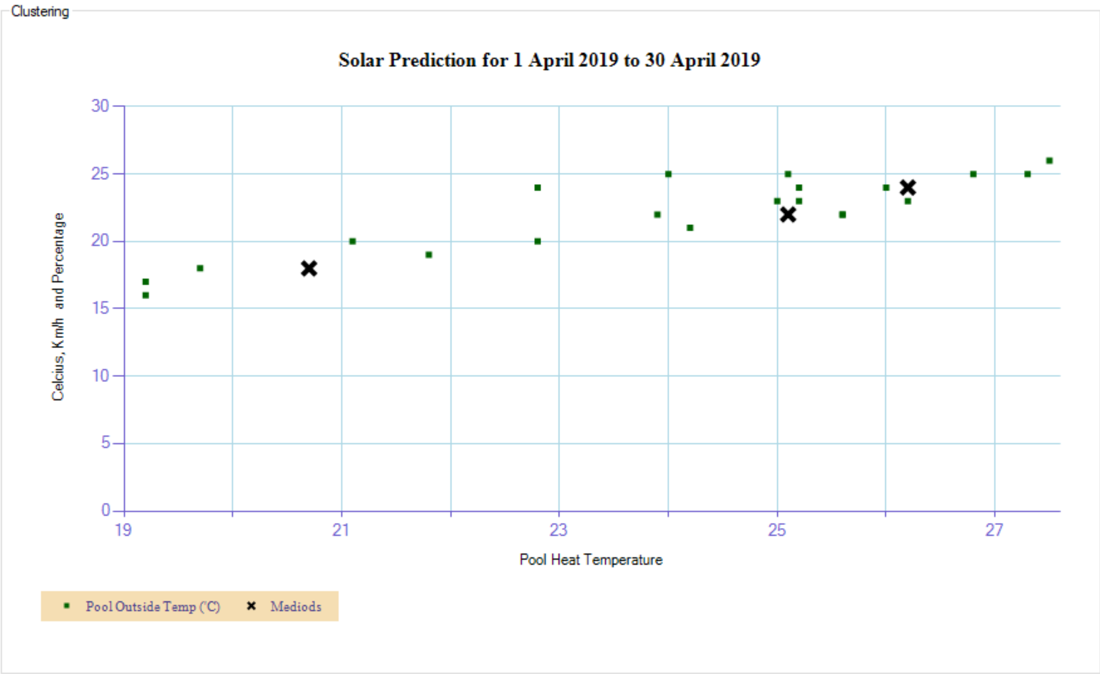


Figure 4.48 Partition clustering pool heated vs maximum outside temperatures

Table 4.22 Monthly pool heated temperatures vs maximum outside temperatures

Month	Medoid1			Medoid2			Medoid3		
	Pool Solar Temperature	Outside Temperature	Persentage	Pool Solar Temperature	Outside Temperature	Persentage	Pool Solar Temperature	Outside Temperature	Persentage
Mar 2019	30.9	28.0	50.0	34.3	30.0	33.3	43.4	29.0	16.7
Apr 2019	26.2	24.0	42.3	25.1	22.0	30.7	20.7	18.0	26.9
May 2019	23.0	22.0	48.3	20.3	19.0	27.6	24.3	24.0	24.1
Jun 2019	17.7	18.0	44.0	19.2	19.0	32.0	19.7	20.0	24.0
Jul 2019	15.2	16.0	38.5	17.6	18.0	34.6	19.2	20.0	26.9
Aug 2019	23.3	26.0	56.7	21.4	23.0	30.0	18.5	19.0	13.3
Sep 2019	27.7	30.0	53.9	24.7	25.0	26.9	21.3	21.0	19.2
Oct 2019	31.0	44.0	48.0	28.9	35.0	36.0	21.6	22.0	16.0
Nov 2019	31.2	29.0	45.0	31.3	39.0	30.0	29.1	26.0	25.0
Dec 2019	33.3	32.0	33.3	27.6	27.0	33.3	18.8	18.0	33.3
Jan 2020	32.1	29.0	33.3	29.2	27.0	33.3	26.9	25.0	33.3

Figure 4.48 illustrates a comparison between the recorded maximum and heated water temperature with partial clustering. Each medoid calculated and the weight of each medoid from March 2019 are displayed in Table 4.22.Investigating the table, the following was noted:

- During March the heated water temperature medoids were 2.9°C and 4.5°C higher than the outside temperature for 83% of the time and 14.3°C higher for 16% of the time indicating only 5 days it required to run the pump motor for longer than required to heat the pool water if needed. All three medoids water temperatures, were above 30°C and ambient temperature near 30°C indicting a high possibility for a pleasant swimming day.
- During April with a high possibility to rain and autumn around the corner, a drop in both temperatures is noted. The difference between heated water and ambient temperatures differ between 2-3°C. In April the prediction model can save cost as it is suspected that days suitable for swimming would have drop. Cloud cover and humidity needs to be taken into account.
- May to August is during autumn and winter, no prediction is required as the water and ambient temperatures are too cold for any outdoor swimming activities. Between June and August, it is noted that the maximum heated water temperature is lower than the ambient temperatures.
- September to October is during spring a steep incline in temperatures is noted with pool temperatures above 27°C.
- During November 45% of the month and 100% during December and January, the pool solar temperatures were higher that the outside temperatures indicating perfect swimming conditions.

4.6.2 **Pool solar temperature vs Cloud cover**

A comparison is done between the predicted cloud cover and recorded heated water temperature with partial clustering as indicated in Figure 4.49 for April 2019.

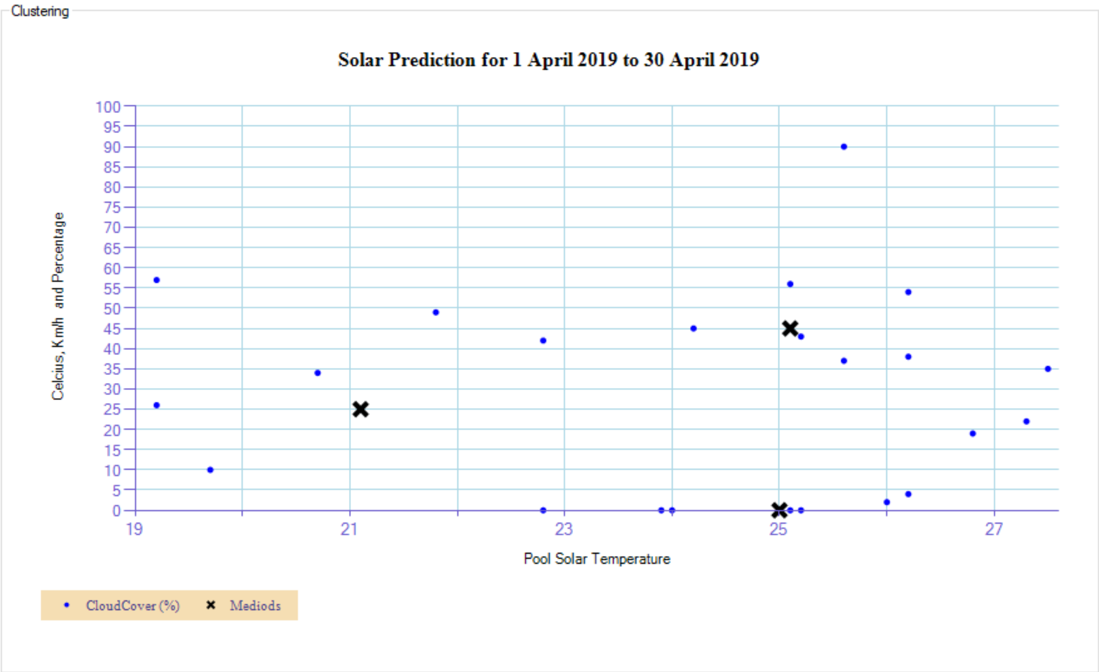


Figure 4.49 Partition clustering pool heated water temperatures vs cloud cover

Each medoid calculated and the weight of each medoid from March 2019 are displayed in Table 4.23. It is suspected that the heated water will decrease with an increase in cloud cover. A higher cloud cover does not indicate unsuitable rainy days, but only the efficiency running the pool motor to heat the pool water. Humidity and cloud cover together are a better indication for rainy days.

Table 4.23 Monthly pool heated water temperatures vs cloud cover

	Pool Solar Temperature	Cloud Cover	Persentage	Pool Solar Temperature	Cloud Cover	Persentage	Pool Solar Temperature	Cloud Cover	Persentage
Mar 2019	34.3	0.0	55.6	31.2	26.0	33.3	32.9	60.0	11.1
Apr 2019	25.1	45.0	46.2	25.0	0.0	34.6	21.1	25.0	19.2
May 2019	23.1	0.0	82.8	21.7	31.0	13.8	21.0	79.0	3.5
Jun 2019	18.2	0.0	72.0	18.4	4.0	20.0	20.9	48.0	8.0
Jul 2019	18.0	0.0	57.7	15.2	0.0	38.5	16.8	14.0	3.9
Aug 2019	22.7	0.0	80.0	21.0	7.0	16.7	21.7	26.0	3.3
Sep 2019	27.2	0.0	69.2	24.9	24.0	19.2	17.2	81.0	11.5
Oct 2019	28.5	31.0	53.9	28.4	0.0	26.9	27.7	59.0	19.2
Nov 2019	31.0	40.0	50.0	31.7	88.0	30.0	29.1	0.0	20.0
Dec 2019	29.1	13.0	41.2	19.1	98.0	41.2	31.3	31.0	17.7
Jan 2020	29.3	38.0	48.2	30.1	4.0	29.6	27.0	79.0	22.2

Investigating the table, the following was noted:

- For March there is no proof that an increase in cloud cover did impact the efficiency of the solar solution. The maximum temperature per day only differs 1.4°C for 0% and 60% cloud cover.
- During April it is noted that the heated temperatures are the same for 0% and 45% cloud cover, but the temperature is 4°C lower with only 25% cloud cover. This proof, that the prediction model must include multiple weather conditions.
- From May the temperature did drop because it is during autumn and winter.
- During November 30% of the time the cloud cover was above 88%, but with a higher pool solar temperature.

Above findings proof that humidity must also be investigated.

4.6.3 Pool solar temperature vs Humidity

A comparison is made between the predicted humidity and recorded heated water temperature with partial clustering as indicated in Figure 4.50 and Table 4.24 for April 2019.

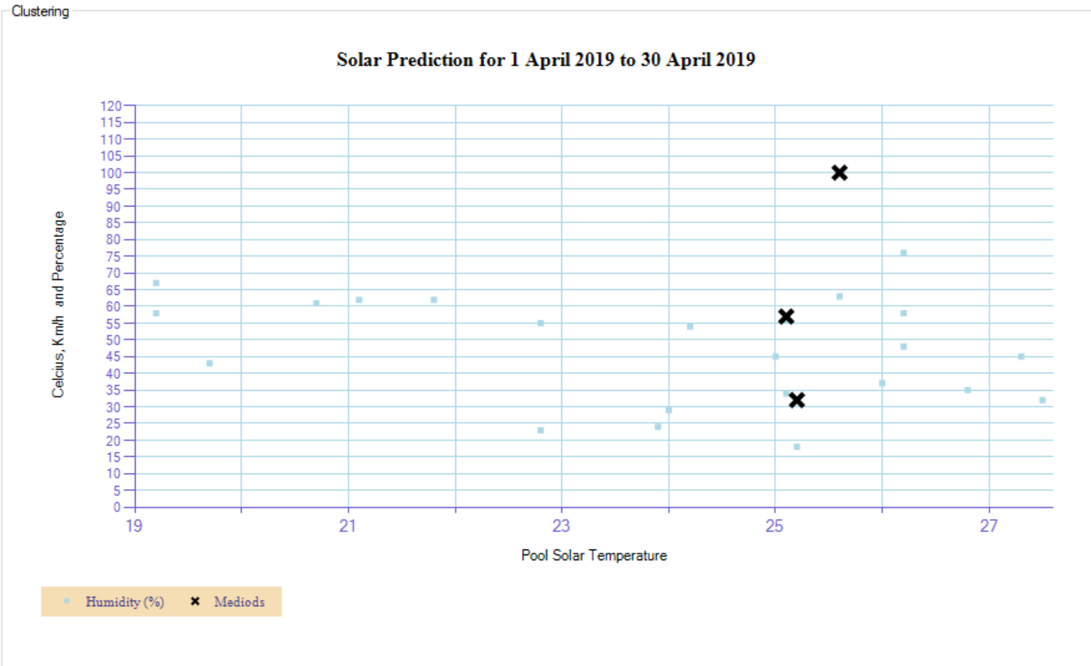


Figure 4.50 Partition clustering pool heated water temperatures vs humidity

Each medoid calculated and the weight of each medoid from March 2019 are displayed in Table 4.24. It is suspected that the heated water will decrease with an increase in humidity. Investigating the table, the following was noted:

- During March, an increase in humidity did decrease the heated water temperatures.
- For April humidity cannot be used as in indication as the heated water temperatures were the same for all three medoids.

Table 4.24 Monthly pool heated water temperatures vs humidity

Month	Medoid1			Medoid2			Medoid3		
	Pool Solar Temperature	Humidity	Persentage	Pool Solar Temperature	Humidity	Persentage	Pool Solar Temperature	Humidity	Persentage
Mar 2019	34.3	28.0	61.1	31.7	44.0	27.8	23.7	72.0	11.1
Apr 2019	25.1	57.0	57.7	25.2	32.0	38.5	25.6	100.0	3.85
May 2019	23.2	20.0	55.2	22.7	26.0	37.9	21.7	43.0	6.9
Jun 2019	19.2	17.0	64.0	17.7	28.0	24.0	16.9	43.0	12.0
Jul 2019	17.6	18.0	50.0	17.7	11.0	26.9	15.2	31.0	23.1
Aug 2019	22.3	20.0	50.0	23.8	9.0	26.7	20.1	34.0	23.3
Sep 2019	27.2	13.0	61.5	20.1	18.0	30.7	20.8	53.0	7.7
Oct 2019	28.2	22.0	42.3	31.8	14.0	34.6	29.8	33.0	23.1
Nov 2019	29.9	47.0	45.0	30.8	18.0	30.0	29.2	29.0	25.0
Dec 2019	28.4	44.0	35.3	33.3	23.0	35.3	18.6	78.0	29.4
Jan 2020	31.2	38.0	44.4	28.2	53.0	37.0	29.1	22.0	18.5

- From May the temperature did drop because it is during autumn and winter.
- From September higher solar temperatures were recorded as without a slight increase in humidity.

As seen, there are still conditions that influence the heating system that was not investigated until now. If the pool water is lower, it will influence the outcome of the solar heating.

4.6.4 **Pool temperature vs Wind speed**

As discussed in section 2.6 wind speeds can cool down the pool surface water and increase water loss through evaporation. Including wind speed in the prediction model is more complex as it is an important factor that inject dirt and leaves into the pool. A balance must be found between cleaning and evaporation. A user preferred variable will be needed in the prediction model.

Table 4.25 illustrates the Beaufort scale, which is the most popular scale defining wind speeds and are used in weather forecasts [48, p.17]. During the research it was noted that the pool water was dirtier with a Beaufort scale 4 and higher.

Table 4.25 Beaufort wind speed scale

Scale	Designation	Wind speed m/s	Effect
0	Calm	< 0.3	Nothing
1	Light air	0.3 - 1.5	Diversion of smoke
2	Light breeze	1.6 - 3.3	Contractions of leaves
3	Gentle breeze	3.4 - 5.4	Movement of branches
4	Moderate breeze	5.5 - 7.9	Movement of limbs
5	Fresh breeze	8.0 - 10.7	Movement of small trees
6	Strong breeze	10.8 - 13.8	Movement of mall branches
7	High wind	13.9 - 17.1	Movement of trees
8	Gale	17.2 - 20.7	Difficulty in walking
9	String gale	20.8 - 24.4	House damage
10	Strom	24.5 - 28.4	Uprooting of trees
11	Violemt strorm	28.5 - 32.6	Storm damage
12	Hurricane	> 32.7	Devastation

A comparison is done between the predicted wind speed and recorded pool water temperature with partial clustering as indicated in Figure 4.51 during April 2019.

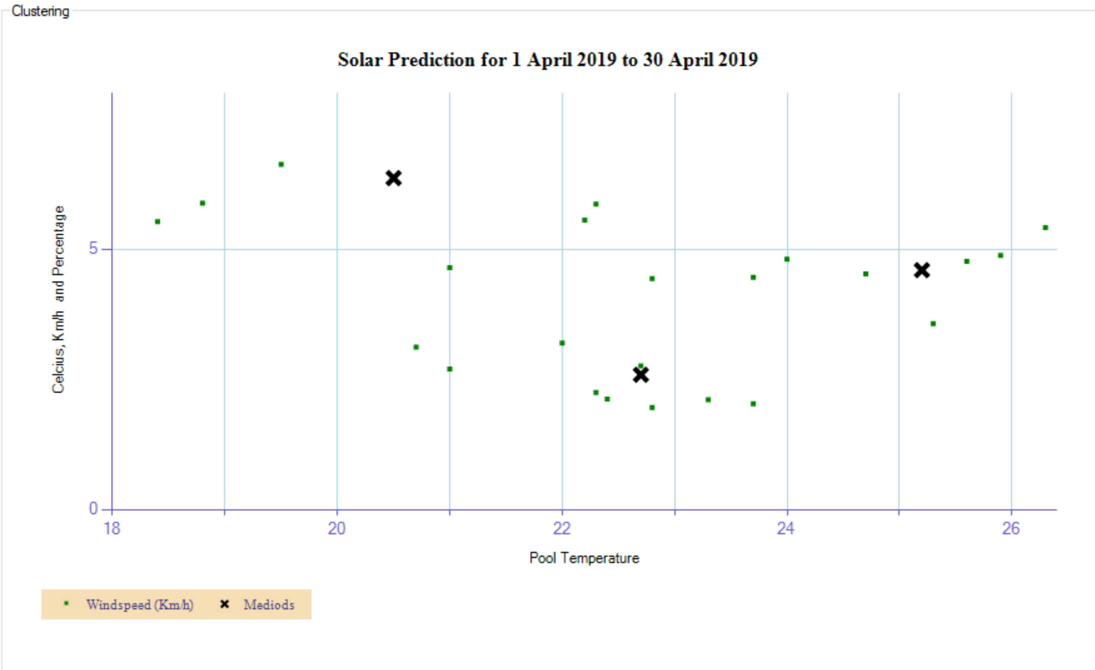


Figure 4.51 Partition clustering pool water temperatures vs wind speed

Table 4.26 Monthly pool water temperatures vs wind speed

Month	Medoid1			Medoid2			Medoid3		
	Pool Temperature	Wind speed	Persentage	Pool Temperature	Wind speed	Persentage	Pool Temperature	Wind speed	Persentage
Mar 2019	27.5	4.7	44.4	29.8	4.3	33.3	28.4	2.7	22.2
Apr 2019	22.7	2.6	42.3	25.2	4.6	30.8	20.5	6.4	26.9
May 2019	22.2	1.8	51.7	20.6	3.9	31.0	16.6	1.7	17.2
Jun 2019	17.1	3.7	36.0	18.8	2.2	32.0	18.2	4.5	32.0
Jul 2019	17.6	4.4	42.3	14.8	5.5	34.6	19.2	3.6	23.1
Aug 2019	21.5	3.7	43.3	22.2	6.5	33.3	19.9	6.4	23.3
Sep 2019	28.7	6.2	38.5	21.9	6.4	34.6	24.8	4.0	29.9
Oct 2019	26.9	3.4	38.5	27.4	7.3	30.8	22.8	5.7	30.8
Nov 2019	29.9	5.0	50.0	31.8	6.0	35.0	24.6	7.0	15.0
Dec 2019	19.2	3.8	41.2	29.2	5.5	35.3	34.5	4.8	23.5
Jan 2020	28.9	4.3	38.5	26.5	5.8	30.8	32.5	3.9	30.8

Investigating Table 4.26, the following were noted:

- All the medoids with a Beaufort scale of four is highlighted in red font.
- Pool temperatures are marked in red where the wind speed has a scale of four with a drop in temperatures.
- As seen, there is a decline in temperature, but an important factor needs to be built into the model. Running the pump cleaning the pool while cooling down the water. This factor will be different for each pool dependent on surrounding areas around the pool for example type of flower bed and trees.

4.6.5 Pool solar temperature vs Pool temperature

In section 4.6.1 to 4.6.3 the heated temperature was compared against predicted weather values. The most important factor is when the pump motor is running does the water get heated or not and by how much.

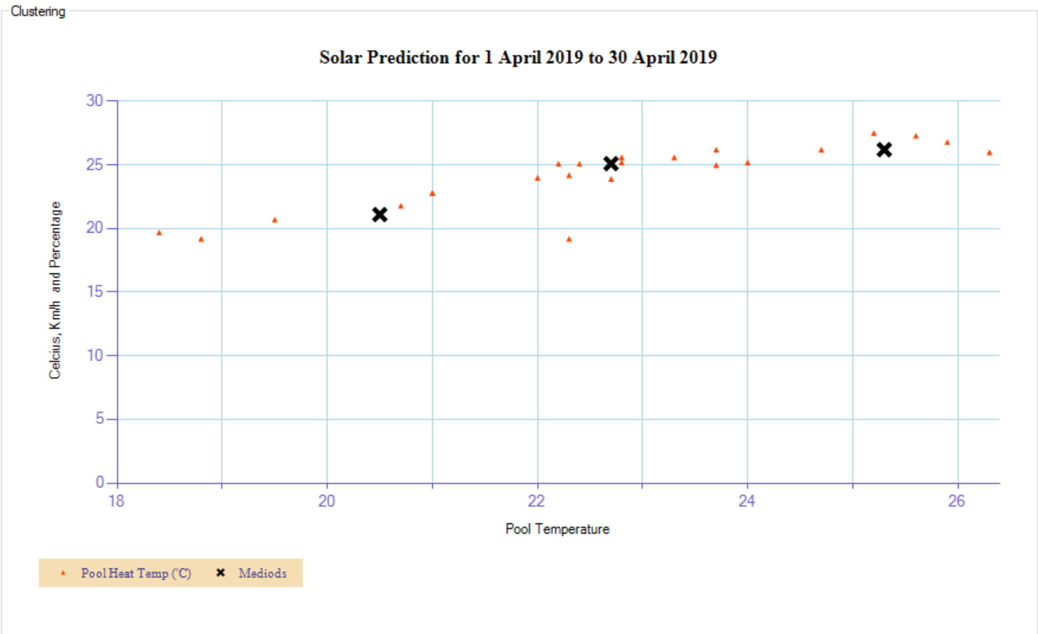


Figure 4.52 Partition clustering pool water vs heated water temperatures

A comparison is made between the recorded pool water and heated water temperatures with partial clustering as indicated in Figure 4.52 for April 2019.

Table 4.27 Monthly pool water vs solar heated temperatures

Month		Medoid1				Medoid2				Medoid3		
	Pool Temperature	Pool Solar Temperature	Delta	Persentage	Pool Temperature	Pool Solar Temperature	Delta	Persentage	Pool Temperature	Pool Solar Temperature	Delta	Persentage
Mar 2019	28.2	31.2	3	55.6	29.2	34.4	5	27.8	27.5	42.1	15	16.7
Apr 2019	22.8	25.1	2	42.3	20.5	21.1	1	30.8	25.3	26.2	1	26.9
May 2019	22.5	23.6	1	55.2	20.1	21.9	2	37.5	15.8	16.0	0	10.3
Jun 2019	17.7	18.2	1	28.0	16.9	17.7	1	28.0	18.8	19.5	1	44.0
Jul 2019	17.1	17.5	0	38.5	15.3	15.2	0	34.6	18.9	18.6	0	26.9
Aug 2019	23.3	23.4	0	43.3	21.0	21.4	0	40.0	19.2	19.3	0	16.7
Sep 2019	28.7	27.8	1	34.6	22.3	20.3	2	34.6	24.8	24.9	0	30.8
Oct 2019	29.9	28.4	-2	38.5	26.9	28.4	-2	38.5	24.2	23.4	-1	23.1
Nov 2019	29.7	29.2	0	55.0	31.8	31.5	0	35.0	22.4	31.1	9	10.0
Dec 2019	19.3	19.1	0	41.2	34.2	33.3	-1	29.4	28.6	28.4	0	29.4
Jan 2020	32.1	31.7	0	34.6	26.2	27.0	1	34.6	29.2	29.3	0	30.8

A comparison is made between the recorded pool water and heated water temperatures with partial clustering as indicated in Figure 4.52 for April 2019.

Table 4.27 the following were noted:

- The delta in temperature was included rounded to the nearest integer value.
- During March the lowest delta in temperature was 3°C with a weight of 55.6% and an impressive 15°C with a weight of 16.7%, indicating March was a perfect month for heating the pool water
- During April and May the pool temperature did drop with only 2°C in delta temperatures. Only 26.9% of April the pool temperature as about 25°C.

- From May the pool water temperatures did decrease to unpleasant swimming experience with solar heating having no influence on the pool temperatures.
- From September the temperatures starting to increase but with the delta temperature only differ with a maximum of 2°C.
- The only exception is during November where the delta went up to 9°C.
- During November and December the delta temperature is not higher as expected. The pool water temperature was the same as the outside temperature and as highlighted in red, sometimes higher than the water from the solar panels.

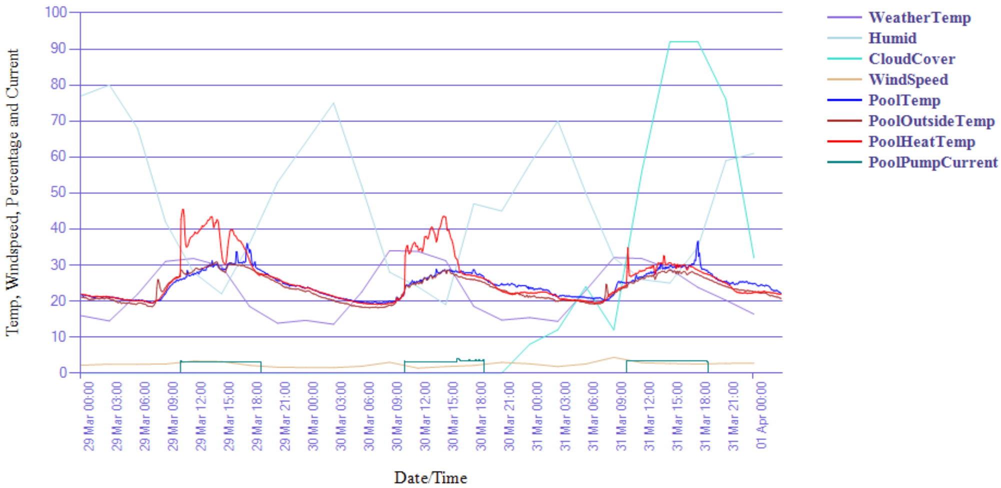


Figure 4.53 Weather predictions against Pool temperature recordings

Previous daily and weekly data will be used in the prediction model as previous year readings do not indicate any accurate patterns. One reason for this can be that one season is dry, and the next season wet. Figure 4.53 illustrates how cloud cover on the 31 March 2019 could be used with advantage to determine that the heat efficiency would be lower.

4.7 *Conclusion*

Fuzzy logics were used to fuzzify all the geyser-related recorded temperatures to linguistic term for usage as None, Low, Medium and High. The cool-down geyser temperature, when no heated water is used, can be calculated hours in advance, if the predicted ambient temperature is known. To ensure the water temperature is at the correct temperature when the timeslot started, the duration to hot water to a specific temperature must be known.

Predicting hot water usages, the system must take the following into account:

1. Change in time when hot water is used;
2. With the addition of a solar collector the weather predictions temperature, humidity and cloud cover;
3. Occupation changes.

Predicting hot water heated by a solar collector, the roof temperature was used as a reference instead of the geyser temperature, as it can change if electrical energy is added to the system via the backup element. The maximum ambient temperatures, average humidity and cloud cover must be used for an accurate prediction module. Previous daily and weekly data will be used in the prediction model as previous year readings do not indicate any accurate patterns. The relationship between the roof and geyser was more optimal, determined by using the previous seven day results rather than the daily results.

The same model heating geyser water can be used to heat swimming pool water. Wind speed must be included in the model heating and cleaning pool water as it cools down surface water, but adds dirt and leaves in the pool. The optimal prediction model must include the medoid percentage or weight values.

Chapter 5

5 Developing the prediction model

Investigating multiple models and algorithms the Hidden Markov model, as discussed in section 2.4, has the best characteristics solving the prediction model. The model with Fuzzy logic and partition clustering, as discussed in section 2.1 and 2.2.2, summarised all the big data. A suitable solution to be developed is discussed in this chapter, providing the required outputs from the weather predictions, solar water heated and residence behaviours.

5.1 *Basic layout*

A high-level block diagram of the prediction model is illustrated in Figure 5.1.

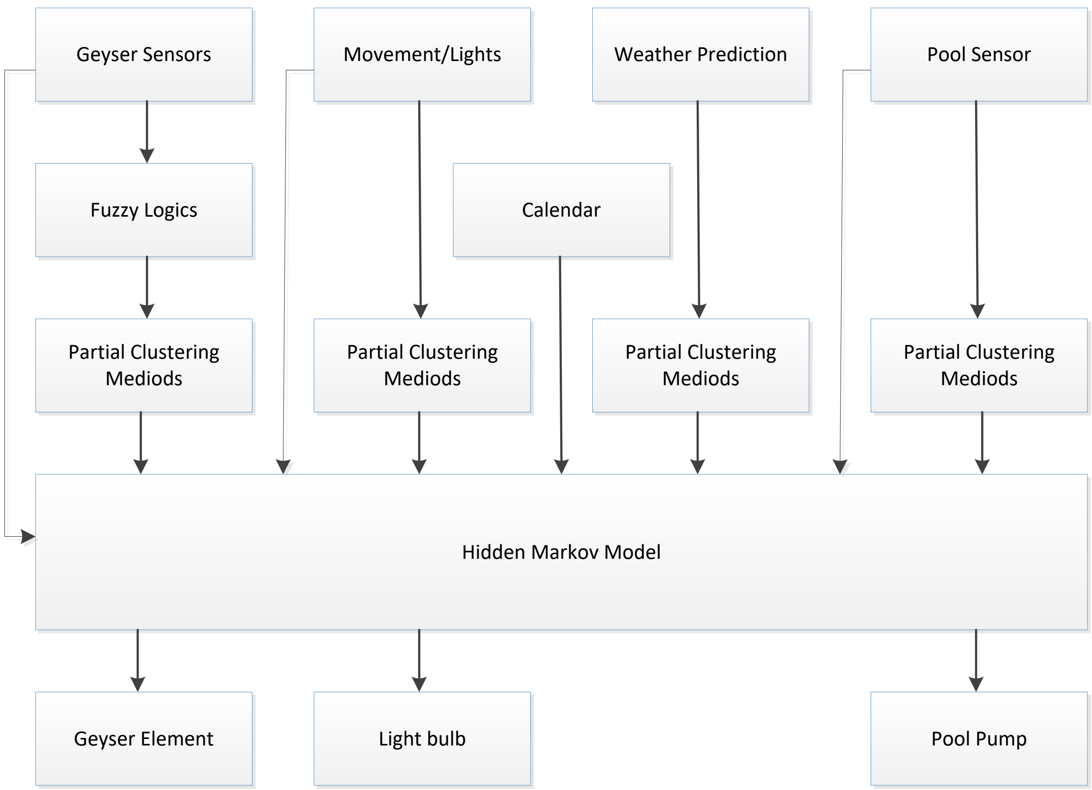


Figure 5.1 Basic block diagram of prediction algorithm.

Combined sensor reading from the pool, geyser and lighting environment together with month of year and weather predictions will be used to control a light bulb, geyser element and pool pump.

5.2 *Weather predictions predicting solar heating*

Wind strength will not be included in the weather model, but in the pool model as it tends to be a personal choice for pools only and have no influence to the solar geyser functionality as discussed in section 4.6.4. The model will include maximum temperature, cloud cover and humidity as the observation states and the radiation heating water will be the hidden states.

Analysing the data grouped together in Chapter 4 with partial clustering three medoids was chosen. Having three states, “Low”, “Medium” and “High”, per observation state would result into $3^3=27$ observation states making the module complex as seen in Figure 5.2.

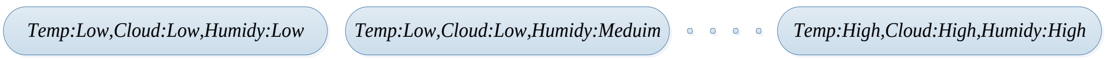


Figure 5.2 First proposed weather observation states.

By investigating Figure 4.44 and Figure 5.3, high humidity and cloud cover can influence the heated pool water and roof temperatures. These diagrams indicated that initial value should be around 70% and higher for High.

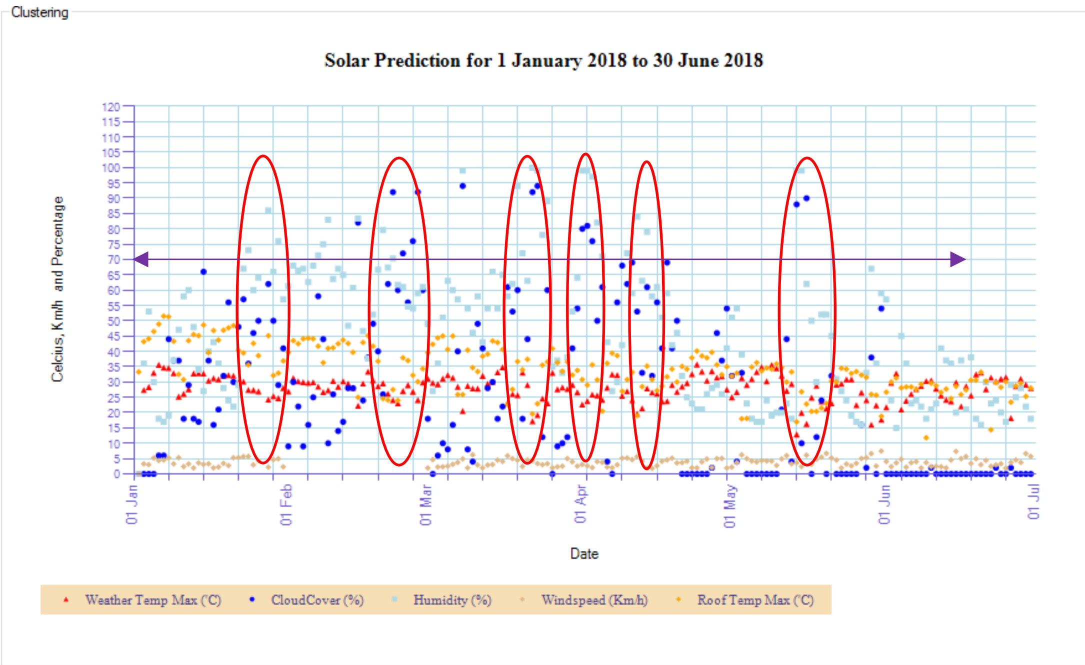


Figure 5.3 Daily recordings for weather predictions with geyser roof temperatures from January to June 2018.

The daily maximum temperature, cloud cover, humidity, wind speed and roof temperature for 2018 are shown in Figure 5.4. The green line illustrates the maximum daily cloud cover and humidity trends, the red line the maximum temperatures and the orange line the recorded maximum roof temperatures. Cloud cover and humidity does not have any significant influence during the colder winter month to the roof temperatures. Including the findings made in Table 4.11 and Figure 5.3 the following three seasons were classified:

Rainy season: High {Jan, Feb, Mar, Apr} where Cloud cover and Humidity goes above 70%

Rainy season: Low {Oct, Nov, Dec} where only Cloud cover goes above 70% and

Rainy season: None {May, Jun, Jul, Aug, Sep} where Cloud Cover and Humidity readings are below 70% on average.

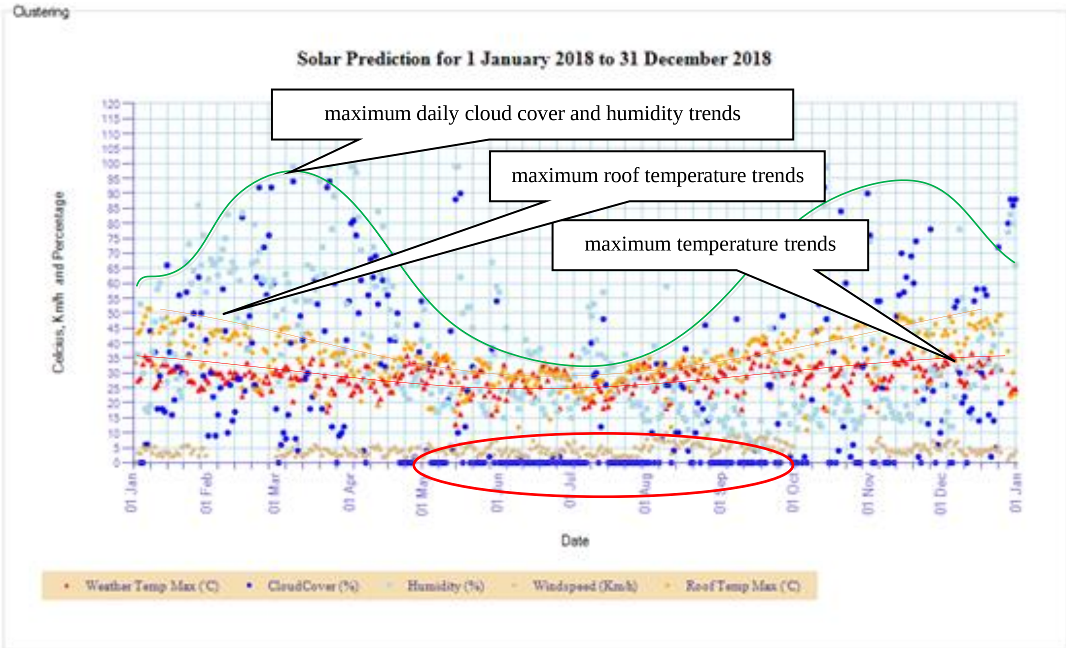


Figure 5.4 Daily recordings for weather predictions with geyser roof temperatures from 2018.

An assumption was made to classify cloud cover and humidity above 70% together as cloudy, preventing using multiple order HMM. The observation states of weather are:

- Cloudy, Any Temperature
- No Clouds, High Temperature
- No Clouds, Mild Temperature
- No Clouds, Low Temperature

Each season requires each own HMM modem with observation states. The hidden state, radiation to heat water, will be {Good, Average, Bad}. Radiation cannot be measured and the difference in water temperature was taken before and after the solar water heater, as indication to measure the radiation via heat transferred to the pool water. Historical weather, pool and geyser solar heating will be used to train the model. Firstly, the hidden solar radiation state will be measured as the inside roof temperature as Figure 4.5 illustrates the parallel relationship between the heating of the geyser water and the inside roof temperature.

All readings must be normalised making it usable for the hidden Markov model. Cloud cover and humidity are already in percentage format. Normalising the temperature values, the highest maximum predicted temperature between 2017 and 2019 was 40°C on 18 September 2018 and the lowest maximum temperature was 12.9°C on 14 May 2018. There was a heat wave in the area as the temperature is abnormal in the area as shown in Figure 5.5. The maximum wind speed predicted was 9.86km/h on 9 August 2018. Another observation forms the geyser sensor is the roof temperature to the maximum of 51.5°C recorded on 7 January 2018.

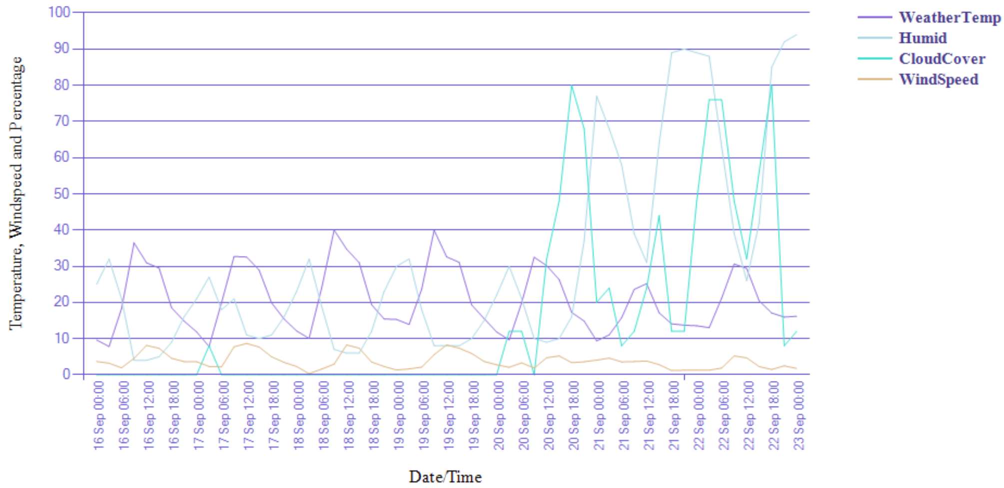


Figure 5.5 Weather Predictions between 16 and 22 September 2018

The expected outcome from the HMM is the probability to maximise the heating process with solar radiation from weather predictions. Creating the HMM an assumption will be made that the normalised values between 0 and 0.32 will be Low, 0.33 to 0.65 Medium and 0.66 to 1 High for temperatures and 0 to 0.69 Low and 0.8 to 1 High for humidity and cloud cover.

5.3 *Calculating Hidden Markov Model state values for Geyser and Pool solution*

Figure 5.6 illustrates the basic flow chart for the software procedure used in Appendix J calculating the HMM initial, transition and emission probability distribution states from normalised database values.

The procedure requires the first and last date from which it must retrieve the normalised values from the database. It also requires the rainy season and if it must calculate the pool or roof HMM model using the same procedure.

The entry “Pool start” and “Roof start” illustrated in Figure 5.7 in more detail where each value from the database is compared with previous day’s values. After each value was investigated,the final HMM states are calculated in “Calculate HMM values start”, as shown in more detail in Figure 5.9.

Lastly, all calculations will be saved according to rain season and Roof or Pool solution.

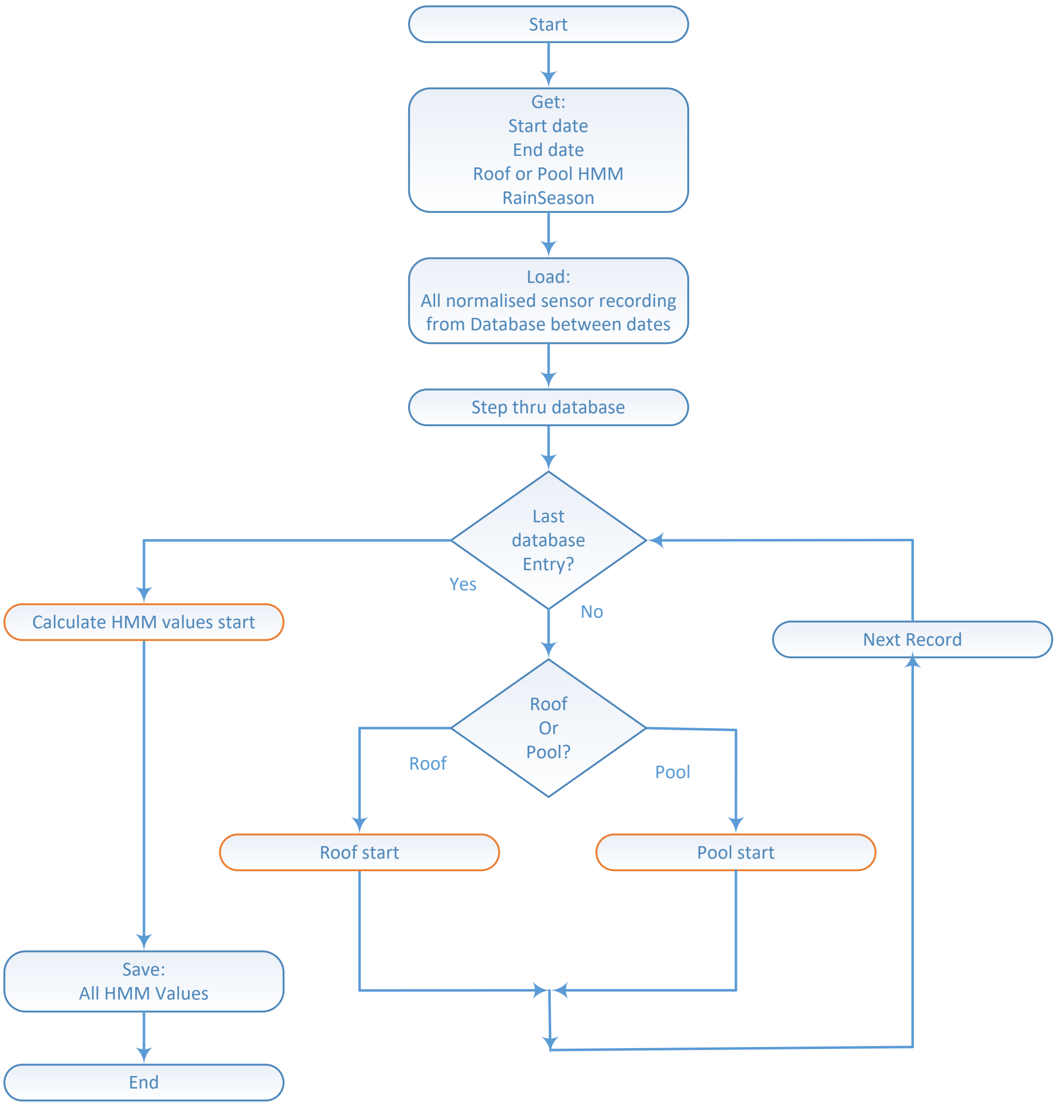


Figure 5.6 Flow chart one to calculate HMM Transition and Emission states

Figure 5.7 illustrates the process categorised each normalised database entry to the correct HMM state. This procedure can be used for the pool and roof temperature HMM model where the roof entries are replaced with the pool entries. Firstly, all Roof and Weather (Roof related) values higher than 0.6 will be classified as High values, 0.4 and 0.59 Mild and below 0.4 Low. All Pool values higher than 0.75 will be classified as High values, 0.55 and 0.74 Mild and below 0.55 Low. All Weather (Pool related) values higher than 0.5 will be classified as High values, 0.35 and 0.49 Mild and below 0.35 Low.

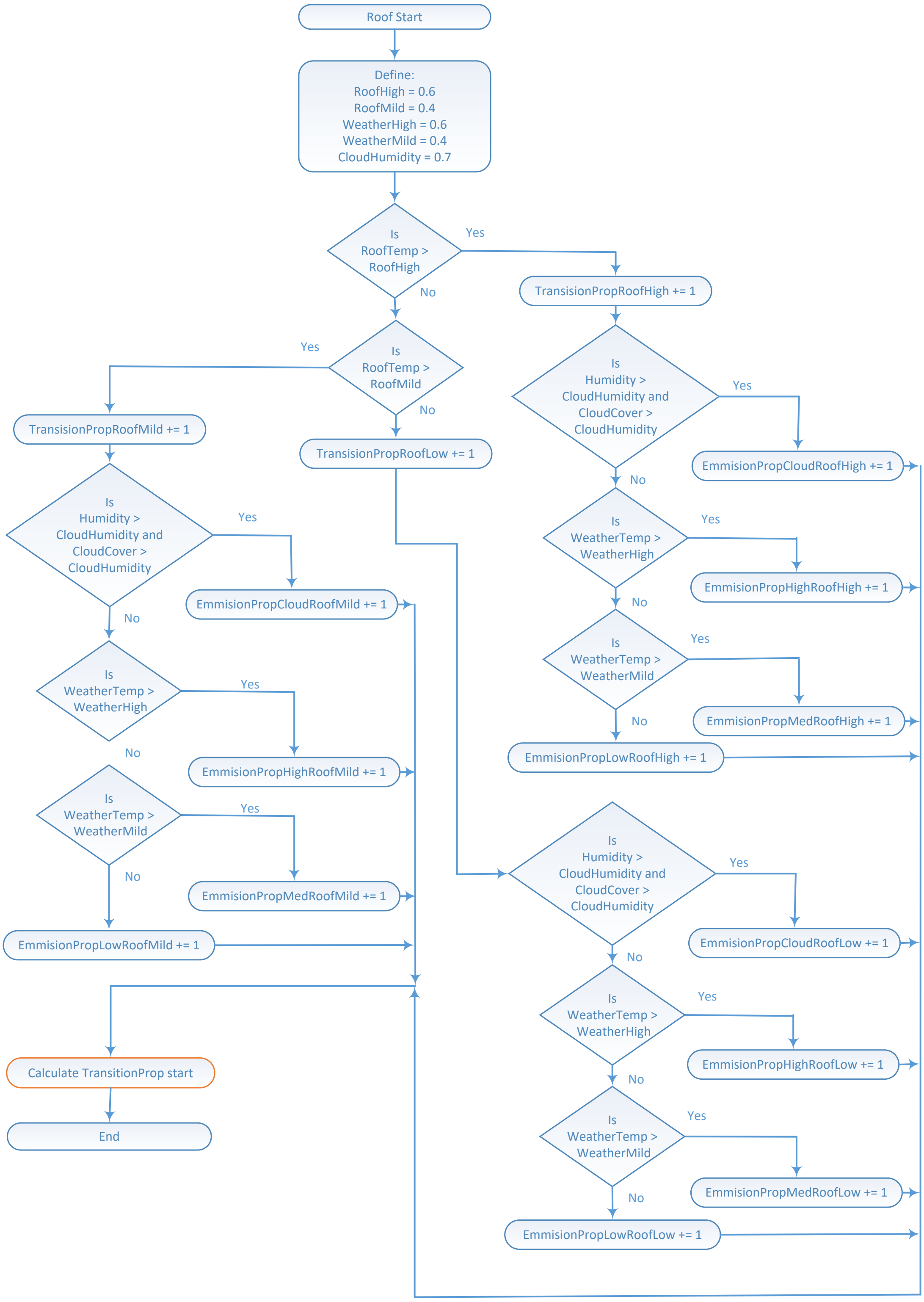


Figure 5.7 Flow chart two to calculate HMM Transition and Emission states

As discussed in section 5.2 and Figure 5.3, cloud and humidity entries higher than 0.7 together will be classified as cloudy emission state.

Weather conditions are marked as emission probabilities [Cloudy, Temp High, Temp Mild, Temp Low] per Roof or Pool transition probability [High, Mild, Low].

Entry “Calculate TransitionProb start” is illustrated in Figure 5.8 in more detail. Calculate the transition probability for the roof or pool temperatures, the difference between the previous and current days state must be recorded as;

[High] to [High, Mild, Low];

[Mild] to [High; Mild; Low] and

[Low] to [High; Mild; Low].

After all database entries were classified the final HMM calculations are done as shown in Figure 4.9.

Firstly, the initial probability are calculated for the roof or pool temperatures been high, medium or low, as discussed in section 2.4.1. The mathematical formula for the complete process is described at equation 2.76 is $\pi_i = P[q_1 = s_i]$.

Secondly the transition probabilities are calculated for the roof or pool temperatures as HighHigh, HighMild, HighLow, MildHigh, MildMed, MildLow, LowHigh, LowMild and LowLow as discussed in section 2.4.1. The mathematical formula for the complete process are described at equation 2.75 is $a_{ij} = P[q_t = S_j \mid {}_{qt-1} = S_i$.

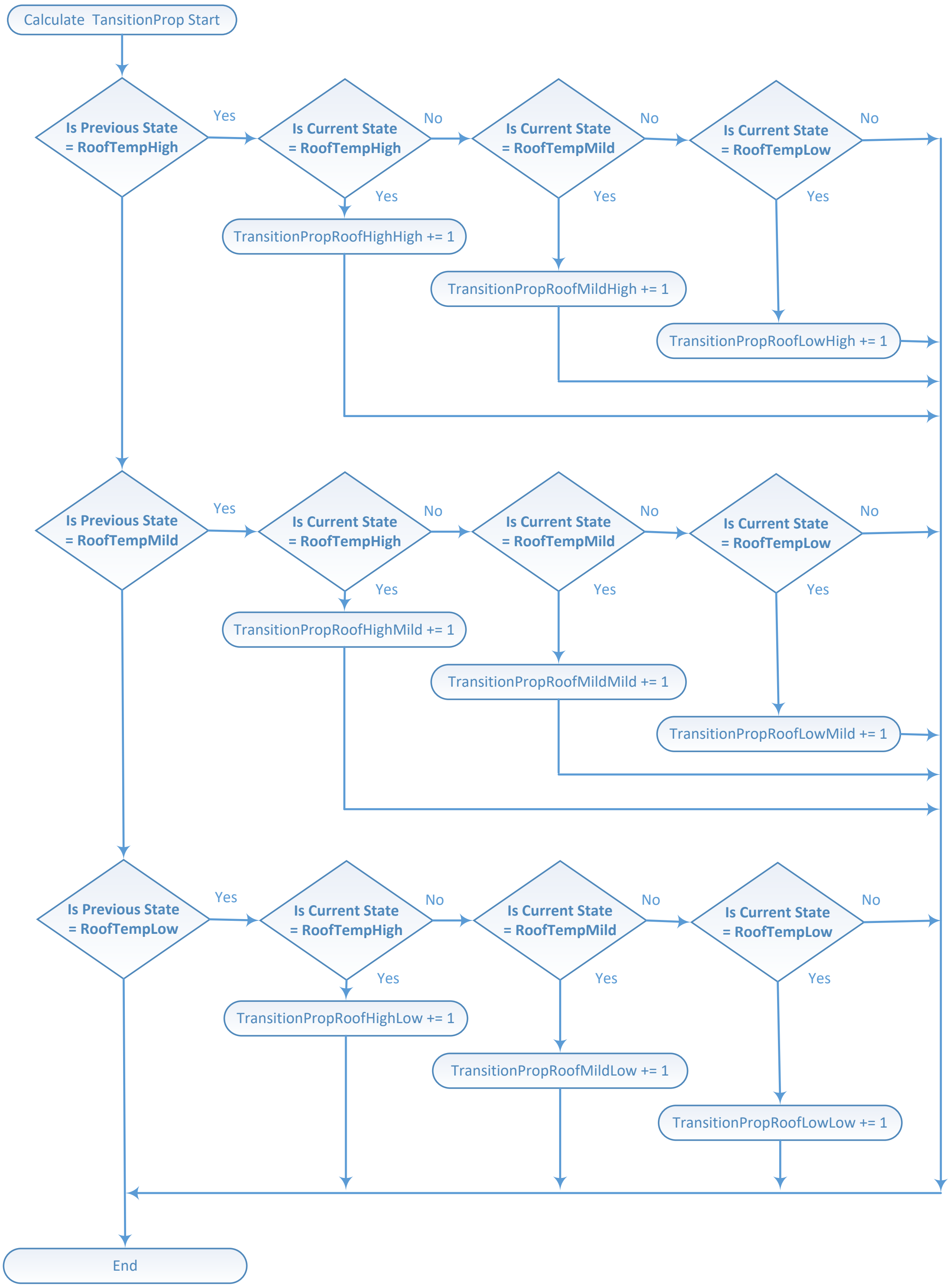


Figure 5.8 *Flow chart three to calculate HMM Transition and Emission states*

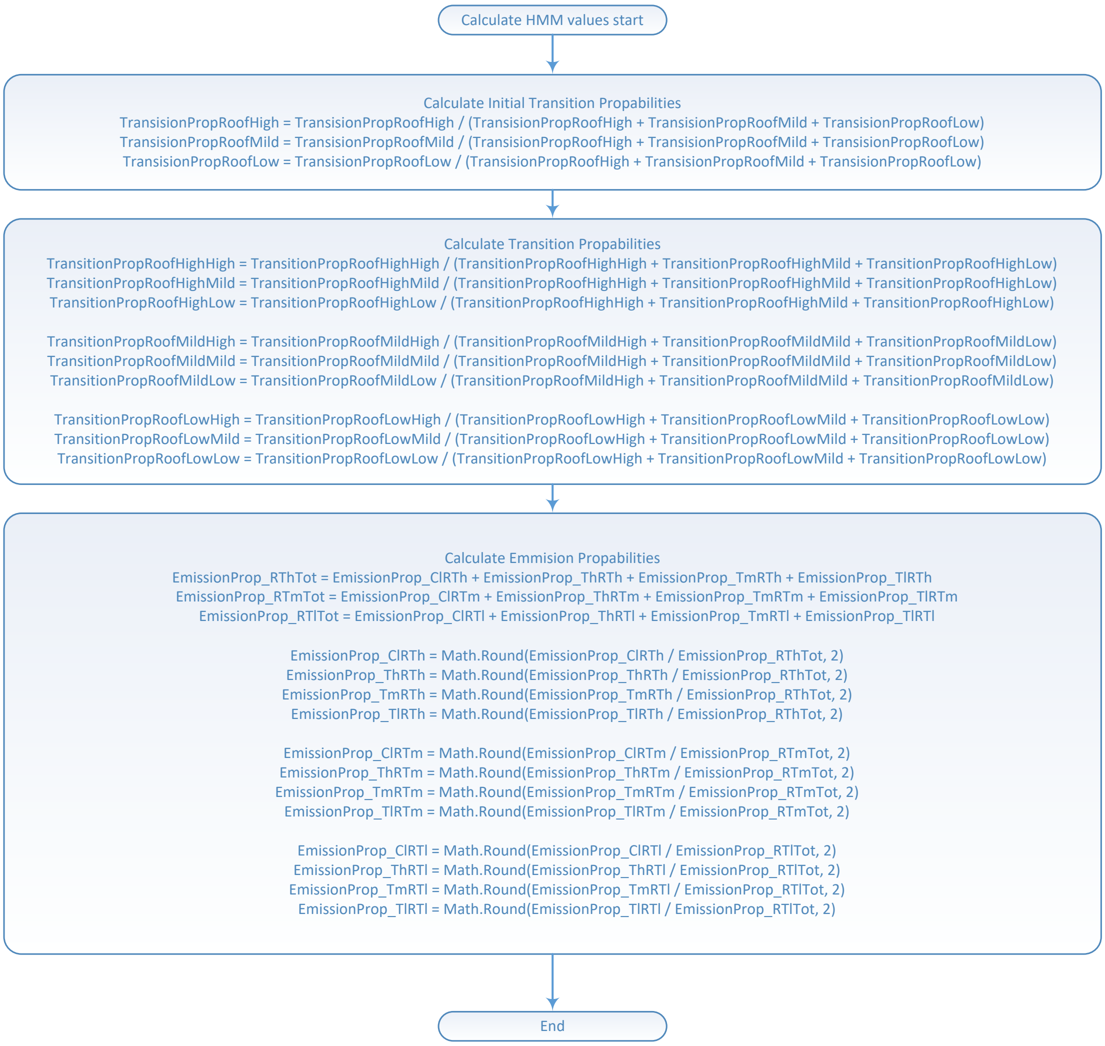


Figure 5.9 Flow chart calculate four to HMM Transition and Emission states

Lastly, the emission probabilities are calculated where the Roof or Pool temperatures are [High, Mild, Low] and the weather condition are [Cloudy, High, Mild, Low] as discussed in section 2.4.2. The mathematical formula for the complete process is described at equation 2.79 is $b_j(k) = P[v_k \text{ at } t \mid q_t = S_j]$.

The HMM transition and emission probability state diagrams and matrices for each rainy season are shown in section 5.6 for the geyser and pool in section 5.5.

5.4 *Calculating Viterbi prediction for Geyser and Pool solution*

The Viterbi algorithm is used to predict the roof and pool temperatures based on the weather predictions from HMM states calculated per rainy season. Figure 5.10 illustrates the basic flow chart for the software procedure used in Appendix K calculating the pool and roof temperature five day predictions to be [High, Mild, Low] with the Verbiti algorithm.

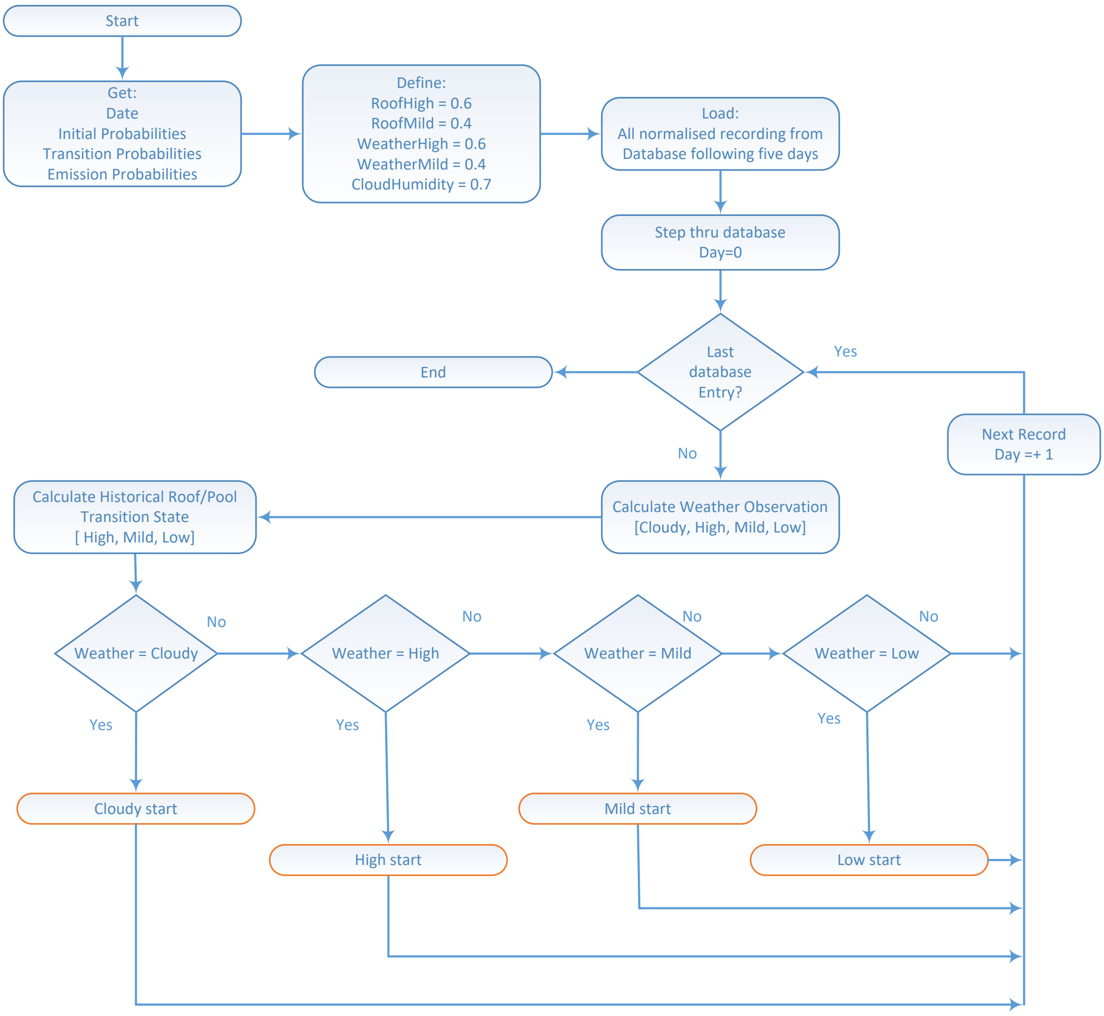


Figure 5.10 Flow chart one to calculating Viterbi five day predictions

The procedure requires the first day, HMM initial, transition and emission probabilities. All values higher than 0.6 will be classified as High for Roof, Pool and

Weather values. Values between 0.4 and 5.9 will be Mild and below 3.9 as Low. As discussed in Section 5.2 and Figure 5.3 cloud and humidity entries higher than 0.7 together will be classified as cloudy.

The weather conditions are classified as observation states [Cloudy, High, Mild, Cold] as illustrated in Figure 5.44 as Weather and if historical data are used the recorded temperatures are classified as transition states [Hot, Mild, Cold] as illustrated in Figure 5.44 as recorded.

Figure 5.11 illustrates a detailed flow chart after observation state Cloudy was calculated. The same flow chart can be used for another observation states as WeatherCloudy must be replace with WeatherHigh, WeaterMild and WeatherLow depending on the emission states.

Firstly, the product of the initial probability for the roof or pool and the emission of the roof or pool temperatures for each state [High, Mild, Low] are calculated. The mathematical formula for the complete process is described at equation 2.88 where $\delta_1(i) = \pi_i \, b_i O_1$ and there is no best route as this is day zero as described in equation 2.89 where $\gamma_1(i) = 0$.

For each day, from one to five the Viterbi states are calculating where each Viterbi state [High, Mild, Low] the product of each previous state and all three transition states and single emission state are calculated. The highest values for [High, Mild, Low] are recorded for each day. The mathematical formula for the complete process is described in equation 2.90 as $\delta_t(j) = \max [\delta_{t-1}(i) \, a_{ij}] \, . \, b_j O_t$. The results can be seen in Figure 5.44 for all five days under High, Mild and Low.

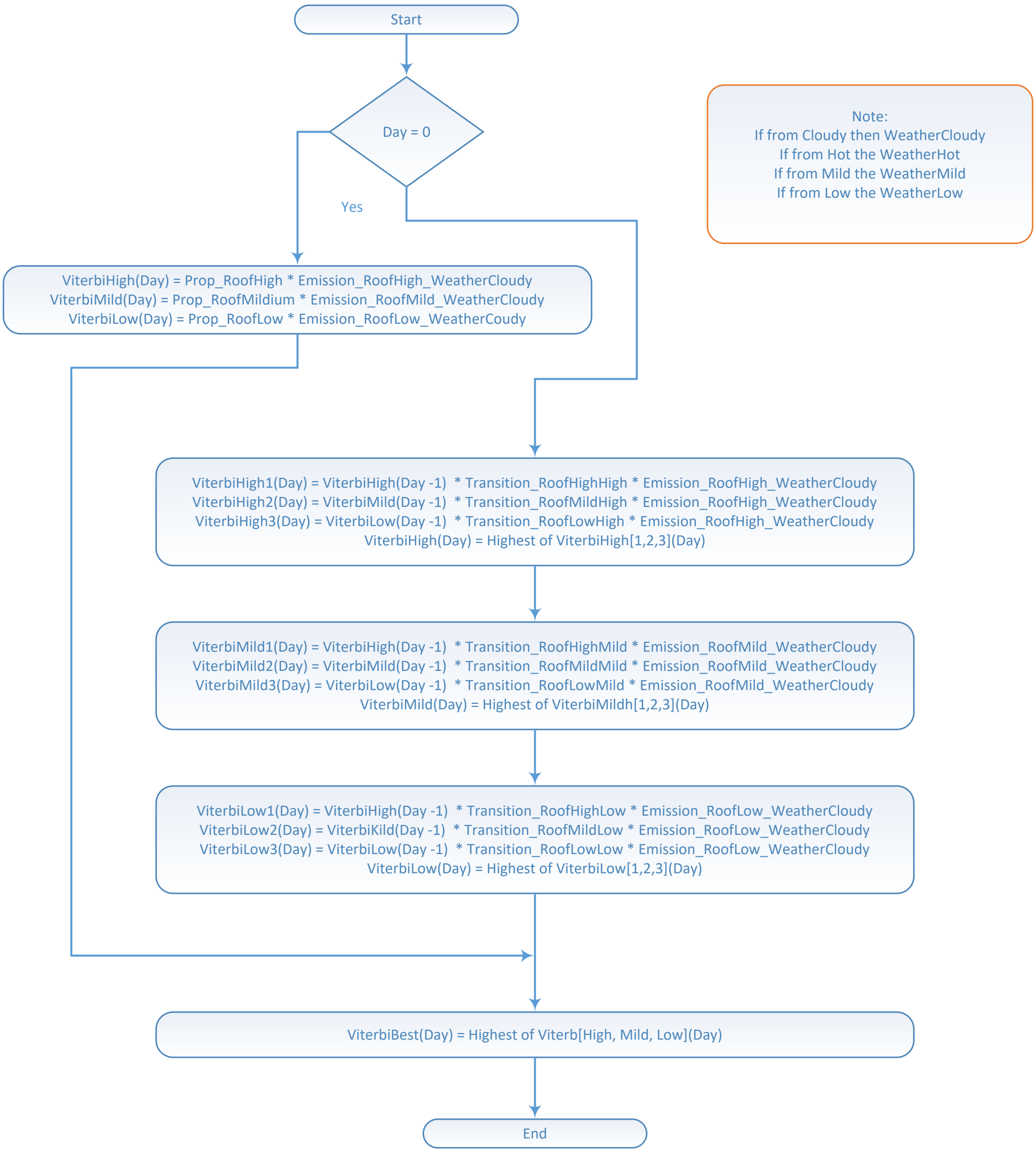


Figure 5.11 *Flow chart two calculating Viterbi five day predictions*

Lastly, the best path is calculated per day where the highest Viterbi state form [High, Mild, Low] is selected for next day's calculations and are marked as green under each day in Figure 5.44 [49, p.289]. The mathematical formula for the complete process is described in equation 2.91 where $\gamma_t(j)$ = arg max $[\delta_{t-1}(i)\ a_{ij}]$.

The Viterbi diagram for each month for each rainy season are shown in section 5.6 for the geyser and pool in section 5.5.

5.5 *Viterbi prediction values for Weather prediction with Pool*

The difference between the pool water and the solar panel temperature can be used to measure the solar heat efficiency as described in section 4.6.5 but is not an indication of swimming conditions as when there is a repeat in hot days, the water temperature reaches maximum heated temperature as indicated in Figure 5.12 where the pool and solar heated water is the same temperature.

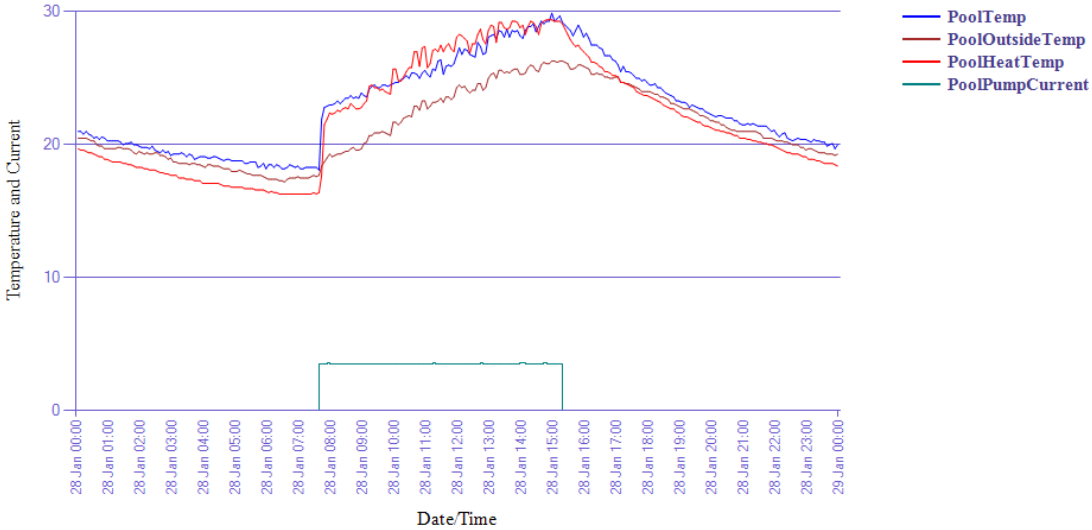


Figure 5.12 Pool related temperatures 28 Jan 2020

5 Day Prediction Viterbi Algorithm

Weather

Date	2019/6/30	2019/7/1	2019/7/2	2019/7/3	2019/7/4
Weather	M	L	L	M	M
Pool Heating					
High	0.476	0.289598	0.176191	0.0962	0.052525
Mild	0.0754	0.007306	0.000708	9.4E-05	1.2E-05
Low	0.0012	0.000113	1.1E-05	1E-06	0
Recorded	H	H	H	H	H

Figure 5.13 Viterbi five day prediction using delta pool and solar temperature: winter

Figure 5.13 indicates that the Viterbi prediction and recordings heating efficiency of the pool solar system as High where the pool water was only 14°C. The water cool downs at a higher rate as during the summer resulting in not heating up the pool.

5 Day Prediction Viterbi Algorithm

Weather

Date	2019/9/14	2019/9/15	2019/9/16	2019/9/17	2019/9/18
Weather	H	M	H	H	H

Roof Temperature

High	0.0392	0	0.003207	0.001495	0.000697
Mild	0.3402	0.163636	0.076287	0.035565	0.01658
Low	0.0288	0.021433	0.002651	0.001236	0.000576
Recorded	L	L	L	L	L

Figure 5.14 Viterbi five day prediction using delta pool and solar temperature: summer

The opposite is shown in Figure 5.14 where the pool water does not cool down as rapidly as during the winter, now indicate a low efficiency heating the pool. The normalised heated pool temperature will be used instead in the algorithm as above is incorrect.

Wind speed needs to be included with some human indication, as gardens with more leaves and dust needs to be cleaned more regularly than other pools. If dirt was not a factor, the prediction model would deactivate the pool pump if the wind speed was above a defined value, as the wind cools down the solar panels and the pool surface water due to evaporation and convection losses.

The normalised pool related temperatures are 30.8°C for the pool on 26 March 2019, Pool solar temperature was 43.4°C on 30 March 2019 and the outside temperature was 31°C on 31 March 2019. The normalised delta pool temperature is 4°C. It is important to note the pool temperature was only recorded from 2019 and the weather temperatures from 2017. The system will automatically update the maximum values if higher temperature values are recorded.

5.5.1 Rainy season: High

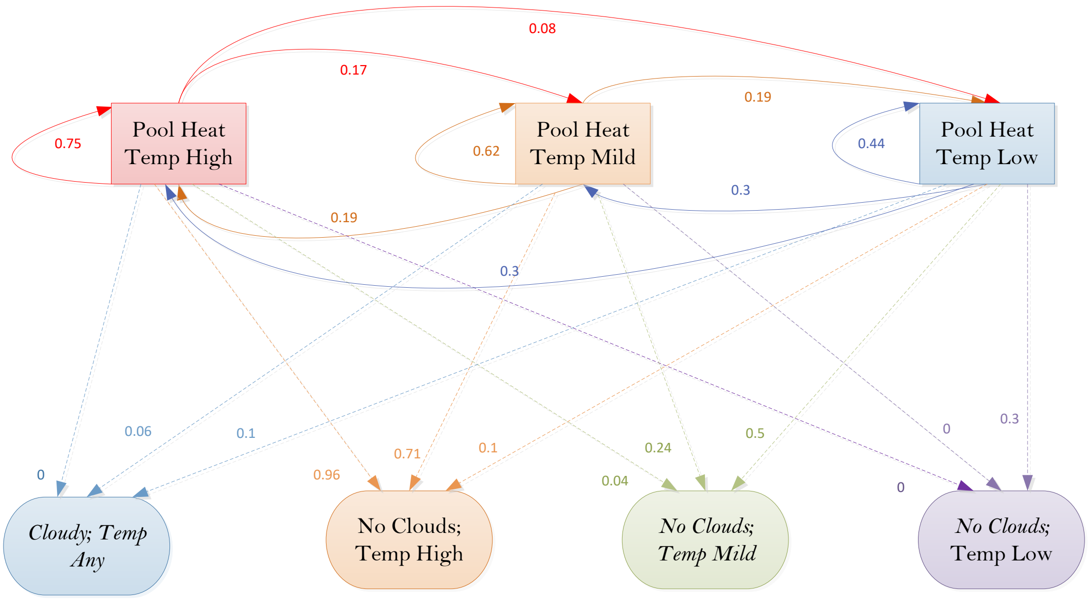


Figure 5.15 Rainy season High: Normalised pool and weather temperatures

Figure 5.15 illustrates the HMM transition and emission probability state diagram for the weather conditions and pool temperatures from 15 March to 30 April 2019.

Table 5.1 illustrates the initial state probability distribution matrix, Table 5.2 the HMM Transition probability distribution matrix and Table 5.3 the HMM Emission probability distribution matrix for the pool temperatures. The process calculating the values were discussed in section 5.3.

Table 5.1 Initial state probability distribution for Pool. Rainy Season: High

Current	Pool High	Pool Mild	Pool Low
Start	0.47	0.33	0.2

Table 5.2 HMM Transition probability distribution for Pool. Rainy Season: High

	Next		
Current	Pool High	Pool Mild	Pool Low
Pool High	0.75	0.17	0.08
Pool Mild	0.19	0.62	0.19
Pool Low	0.3	0.3	0.4

Table 5.3 HMM Emission probability distribution for Pool. Rainy Season: High

	Observation / Emission			
State	Temp: Any Cloudy	Temp: High No Clouds	Temp: Mild No Clouds	Temp: Low No Clouds
Pool High	0	0.96	0.04	0
Pool Mild	0.06	0.71	0.24	0
Pool Low	0.1	0.1	0.5	0.3

Figure 5.16 to Figure 5.17 illustrates the Viterbi results, including comparison to actual recorded data if the HMM states are used for historical data for 2019, for the selected five-days during March and April. The method calculating the results is the same as during the high rainy season described in section 5.6.1 example 5.1, except that this season's HMM transitions and emission values will be used.

5 Day Prediction Viterbi Algorithm

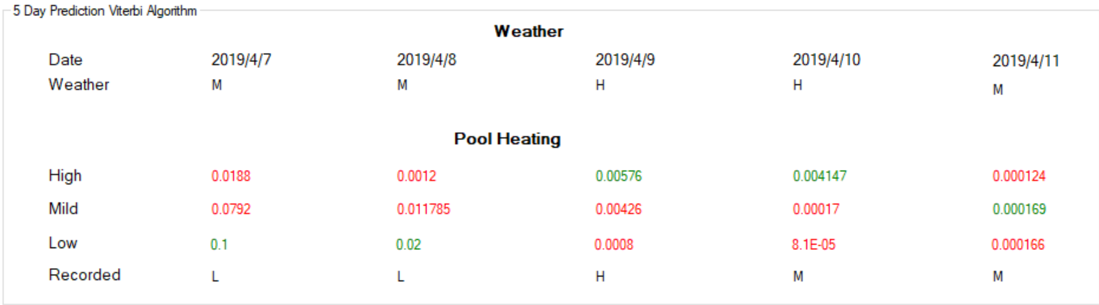
Weather

Date	2019/3/20	2019/3/21	2019/3/22	2019/3/23	2019/3/24
Weather	H	H	H	H	H

Pool Heating

High	0.4512	0.324864	0.233902	0.168409	0.121254
Mild	0.2343	0.00426	0.000948	0.000554	0.000399
Low	0.02	0.004452	0.002599	0.001871	0.001347
Recorded	H	H	H	H	H

Figure 5.16 Five day pool temperature Viterbi prediction 20 to 24 March 2019



5 Day Prediction Viterbi Algorithm

Weather

Date	2019/4/7	2019/4/8	2019/4/9	2019/4/10	2019/4/11
Weather	M	M	H	H	M

Pool Heating

High	0.0188	0.0012	0.00576	0.004147	0.000124
Mild	0.0792	0.011785	0.00426	0.00017	0.000169
Low	0.1	0.02	0.0008	8.1E-05	0.000166
Recorded	L	L	H	M	M

Figure 5.17 Five day pool temperature Viterbi prediction 7 to 11 April 2019

The HMM values accuracy for March and April are shown in Table 5.4. Calculating the accuracy, two points were given for the correct pool heated temperature collection and one point if the neighbouring temperature collection were calculated.

Table 5.4 Rainy Season: High - Pool temperature probability accuracy

	Month	
Accuracy (%)	Mar	Apr
Day [0]	100	100
Day [1]	100	100
Day [0 - 4]	100	90

For the weeks selected, the accuracy of Day[0] and Day[1], where 100% correct. The 5-day prediction where 100% correct for March and 90% for April.

5.5.2 **Rainy season: Low**

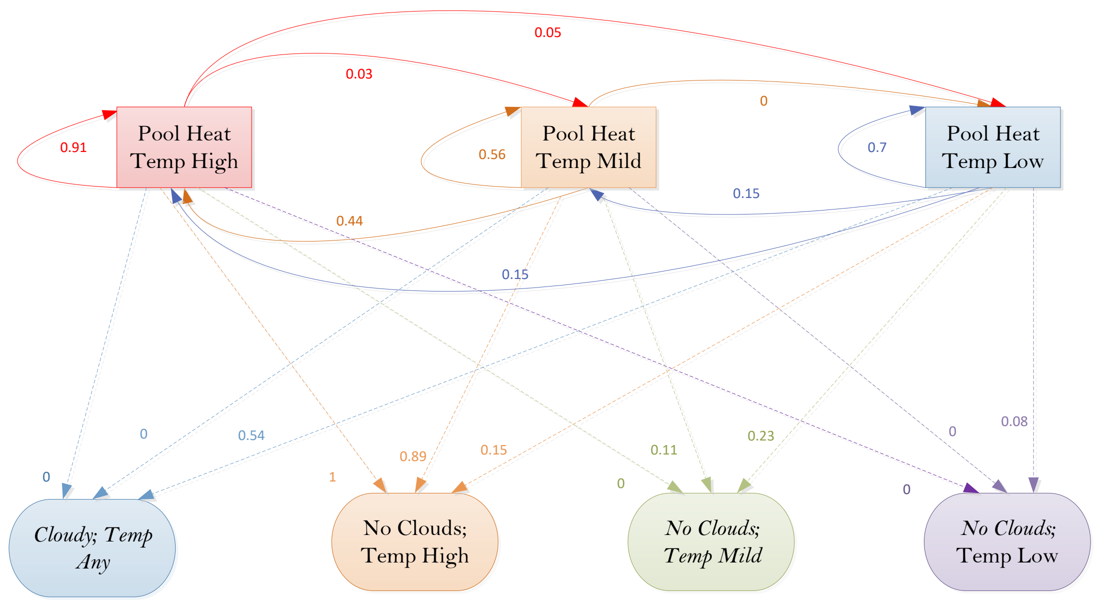


Figure 5.18 Rainy season Low: Normalised pool and weather temperatures

Figure 5.18 illustrates the HMM transition and emission probability state diagram for the weather conditions and pool temperatures from 1 October to 31 December 2019.

Table 5.5 illustrates the initial state probability distribution matrix, Table 5.6 the HMM Transition probability distribution matrix and Table 5.7 the HMM Emission probability distribution matrix for the pool temperatures. The process calculating the values were discussed in section 5.3.

Table 5.5 Initial state probability distribution for Pool. Rainy Season: Low

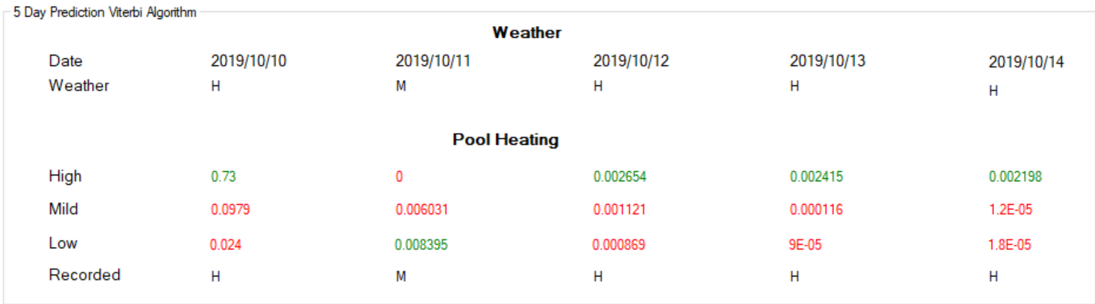
Current	Pool High	Pool Mild	Pool Low
Start	0.73	0.11	0.16

Table 5.6 *HMM Transition probability distribution for Pool. Rainy Season: Low*

	Next		
Current	Pool High	Pool Mild	Pool Low
Pool High	0.91	0.03	0.05
Pool Mild	0.44	0.56	0
Pool Low	0.15	0.15	0.7

Table 5.7 *HMM Emission probability distribution for Pool. Rainy Season: Low*

	Observation / Emission			
State	Temp: Any Cloudy	Temp: High No Clouds	Temp: Mild No Clouds	Temp: Low No Clouds
Pool High	0	1	0	0
Pool Mild	0	0.89	0.11	0
Pool Low	0.54	0.15	0.23	0.08



5 Day Prediction Viterbi Algorithm

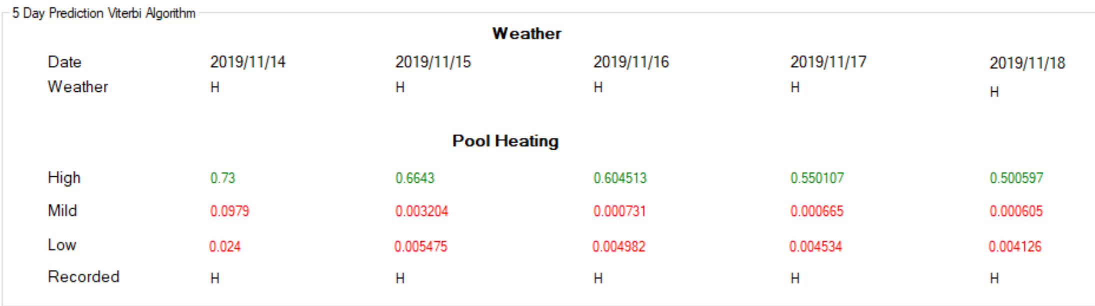
Weather

Date	2019/10/10	2019/10/11	2019/10/12	2019/10/13	2019/10/14
Weather	H	M	H	H	H

Pool Heating

High	0.73	0	0.002654	0.002415	0.002198
Mild	0.0979	0.006031	0.001121	0.000116	1.2E-05
Low	0.024	0.008395	0.000869	9E-05	1.8E-05
Recorded	H	M	H	H	H

Figure 5.19 *Five day pool temperature Viterbi prediction 10 to 14 October 2019*



5 Day Prediction Viterbi Algorithm

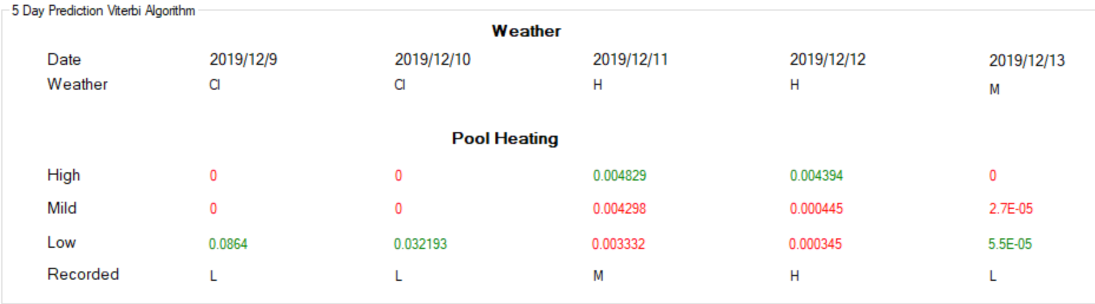
Weather

Date	2019/11/14	2019/11/15	2019/11/16	2019/11/17	2019/11/18
Weather	H	H	H	H	H

Pool Heating

High	0.73	0.6643	0.604513	0.550107	0.500597
Mild	0.0979	0.003204	0.000731	0.000665	0.000605
Low	0.024	0.005475	0.004982	0.004534	0.004126
Recorded	H	H	H	H	H

Figure 5.20 *Five day pool temperature Viterbi prediction 14 to 18 November 2019*



5 Day Prediction Viterbi Algorithm

Weather

Date	2019/12/9	2019/12/10	2019/12/11	2019/12/12	2019/12/13
Weather	Cl	Cl	H	H	M

Pool Heating

High	0	0	0.004829	0.004394	0
Mild	0	0	0.004298	0.000445	2.7E-05
Low	0.0864	0.032193	0.003332	0.000345	5.5E-05
Recorded	L	L	M	H	L

Figure 5.21 Five day pool temperature Viterbi prediction 9 to 13 December 2019

The HMM values accuracy for October to December are shown in Table 5.8. Calculating the accuracy, two points were given for the correct pool temperate collection and one point if the neighbouring temperature collection were calculated.

Table 5.8 Rainy Season: Low - Pool temperature probability accuracy

	Month		
Accuracy (%)	Oct	Nov	Dec
Day 0	100	100	100
Day1	50	100	100
Day 0 to 4	90	100	90

The accuracy was 100% correct for all Day[0] but for October Day[1] were calculated as Low but Mild where recorded. During the 5-day prediction, the probability varied between 90% and 100% for November.

5.5.3 **Rainy season: None**

Figure 5.22 illustrates the HMM transition and emission probability state diagram for the weather conditions and pool temperatures from 1 May to 30 September 2019.

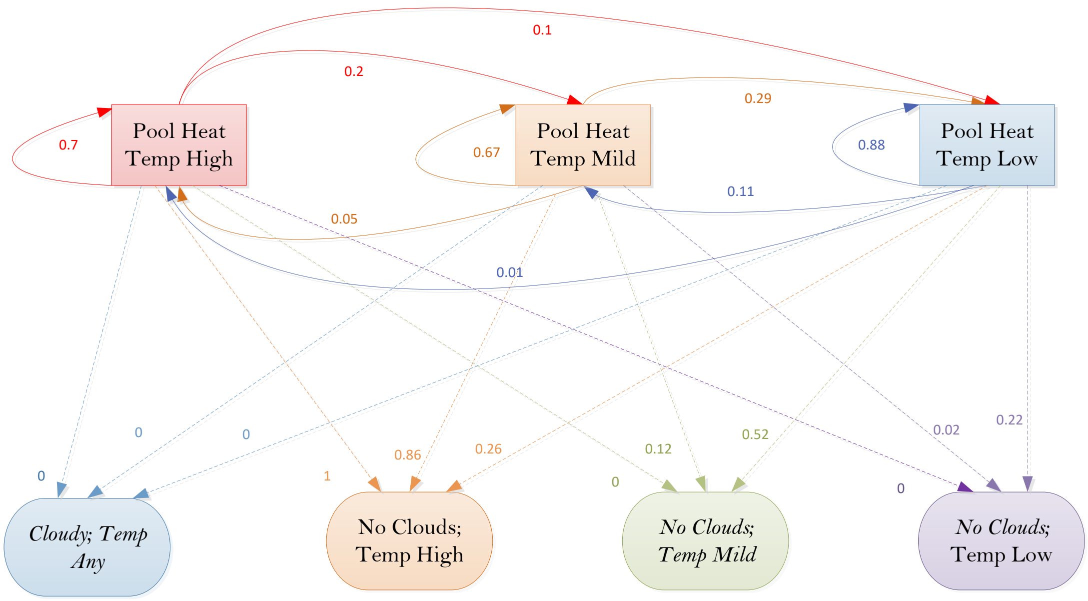


Figure 5.22 Rainy season None: Normalised pool and weather temperatures

Table 5.9 illustrates the initial state probability distribution matrix, Table 5.10 the HMM Transition probability distribution matrix and Table 5.11 the HMM Emission probability distribution matrix for the pool temperatures. The process calculating the values were discussed in section 5.3.

Table 5.9 Initial state probability distribution for Pool. Rainy Season: None

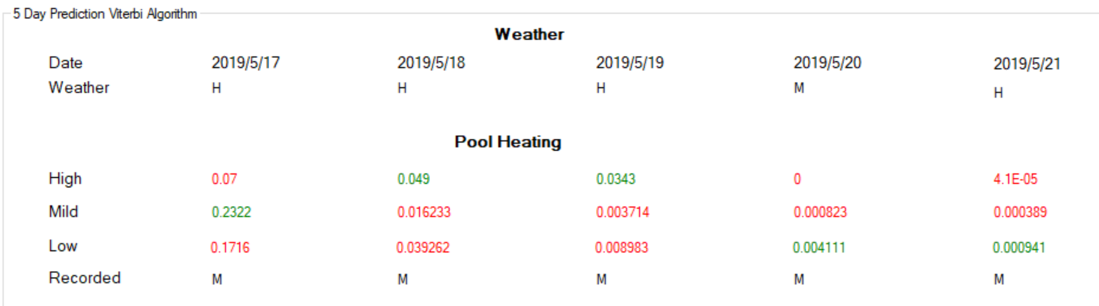
Current	Pool High	Pool Mild	Pool Low
Start	0.07	0.27	0.66

Table 5.10 HMM Transition probability distribution for Pool. Rainy Season: None

	Next		
Current	Pool High	Pool Mild	Pool Low
Pool High	0.7	0.05	0.01
Pool Mild	0.2	0.67	0.11
Pool Low	0.1	0.29	0.88

Table 5.11 HMM Emission probability distribution for Pool. Rainy Season: None

	Observation / Emission			
State	Temp: Any Cloudy	Temp: High No Clouds	Temp: Mild No Clouds	Temp: Low No Clouds
Pool High	0	1	0	0
Pool Mild	0	0.86	0.12	0.02
Pool Low	0	0.26	0.52	0.22


5 Day Prediction Viterbi Algorithm

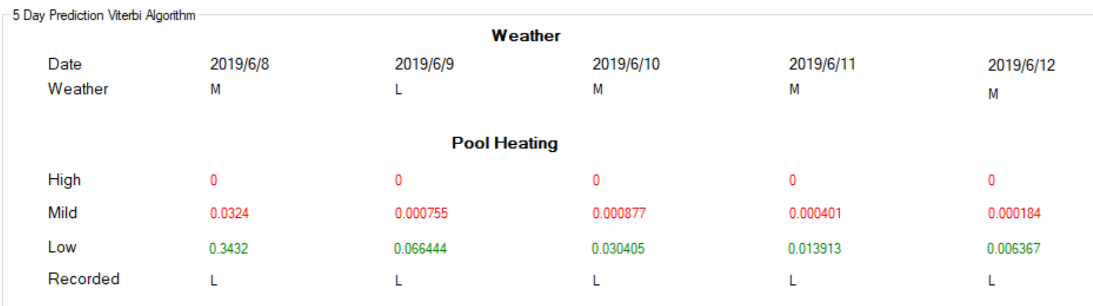
Weather

Date	2019/5/17	2019/5/18	2019/5/19	2019/5/20	2019/5/21
Weather	H	H	H	M	H

Pool Heating

High	0.07	0.049	0.0343	0	4.1E-05
Mild	0.2322	0.016233	0.003714	0.000823	0.000389
Low	0.1716	0.039262	0.008983	0.004111	0.000941
Recorded	M	M	M	M	M

Figure 5.23 Five-day pool temperature Viterbi prediction 17 to 21 May 2019


5 Day Prediction Viterbi Algorithm

Weather

Date	2019/6/8	2019/6/9	2019/6/10	2019/6/11	2019/6/12
Weather	M	L	M	M	M

Pool Heating

High	0	0	0	0	0
Mild	0.0324	0.000755	0.000877	0.000401	0.000184
Low	0.3432	0.066444	0.030405	0.013913	0.006367
Recorded	L	L	L	L	L

Figure 5.24 Five day pool temperature Viterbi prediction 8 to 12 June 2019

5 Day Prediction Viterbi Algorithm

Weather

Date	2019/7/20	2019/7/21	2019/7/22	2019/7/23	2019/7/24
Weather	L	L	L	L	L

Pool Heating

High	0	0	0	0	0
Mild	0.0054	0.000319	6.2E-05	1.2E-05	2E-06
Low	0.1452	0.028111	0.005442	0.001054	0.000204
Recorded	L	L	L	L	L

Figure 5.25 Five day pool temperature Viterbi prediction 20 to 24 July 2019

5 Day Prediction Viterbi Algorithm

Weather

Date	2019/8/14	2019/8/15	2019/8/16	2019/8/17	2019/8/18
Weather	H	M	H	H	M

Pool Heating

High	0.07	0	0.000933	0.000653	0
Mild	0.2322	0.018669	0.007428	0.0017	0.000137
Low	0.1716	0.078524	0.017966	0.004111	0.001881
Recorded	M	L	L	M	L

Figure 5.26 Five day pool temperature Viterbi prediction 15 to 19 August 2019

5 Day Prediction Viterbi Algorithm

Weather

Date	2019/9/8	2019/9/9	2019/9/10	2019/9/11	2019/9/12
Weather	M	H	H	H	H

Pool Heating

High	0	0.003432	0.002402	0.001681	0.001177
Mild	0.0324	0.032467	0.007428	0.0017	0.000389
Low	0.3432	0.078524	0.017966	0.004111	0.000941
Recorded	L	L	M	H	L

Figure 5.27 Five day pool temperature Viterbi prediction 8 to 12 September 2019

The HMM values accuracy from May to April are shown in Table 5.12. Calculating the accuracy, two points were given for the correct roof temperate collection and one point if the neighbouring temperature collection were calculated.

Table 5.12 Rainy Season: Low - Roof temperature probability accuracy

	Month				
Accuracy (%)	May	Jun	Jul	Aug	Sep
Day [0]	100	100	100	100	100
Day [1]	50	100	100	100	100
Day [0 - 4]	60	100	100	90	50

The accuracy was 100% correct for all Day[0] predictions. Only during May, Day[1] was incorrectly calculated. The 5-day probability varied between 50% and 100%, with the first month, May and last month, September the lowest.

5.5.4 Baum-Welch algorithm training one HMM model

The HMM proof to be relatively accurate for the first two day predictions, but as the climate change each month the HMM model needs to be updated as seen from section 5.5.1 to 5.5.3. Baum-Welch algorithm as described in section 2.4.3.3, will be used to train the model. Simulations will be done, training only one HMM season model as multiple seasonal models are impractical because;

- Rainy seasons may vary each year, influence of El Nino and La Nina events, as one year can be a wet season and another a dry season;
- The model will be inaccurate when changing from one model to another, as the first model is already trained for current conditions, but the following model will still be trained in the previous climate conditions.
- Training multiple models more complex than a single model.

Training the HMM model is by far the most complex of the three HMM problems as explained in section 2.4.3 [18 p. 264].

Before discussing the solution, it is important to discuss the first problem that was encountered when one year's data were simulated. The module starts getting stuck into a specific prediction. Investigating the HMM trained transition and emission values it was found that certain values become zero. Investing the formulas 2.107 and 2.108 used to train the module, it was found that the values used to multiplying or dividing result in an outcome of zero or undefined. The outcome was that certain predictions would be stuck as a zero or one, as shown in equation 5.11.

Resolving the stuck in local maxima appearance by first ensure that no HMM model values were zero or one [37]. Another problem was that after the Baum-Welch algorithm altered the model values the sum of the normalised values was not always "1" called label bias problem. First, the model was re-normalised after the Baum-Welch calculation and then by subtract 0.01 from the highest prediction and add to the zero valued probability ensuring the lowest value to be 0.01 and the highest 0.99. Modifications are done for initial, transition and emission probabilities and shown in Figure 5.31 as modification 1-1 to modification1-3.

Appendix L explains a snippet of the program executes the Baum-Welch procedure. Figure 5.28 illustrates the main flow chart training the HMM model. By inspection, the most accurate predictions with lowest CPU requirements and fastest processing were calculated from eight observations, seven previous and one current day, as illustrated in Figure 5.32. If the observation sequence is too long, the model will calculate seasonal outputs and is unresponsive to sudden weather changes like cold front or a rainy week after a few weeks of hot and dry days. It is important to remember that predictions made in this section were made before sun rise based on official weather predictions by the weather bureau. Current local weather temperature recording is excluded as it is out of the scope of this research and it is acknowledged that it can increase the accuracy of the prediction model.

The training model requires an observation sequence, transition and emission probabilities, recorded weather and pool temperatures. First, the forward and backwards likelihoods are calculated. These values together with the transition and emission probabilities are used by the Baum-Welch algorithm training the model. The illustrated modification 3 will be explained a later stage. Lastly, the pool temperature is predicted by the Viterbi algorithm using the trained HMM model. The Viterbi algorithm is explained in detail in section 5.4.

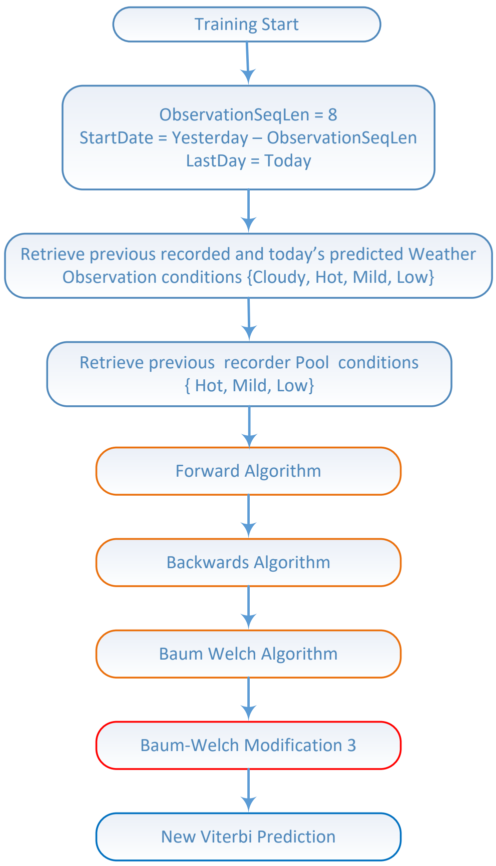


Figure 5.28 Main flow chart training the HMM model

Figure 5.29 illustrates the forward algorithm and the theory was discussed in section 2.4.3.1. The algorithm will be explained by calculations for 22 March 2019.

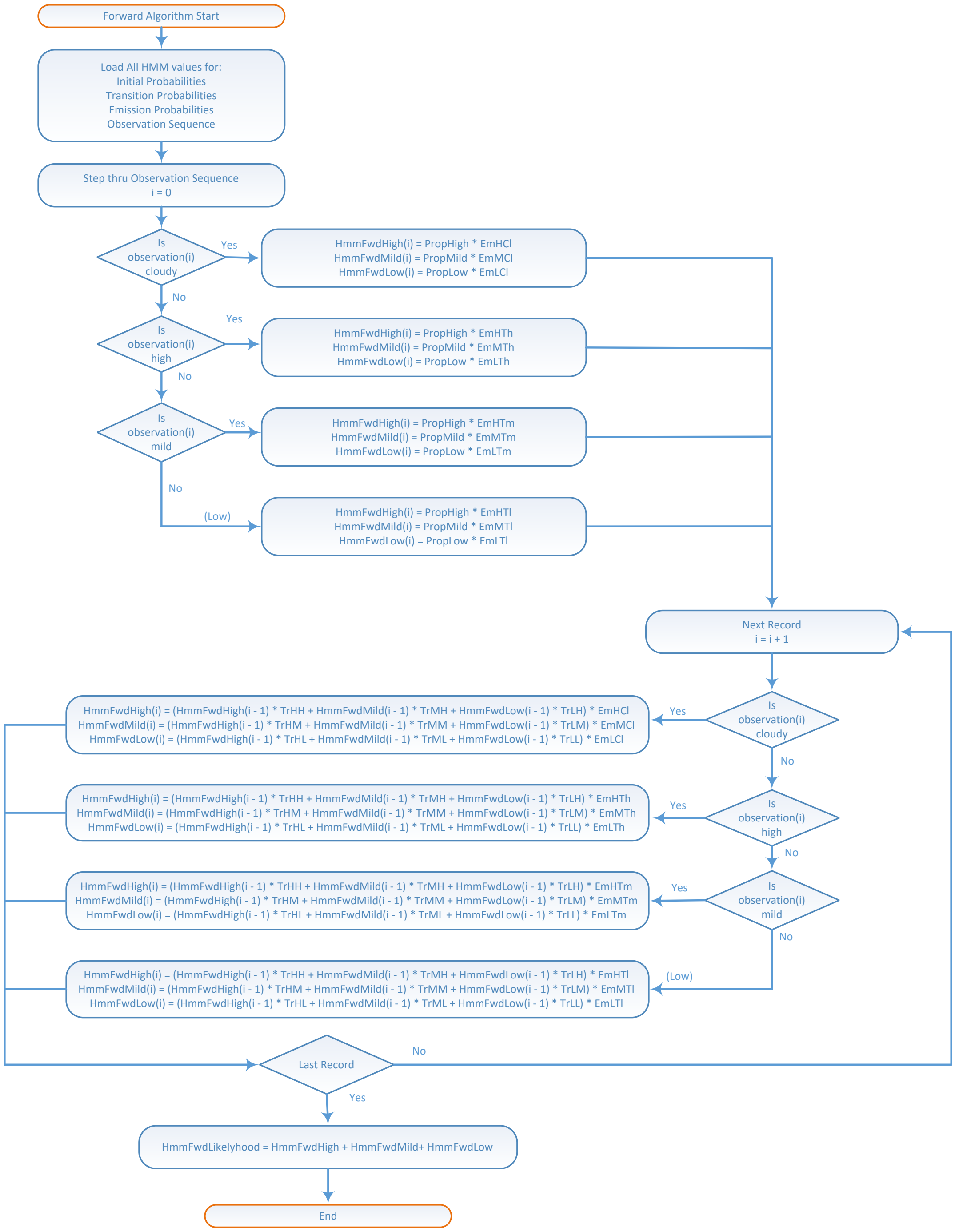


Figure 5.29 Forward Algorithm

The forward algorithm requires the observation sequence, initial, transition and emission probabilities as shown in equation 5.1 to 5.4 and will be used in following calculations.

$$O = H,H,H,H,H,H,H,H \tag{5.1}$$

$$\pi = \begin{bmatrix} 0.48 & 0.34 & 0.18 \end{bmatrix} \tag{5.2}$$

$$A = \begin{bmatrix} 0.75 & 0.17 & 0.08 \\ 0.19 & 0.62 & 0.19 \\ 0.22 & 0.33 & 0.45 \end{bmatrix} \tag{5.3}$$

$$B = \begin{bmatrix} 0 & 0.96 & 0.04 & 0 \\ 0.06 & 0.71 & 0.23 & 0 \\ 0.11 & 0.11 & 0.56 & 0.22 \end{bmatrix} \tag{5.4}$$

The initial forward parameters calculated is based on the first weather observation and the mathematical formula for the complete process is described at equation 2.83 where $\alpha_t(i) = \pi_i \, b_i O_1$ where $1 \leq i \leq N$.

The forward values at T=1 are calculated as follows:

$O_t = H$ where $t = 1$

$\alpha_1(h) = \pi_h \times b_{hh} = 0.48 \times 0.96 = 0.4608$

$\alpha_1(m) = \pi_m \times b_{hm} = 0.34 \times 0.71 = 0.2414$

$\alpha_1(l) = \pi_l \times b_{hl} = 0.18 \times 0.11 = 0.0198$

The next stage is calculating the induction for the following observation sequences as described at equation 2.84 where

$$\alpha_{t+1}(j) = \left[\sum_{i=1} \alpha_t(i) \, a_{ij} \right] b_j O_{t+1} \text{ where } 1 \leq j \leq N \text{ and } 1 \leq t \leq T\text{-}1$$

The forward values at T=2 to T=8 are calculated as follows:

O_t = H where t=2

$$\alpha_2(h) = [(\alpha_1(h) \times a_{hh}) + (\alpha_1(m) \times a_{mh}) + (\alpha_1(l) \times a_{mh})] \times b_{hh}$$
$$= [(0.4608 \times 0.75) + (0.2414 \times 0.19) + (0.0198 \times 0.22)] \times 0.96 = 0.37998912$$

$$\alpha_2(m) = [(\alpha_1(h) \times a_{hm}) + (\alpha_1(m) \times a_{mm}) + (\alpha_1(l) \times a_{ml})] \times b_{hm}$$
$$= [(0.4608 \times 0.17) + (0.2414 \times 0.62) + (0.0198 \times 0.33)] \times 0.71 = 0.16652198$$

$$\alpha_2(l) = [(\alpha_1(h) \times a_{hl}) + (\alpha_1(m) \times a_{ml}) + (\alpha_1(l) \times a_{ml})] \times b_{hl}$$
$$= [(0.4608 \times 0.08) + (0.2414 \times 0.19) + (0.0198 \times 0.45)] \times 0.11 = 0.01005862$$

The same processes are followed for O_3 to O_8

$$\alpha = \begin{bmatrix} 0.4608 & 0.3799 & 0.3061 & 0.2441 & 0.1937 & 0.1534 & 0.1213 & 0.0959 \\ 0.2414 & 0.1665 & 0.1215 & 0.0921 & 0.07133 & 0.0558 & 0.0438 & 0.0345 \\ 0.0198 & 0.0100 & 0.0073 & 0.0055 & 0.0043 & 0.0034 & 0.0026 & 0.0021 \end{bmatrix} \tag{5.5}$$

The final likelihood is calculated as described as equation 2.85 where

$$P(O|\lambda) = \sum_{i=1} \alpha_T(i)\,.$$

The likelihood for the observation sequence is 0.132657661 (5.6)

Figure 5.30 illustrates the backward algorithm and the theory was discussed in section 2.4.3.3. The algorithm will be explained by calculations for 22 March 2019.

The backward algorithm requires the initial, transition and emission probabilities and the observation sequence. The initial backward parameters calculated is based on the first weather observation and the mathematical formula for the complete process is described in equation 2.99 where $\beta_T(i) = 1$ where $1 \leq i \leq N$.

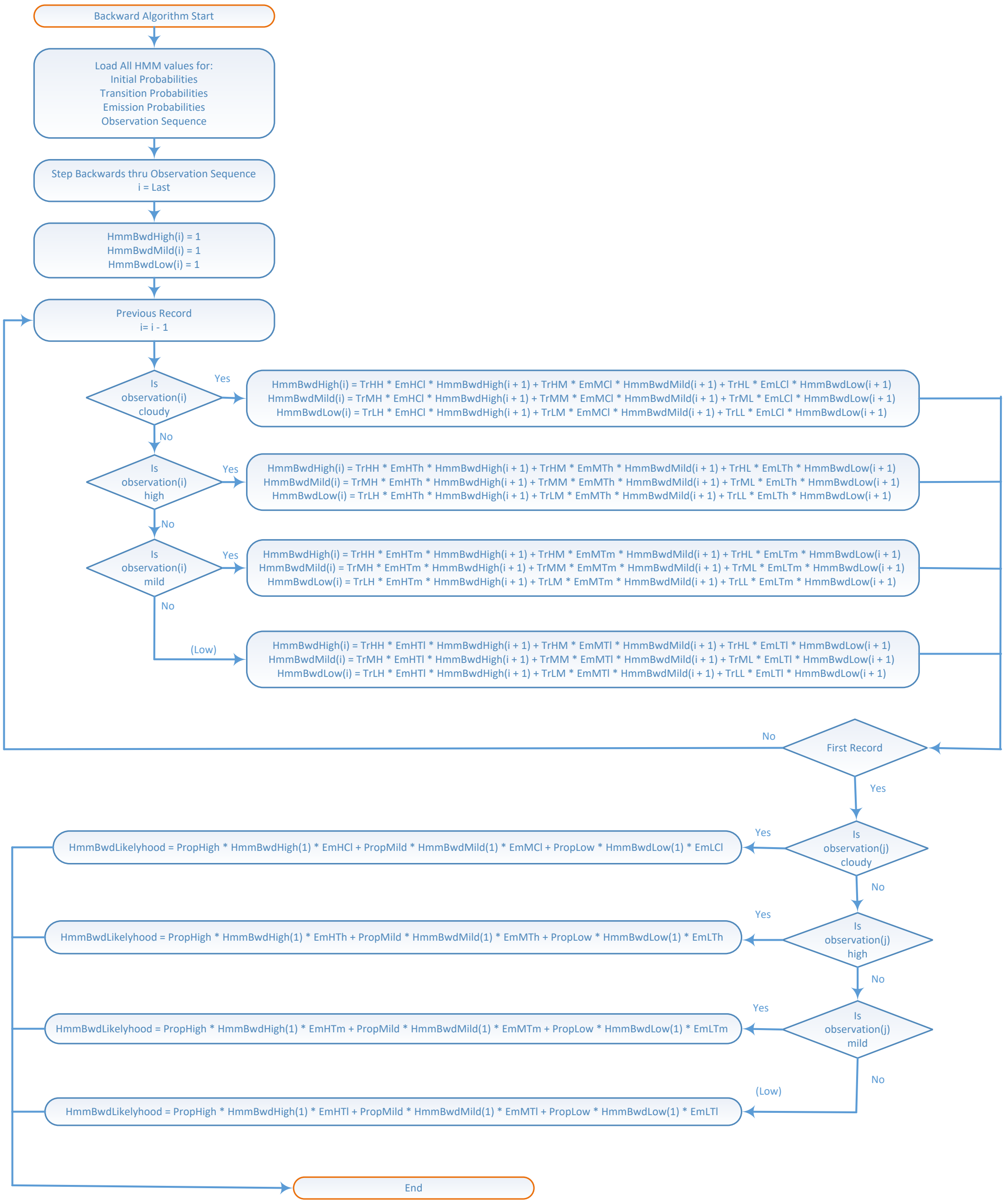


Figure 5.30 Backward Algorithm

The forward values at T=9 are calculated as follows:

O_t = H where t = 8

$\beta_8(h) = 1$

$\beta_8(m) = 1$

$\beta_8(l) = 1$

The next step is calculating the induction for the following observation sequences as described in equation 2.100 where

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j)\, a_{ij}\, b_j\, O_{t+1} \qquad \text{where} \quad 1 \leq j \leq N \text{ and } 1 \leq t \leq T\text{-}1$$

The backwards values at T=7 to T=2 are calculated as follows:

O_t = H where t=7

$\beta_7(h) = [(a_{hh} \times b_{hh} \times \beta_8(h)) + (a_{hm} \times b_{hm} \times \beta_8(m) + (a_{hl} \times b_{hl} \times \beta_8(l)]$
$= [0.75 \times 0.96 \times 1] + [0.17 \times 0.71 \times 1] + [0.08 \times 0.11 \times 1] = 0.8495$

$\beta_7(m) = [(a_{mh} \times b_{hh} \times \beta_8(h)) + (a_{mm} \times b_{hm} \times \beta_8(m) + (a_{ml} \times b_{hl} \times \beta_8(l)]$
$= [0.19 \times 0.96 \times 1] + [0.62 \times 0.71 \times 1] + [0.19 \times 0.11 \times 1] = 0.6435$

$\beta_7(l) = [(a_{lh} \times b_{hh} \times \beta_8(h)) + (a_{lm} \times b_{hm} \times \beta_8(m) + (a_{ll} \times b_{hl} \times \beta_8(l)]$
$= [0.22 \times 0.96 \times 1] + [0.33 \times 0.71 \times 1] + [0.45 \times 0.11 \times 1] = 0.4939$

O_t = H where t=6

$\beta_6(h) = [(a_{hh} \times b_{hh} \times \beta_7(h)) + (a_{hm} \times b_{hm} \times \beta_7(m) + (a_{hl} \times b_{hl} \times \beta_7(l)]$
$= [0.75 \times 0.96 \times 0.8495] + [0.17 \times 0.71 \times 0.6435] + [0.08 \times 0.11 \times 0.4939]$
$= 0.693656$

$\beta_6(m) = [(a_{mh} \times b_{hh} \times \beta_7(h)) + (a_{mm} \times b_{hm} \times \beta_7(m) + (a_{ml} \times b_{hl} \times \beta_7(l)]$
$= [0.19 \times 0.96 \times 0.8495] + [0.62 \times 0.71 \times 0.6435] + [0.19 \times 0.11 \times 0.4939]$
$= 0.44854$

$\beta_6(l) = [(a_{lh} \times b_{hh} \times \beta_7(h)) + (a_{lm} \times b_{hm} \times \beta_7(m) + (a_{ll} \times b_{hl} \times \beta_7(l)]$
$= [0.22 \times 0.96 \times 0.8495] + [0.33 \times 0.71 \times 0.6435] + [0.45 \times 0.11 \times 0.4939]$
$= 0.354091$

The same process is followed for O_5 to O_1

$$\beta = \begin{bmatrix} 0.2200 & 0.2782 & 0.3514 & 0.4431 & 0.5566 & 0.6936 & 0.8495 & 1 \\ 0.1212 & 0.1540 & 0.1965 & 0.2530 & 0.3313 & 0.4485 & 0.6435 & 1 \\ 0.1011 & 0.1281 & 0.1629 & 0.2082 & 0.2687 & 0.3541 & 0.4939 & 1 \end{bmatrix} \tag{5.7}$$

The final likelihood is calculated as follows and must be the same as the forward likelihood calculated in equation 5.6.

$$P(O|\lambda) = [(\pi_h \times \beta_1(h) \times b_{hh}) + (\pi_h \times \beta_1(h) \times b_{hh}) + (\pi_h \times \beta_1(h) \times b_{hh})]$$
$$= [(0.48 \times 0.2200 \times 0.96) + (0.34 \times 0.1212 \times 0.71) + (0.18 \times 0.101 \times 0.96)]$$
$$= 0.1326576612 \qquad (5.8)$$

The Baum-Welch flow chart illustrating the algorithm calculations is shown in Figure 5.31. The flowchart will be explained by calculating the re-estimated HMM probabilities for 22 March 2020.

First, the re-estimated value for the initial probabilities are calculated as discussed in section 2.4.3.3 at equation 2.105 where $\pi'_i = \gamma_1(i)$. Calculations were based on values from equations 5.1 to 5.8.

$\pi'_h = \alpha_1(h) \times \beta_1(h) / \beta_{likelihood} = 0.4608 \times 22003 = 0.76$

$\pi'_m = \alpha_1(m) \times \beta_1(m) / \beta_{likelihood} = 0.2414 \times 12123 = 0.22$

$\pi'_l = \alpha_1(l) \times \beta_1(l) / \beta_{likelihood} = 0.0198 \times 10105 = 0.02$

Values does not need to be re-normalised as sum is 1 and no values are zero

$$\pi' = [0.76 \quad 0.22 \quad 0.02] \qquad (5.9)$$

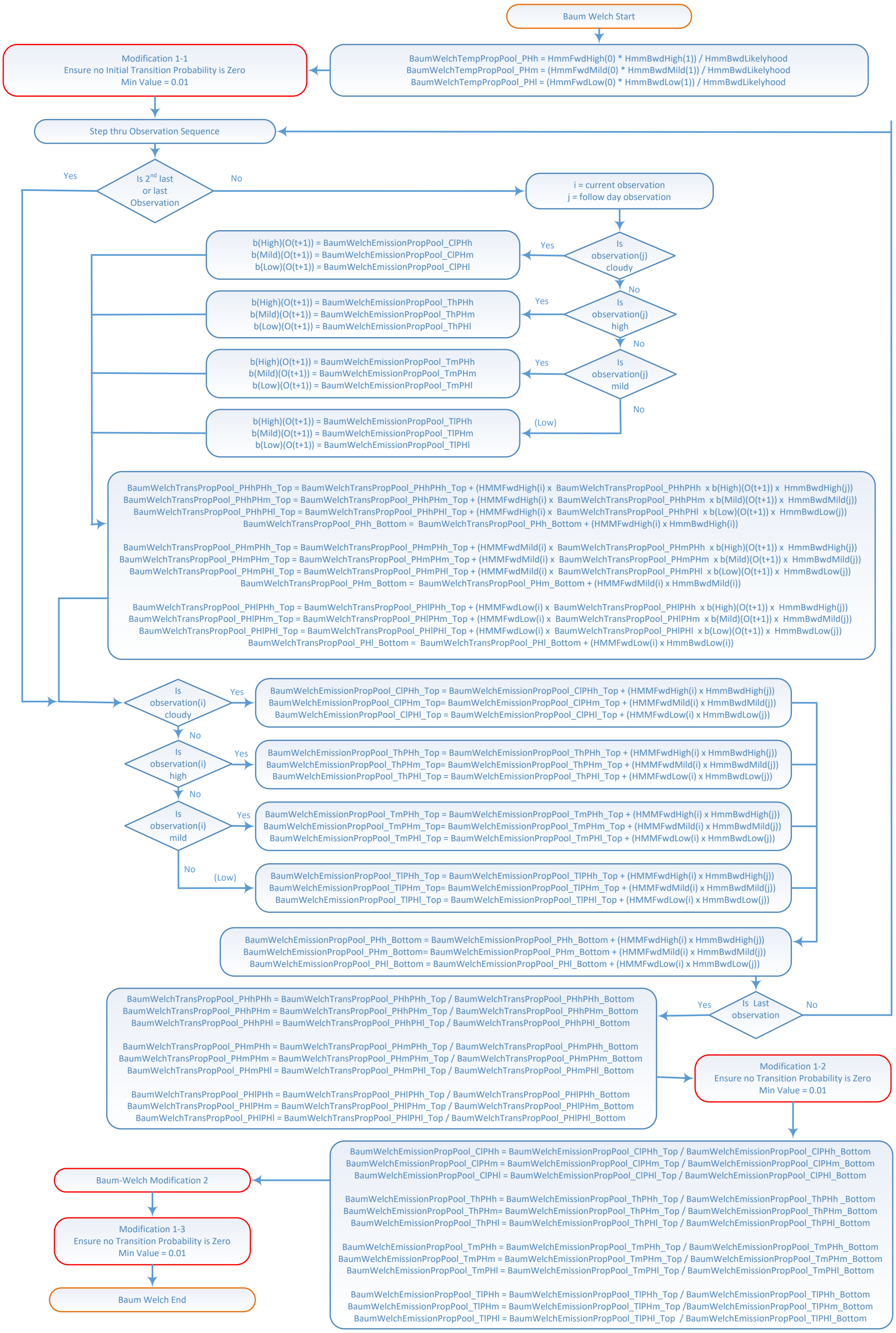


Figure 5.31 Baum-Welch algorithm

At the following stage, the re-estimated transition probabilities are calculated as illustrated at equation 2.106 where:

$$a'_{ij} = \frac{\sum_{t=1}^{T-1} \zeta(i,j) \quad \{\text{Expected number of transitions from } S_i \text{ to } S_j\}}{\sum_{t=1}^{T-1} \gamma_t(i) \quad \{\text{ Expected number of transitions from } S_i \text{ }\}}$$

The re-estimated transition pool high to high, mild and low at observation T=1 are calculated as follow:

O_t = H and O_{t+1} = H where t=1

$a'_{hh[top]} = a'_{hh[top]} + [\alpha_1(h) \times a_{hh} \times b_{hh} \times \beta_2(h)]$
$= 0 + 0.4608 \times 0.75 \times 0.96 \times 0.27821 = 0.0923036$

$a'_{hm[top]} = a'_{hm[top]} + [\alpha_1(h) \times a_{hm} \times b_{hm} \times \beta_2(m)]$
$= 0 + 0.4608 \times 0.17 \times 0.71 \times 0.15404 = 0.0085675$

$a'_{hl[top]} = a'_{hl[top]} + [\alpha_1(h) \times a_{hl} \times b_{hl} \times \beta_2(l)]$
$= 0 + 0.4608 \times 0.08 \times 0.11 \times 0.12817 = 0.00051974$

$a'_{h[bottom]} = a'_{h[Bottom]} + [\alpha_1(h) \times \beta_1(h)]$
$= 0 + 0.4608 \times 22003 = 0.10139099$

The same calculations need to be down for T=2 to T=7. The values now are:

$a'_{h[bottom]} = 0.7402315$

$a'_{hh[top]} = 0.66194302$

$a'_{hh} = a'_{hh[top]} / a'_{h[bottom]} = 0.66194302 / 0.7402315 = 0.89$

$a'_{hm} = a'_{hm[top]} / a'_{h[bottom]} = 0.073747309 / 0.7402315 = 0.1$

$a'_{hl} = a'_{hl[top]} / a'_{h[bottom]} = 0.004541241 / 0.7402315 = 0.01$

Values do not need to be re-normalised as sum is one and no values are zero.

$$a' = \begin{bmatrix} 0.89 & 0.1 & 0.01 \\ 0.38 & 0.6 & 0.2 \\ 0.54 & 0.39 & 0.07 \end{bmatrix} \quad (5.10)$$

The final stage is calculating the re-estimated emission probabilities as discussed is section 2.4.3.3 at equation 2.107 where:

$$b'_j(k) = \frac{\sum_{\substack{t=1 \\ \text{s.t } O_t = V_k}}^{T} \gamma_t(j) \quad \{\text{Expected number of times in state } j \text{ observing } V_k\}}{\sum_{t=1}^{T-1} \gamma_t(j) \quad \{\text{Expected number of transitions from } S_j\}}$$

The re-estimated emission probabilities will be calculated for weather high with Pool high, mild and low at observation T=1 as follows:

O_t = H where t=1

$b'_{hh[top]} = b'_{hh[top]} + [\alpha_1(h) \times \beta_1(h)] = 0.4608 \times 0.22003 = 0.10139$

$b'_{hm[top]} = b'_{hm[top]} + [\alpha_1(m) \times \beta_1(m)] = 0.2414 \times 0.22003 = 0.12123$

$b'_{hl[top]} = b'_{hl[top]} + [\alpha_1(l) \times \beta_1(l)] = 0.0198 \times 0.22003 = 0.10105$

The coefficient is calculated for all weather conditions.

$b'_{h[bottom]} = b'_{h[bottom]} + [\alpha_1(h) \times \beta_1(h)] = 0.4608 \times 0.22003 = 0.10139$

$b'_{m[bottom} = b'_{m[bottom]} + [\alpha_1(m) \times \beta_1(m)] = 0.2414 \times 0.22003 = 0.12123$

$b'_{l[bottom]} = b'_{l[bottom]} + [\alpha_1(l) \times \beta_1(l)] = 0.0198 \times 0.22003 = 0.10105$

The same calculations need to be done for T=2 to T=8. The values now are:

$b'_{h[bottom]} = 0.8361814$

$b'_{hh[top]} = 0.8361814$

$b'_{hh} = b'_{hh[top]} / b'_{h[bottom]} = 0.8361814 / 0.8361814 = 1$

$b'_{hm} = b'_{hm[top]} / b'_{m[bottom]} = 0.2136222 / 0.2136222 = 1$

$b'_{hl} = b'_{hl[top]} / b'_{l[bottom]} = 0.0114575 / 0.0114575 = 1$

$b'_{hx} = 1$ as all eight observations were H for this example.

$$b' = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{5.11}$$

Values do not need to be re-normalised as sum is 1 but with probabilities of 1and 0, the values will not change any more, in the Forward and Backwards algorithms the same parameters will now the multiplied with a "1" or a "0" with the

same result. The problem can be resolved with larger Observation sequences, but it was noticed that the prediction model calculating the same result per season and was insensitive to sudden weather changes. Another reason large observation sequence is impracticable is that the solution is developed for IoT sensors with small cost efficient micro controllers.

It was decided to modify the Beam-Welch algorithm to use small observation sequences without getting stuck at a certain emission probability. As the current emission probability already consists of a historical knowledge, it was investigated how it can be incorporated into the model.

Examine the data two possible modifications was highlighted. The first scenario was to calculate the re-estimated emission as follows:

$$a' = a' \times 0.1 + A \times 0.9 \qquad (5.12)$$

$$b' = b' \times 0.1 + B \times 0.9 \qquad (5.13)$$

Using 10% of the re-estimated probabilities with 90% historical probabilities shown more accurate transition and emission probabilities and sensitive to sudden weather changes.

The second scenario was to incorporate the historical pool recordings equal to the length of the observation sequence to equation 12 where:

$$b' = [(b' \times 0.1 + B \times 0.9) \times 0.9) + (\sum_{j=1}^{N = \text{Observation sequence Length}} \text{PoolObservation}(j)) \times 0.1 \qquad (5.14)$$

Figure 5.32 illustrates the accuracy of the trained Pool prediction model with a variant observation sequence length, with modification one, equation 5.12 and 5.13, and modification two, equation 5.14. The results were taken from 15 March to 20 December 2019. Figure 5.32 indicates correct as that the prediction was 100% correct, border indicates that the prediction was mild, but the actual recording was high or low and vice versa and incorrect indicated a prediction of high and an actual recording of low and vice versa.

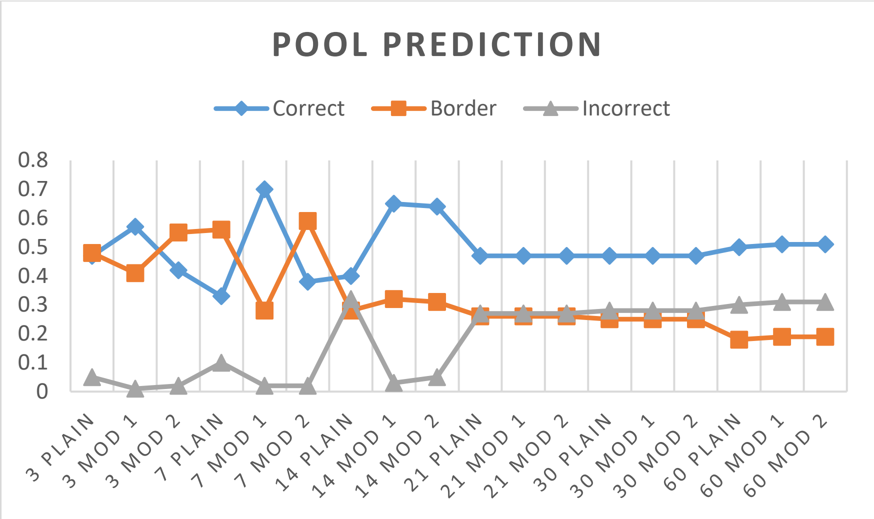


Figure 5.32 Observation length accurate comparison

Without any modification it was noted that:

- Larger observation sequences result in higher accuracy.
- The highest correct prediction is 50%.
- Increase correct prediction, also result in increased incorrect predictions with a decrease in border predictions.

With modification one it was noted that:

- Average correct prediction with small or large sequence is 57%.
- A spike in accuracy is between 7 and 14 days with 70% at 7 days.
- Increase correct prediction with 7 days, result in a decrease in border .prediction at 28% incorrect prediction with a decrease in border predictions and incorrect predictions of 2%.

With modification two it was noted that:

- Percentage correct is lower than modification 1 and higher than the original algorithm for 7 days and lower.
- No significant difference between modification one and two for observation sequences larger than 14 days.

As illustrated the best correct and lowest incorrect predictions are calculated with a seven day observation sequence with modification one.

The re-estimated emission probabilities from equation 5.11 must first be modified not containing any “1” or “0”. It is done by subtracting 0.01 from the highest value and added to the zero entries result in equation 5.15.

$$b' = \begin{bmatrix} 0.01 & 0.97 & 0.01 & 0.01 \\ 0.01 & 0.97 & 0.01 & 0.01 \\ 0.01 & 0.97 & 0.01 & 0.01 \end{bmatrix} \quad (5.15)$$

The modification is illustrated in Figure 5.31 as modification 1-1 to modification 1-3. Figure 5.33 illustrates to process changing the Initial Baum Welch Pool Temperature High value from 0 to 0.01. The same process is followed for the other HMM Initial, Transition and emission values.

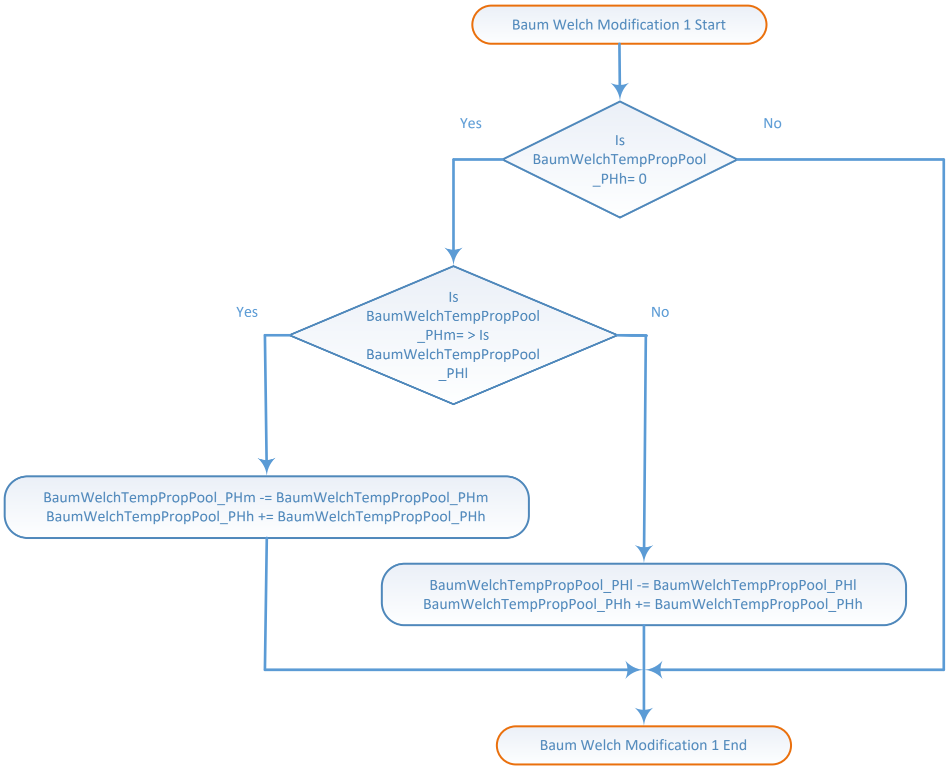


Figure 5.33 Baum-Welch modification one

The same process is done to the emission probabilities from equation 5.4 resulting in equation 5.16

$$B = \begin{bmatrix} 0.01 & 0.94 & 0.04 & 0.01 \\ 0.06 & 0.70 & 0.23 & 0.01 \\ 0.11 & 0.11 & 0.56 & 0.22 \end{bmatrix} \quad (5.16)$$

Next modification one is applied by only add 10% from equation 5.15 and 90% of equation 5.16. The last check is to ensure that the sum of pool probabilities for high, mild and low are one. The final re-estimated emission probabilities are shown in equation 5.17.

$$b' = \begin{bmatrix} 0.01 & 0.94 & 0.04 & 0.01 \\ 0.05 & 0.73 & 0.22 & 0.01 \\ 0.1 & 0.2 & 0.5 & 0.2 \end{bmatrix} \quad (5.17)$$

Lastly the re-estimated probabilities are recorded as the new HMM probabilities where:

$$\lambda = \lambda' \quad (5.18)$$

Appendix M illustrates a snippet of the pool prediction trained output from March to December 2019. The following diagrams illustrate the Baum-Welch algorithm changing the HMM probability parameters over four seasons, summer, autumn, winter and spring.

In Figure 5.34 it is clear the initial high probability is the highest in the summer and start rising from November. Mild is the highest during autumn and spring and low during winter.

Figure 5.35 to Figure 5.37 illustrates the transition probabilities from high to any, mild to any and low to any. In all three diagrams the probability is the best for lower temperatures during the winter shown at point A, higher temperatures during autumns and spring shown at point B. It is important to notice that during the summer when the temperature is high the probability is better for a following mild day, if the temperature is mild after November the probability is slightly higher the following day will be hot.

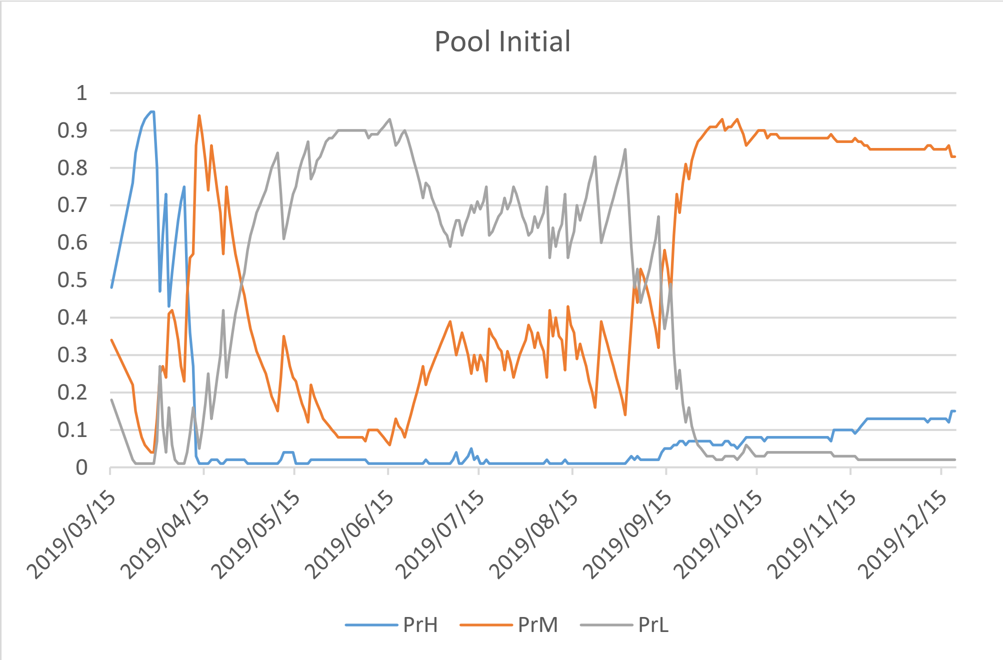


Figure 5.34 *Baum-Welch training Pool HMM model initial probabilities*

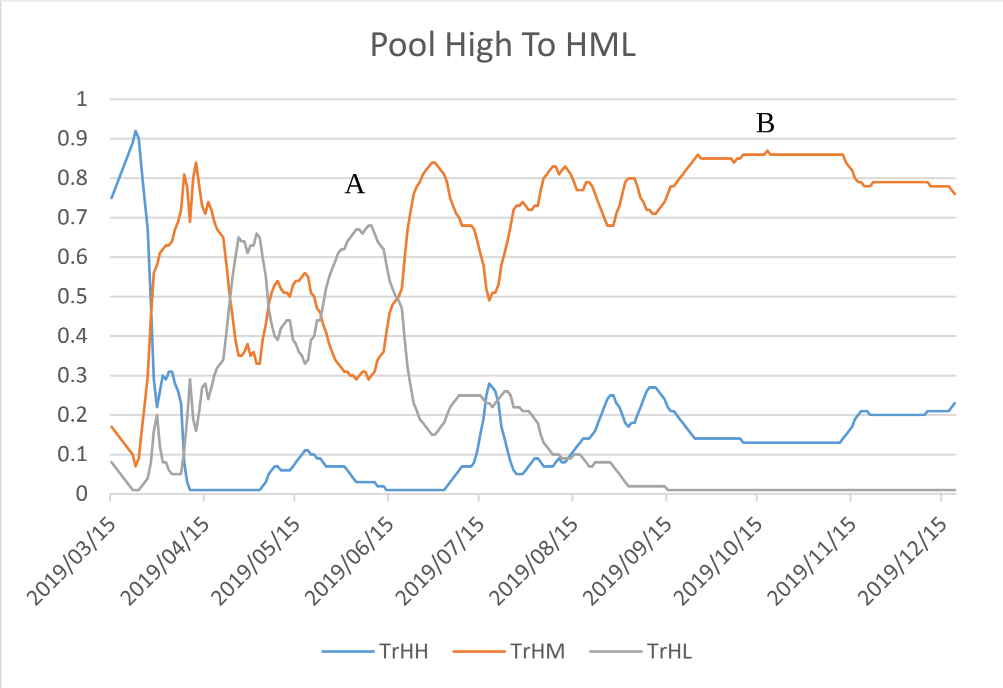


Figure 5.35 *Baum-Welch training Pool High HMM model transition probabilities*

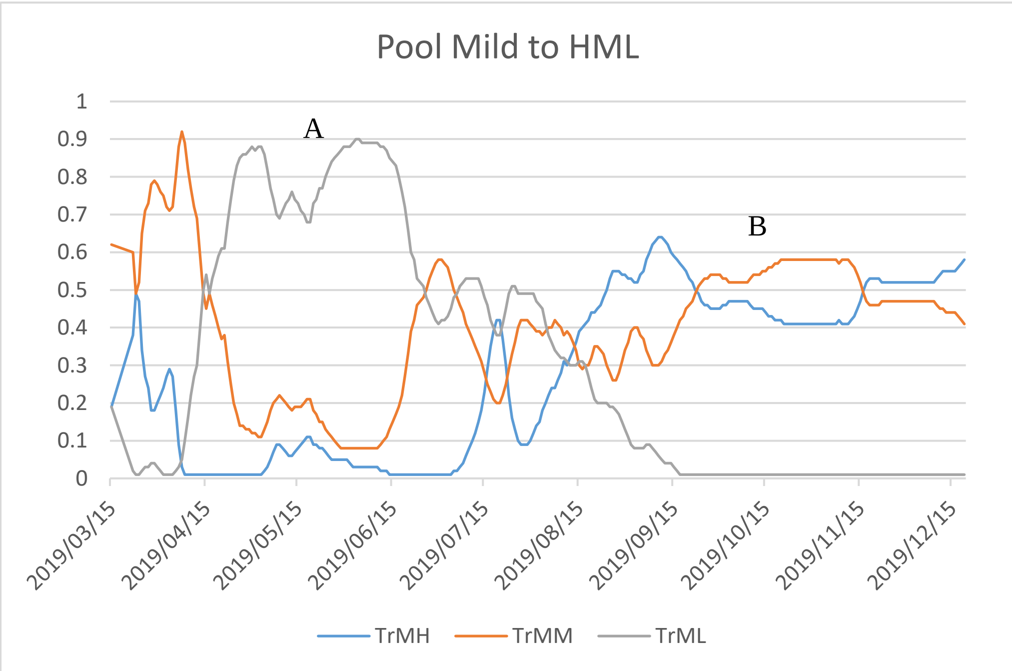


Figure 5.36 *Baum-Welch training Pool Mild HMM model transition probabilities*

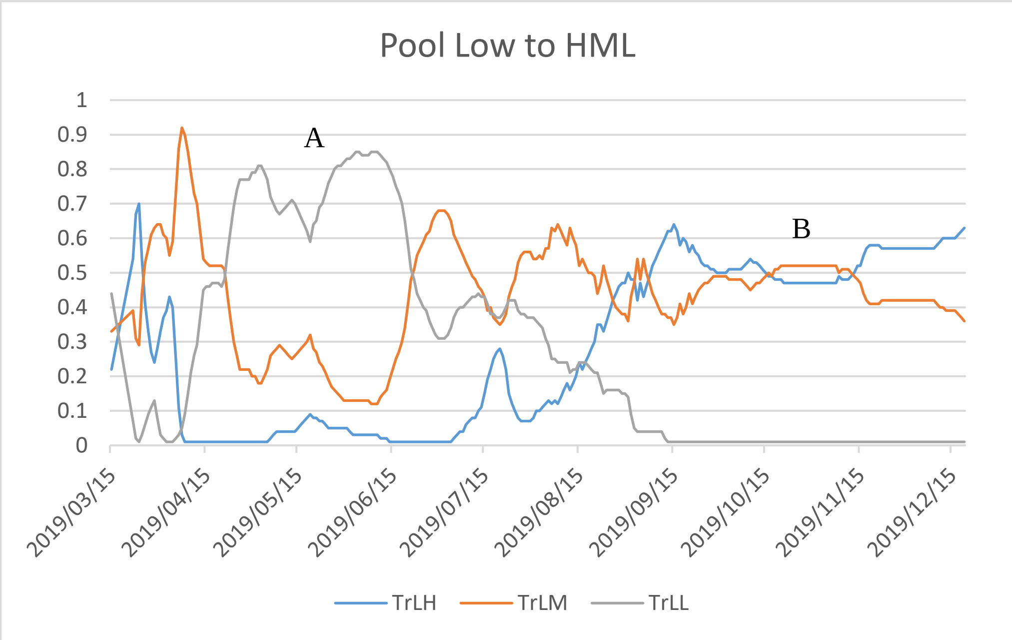


Figure 5.37 Baum-Welch training Pool Low HMM model transition probabilities

Figure 5.38 to Figure 5.40 illustrates the emission probabilities between the weather conditions and the pool temperatures. The diagrams illustrate the HMM model train correctly between seasons and the best scenarios are cold pool temperatures during winter shown at points A, mild during autumn and spring shown at points B and hot during summer shown at points C. The diagrams indicate a change in season end April, which is the start of autumns, June for winter, September for spring. A steep rise in probability for high temperatures is shown from middle October indicating the start of summer. The impact of cloudy weather is minimal during the year except for December when the rainy season starts shown at points D.

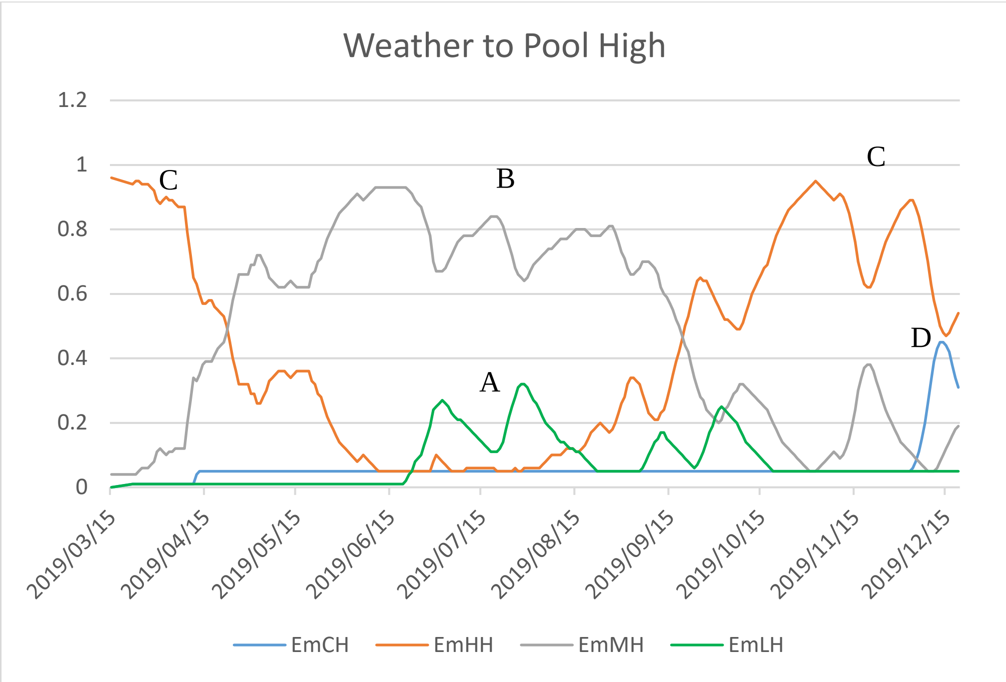


Figure 5.38 Baum-Welch training Pool High HMM model emission probabilities

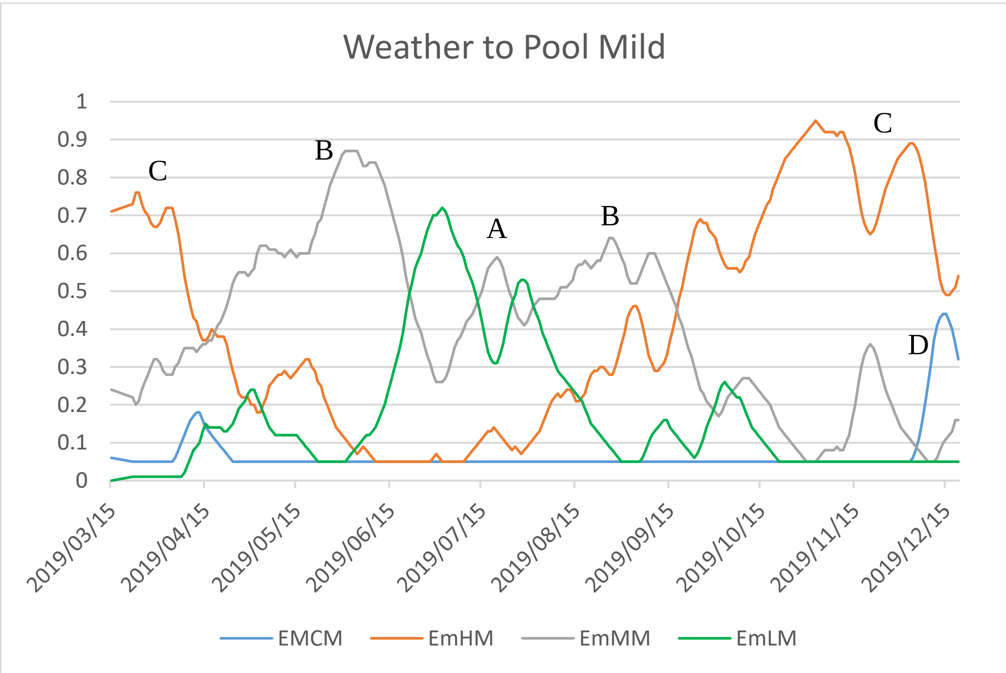


Figure 5.39 Baum-Welch training Pool Mild HMM model emission probabilities

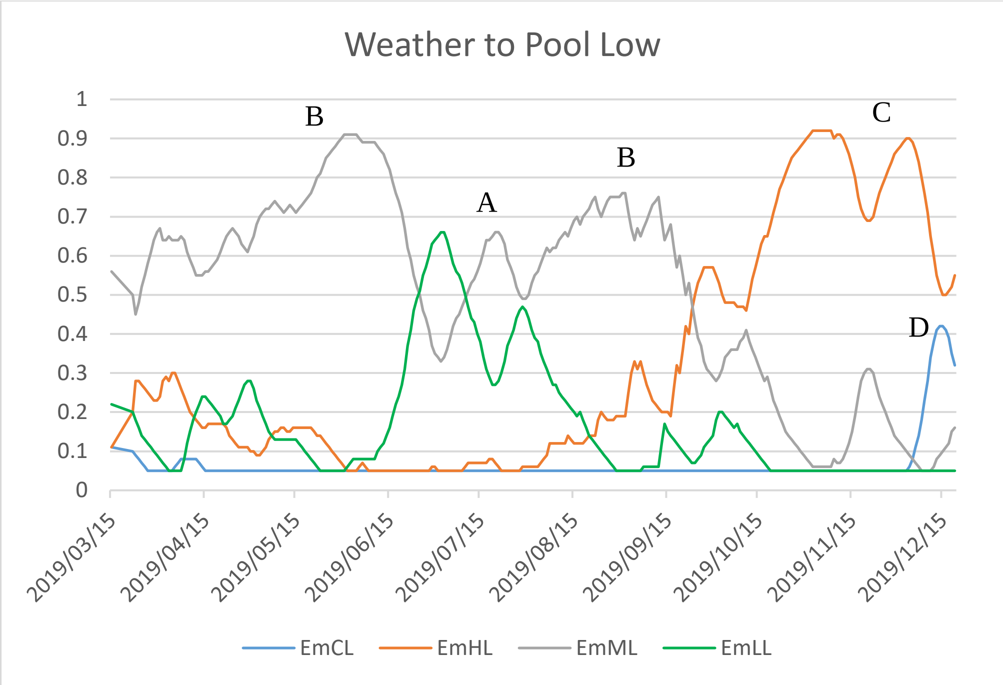


Figure 5.40 *Baum-Welch training Pool Low HMM model emission probabilities*

Figure 5.41 illustrates the module's success rate during the analysed time span. Zero represents an incorrect prediction, one a border prediction and two a correct prediction.

It is important to notice that during summer and winter, the majority correct prediction was made and during spring and autumn the majority border predictions were made, but still there were more correct predictions than border predictions, as the model was adapted to the change in climate.

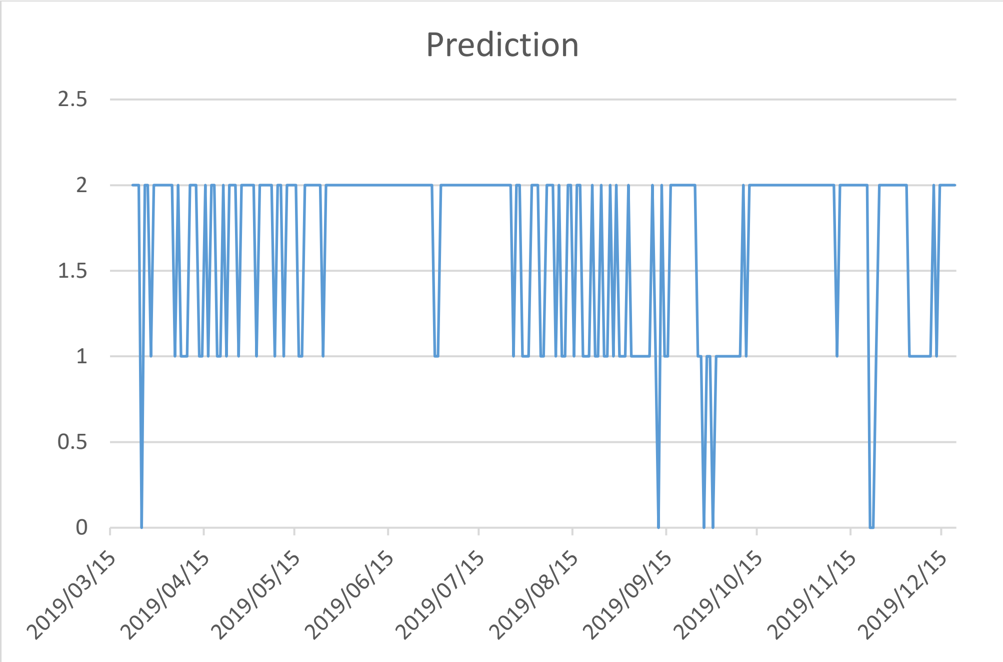


Figure 5.41 Pool prediction accuracies

Figure 5.42 illustrates the difference in kilowatt usage of the pool pump. A comparison is made between usage with a timer, prediction model and the prediction model with pump time maximum if wind speed is higher than 6 km/h. A decision was made to run pump for 4, 5 and 6 hours for low, mild and high predictions.

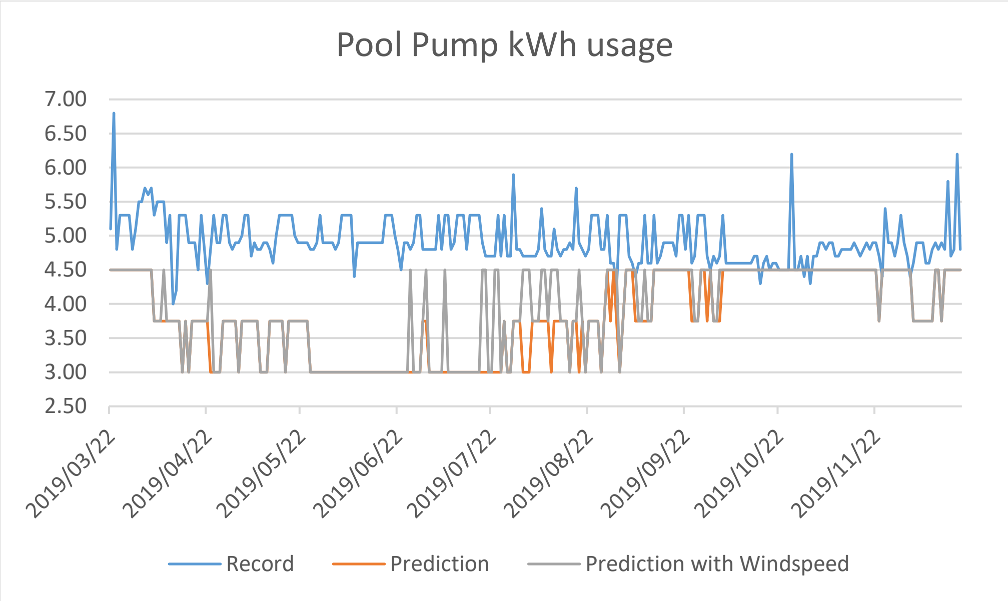


Figure 5.42 Pool Pump kWh usage comparisons

Table 5.13 Domestic tariffs for 60 ampere single phase with cost per usage

Tariff	c/kWh
Block1 (1-1000) kWh	177
Block2 (1001-1500) kWh	189
Block3 (>1500) kWh	196

Table 5.13 indicates the local municipalities' electrical pricing model for 60 amp single-phase power usage.

Table 5.14 Pool pump cost comparisons

Model	kWh	Block1	Block2	Block3
Timer	1355.30	R2 398.88	R2 561.52	R2 656.39
Predicion	1038.75	R1 838.59	R1 963.24	R2 035.95
Predicion with wind	1066.50	R1 887.71	R2 015.69	R2 090.34

Typical cost savings will be between Timer and Prediction with wind speed and the cost savings are R511, R546 and R566 for block1, 2 and 3 cost model. Maximum cost savings can be obtained if the saving result in moving to a lower block cost module. Moving from block 3 to block 2 result in R693 saving and from block 2 to block1 R723 saving. The calculations were done for nine months, from middle March to middle December.

An annual saving between R681 and R964 can be expected.

5.6 *Calculating Viterbi prediction values for Weather prediction with Geyser*

It is important to note that the normalised weather prediction was taken as the observations and the roof temperature as the hidden values. Normally the roof temperatures will be the observation layer predict the weather condition. Weather predictions are accepted and used worldwide and if the weather condition is known to the model, the roof temperature can be predicted and as discussed in section 4.4.4 also the solar geyser efficiency.

5.6.1 **Rainy season: High**

Figure 5.43 illustrates the HMM transition and emission probability state diagram for the weather conditions and roof temperatures from 1 January to 30 April 2018.

Table 5.15 illustrates the initial state probability distribution matrix, Table 5.16 the HMM Transition probability distribution matrix and Table 5.17 the HMM Emission probability distribution matrix for the roof temperatures. The process calculating the values were discussed in section 5.3.

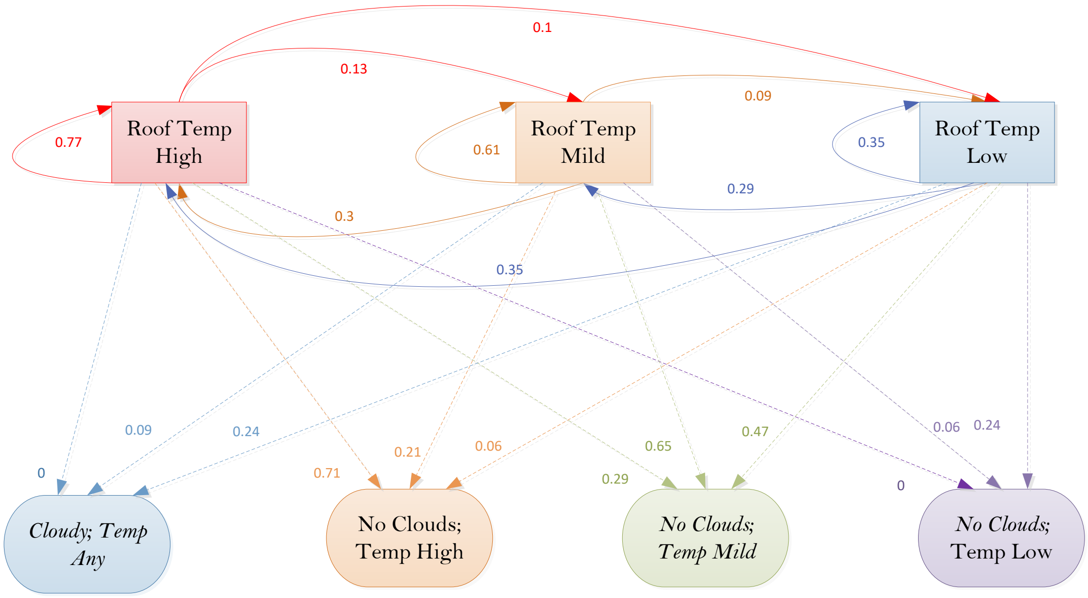


Figure 5.43 Rainy season High: Normalised roof and weather temperatures

Table 5.15 Initial state probability distribution for Roof. Rainy Season: High

Current	Roof High	Roof Mild	Roof Low
Start	0.57	0.28	0.14

Table 5.16 HMM Transition probability distribution for Roof. Rainy Season: High

	Next		
Current	Roof High	Roof Mild	Roof Low
Roof High	0.77	0.13	0.1
Roof Mild	0.3	0.61	0.09
Roof Low	0.35	0.29	0.35

Table 5.17 HMM Emission probability distribution for Roof. Rainy Season: High

	Observation / Emission			
State	Temp: Any Cloudy	Temp: High No Clouds	Temp: Mild No Clouds	Temp: Low No Clouds
Roof High	0	0.71	0.29	0
Roof Mild	0.09	0.21	0.65	0.06
Roof Low	0.24	0.06	0.47	0.24

Figure 5.44 to Figure 5.47 illustrates the results, including comparison to actual recorded data if the HMM states are used for historical data for 2018.

5 Day Prediction Viterbi Algorithm

Weather

Date	2018/1/10	2018/1/11	2018/1/12	2018/1/13	2018/1/14
Weather	M	M	M	H	H

Roof Temperature

High	0.1653	0.036911	0.008242	0.006095	0.003332
Mild	0.182	0.072163	0.028613	0.000186	9E-06
Low	0.0658	0.010824	0.003052	0.000155	3.7E-05
Recorded	M	M	H	H	H

Figure 5.44 Five day roof temperatures Viterbi prediction 10 to 14 January 2018

Example 5.1 explains the mathematical process calculating the Viterbi states between 10 and 14 January 2018. This process was discussed in section 5.4 via the flow charts illustrated in Figure 5.10 and Figure 5.11.

Example 5.1

HMM states were calculated during the transition and emission probability phase as illustrate in Table 5.15 to Table 5.17.

Day 0:

ViterbiHigh[0] = Probability_RoofHigh x Emission_RoofHighWeatherHigh

= 0.57 x 0.29

= 0.1653

ViterbiMild[0] = Probability_RoofMild x Emission_RoofMildWeatherHigh

= 0.28 x 0.65

= 0.182

ViterbiLow[0] = Probability_RoofLow x Emission_RoofMildWeatherHigh

= 0.14 x 0.47

= 0.0658

Highest value ViterbiHigh[0] is the same as the recorder status for 10 Jan 2018 as illustrated in Figure 5.44. The probability is a High roof temperature.

Viterbi[0] = ViterbiMild[0] = 0.182

Day 1:

ViterbiHigh_1[1] = ViterbiHigh[0] x Transition_RoofHighHigh x Emission_RoofHighWeatherMild

= 0.1653 x 0.77 x 0.29

= 0.36911

ViterbiHigh_2[1] = ViterbiMild[0] x Transition_RoofMildHigh x Emission_RoofHighWeatherMild

= 0.182 x 0.3 x 0.29

= 0.015833

ViterbiHigh_3[1] = ViterbiLow[0] x Transition_RoofLowHigh x Emission_RoofHighWeatherMild

= 0.0658 x 0.35 x 0.29

= 0.0066

ViterbiHigh[1] = ViterbiHigh_1[1] = 0.036911

$ViterbiMild_1[1] = ViterbiHigh[0] \times Transition\_RoofHighMild \times$

$Emission\_RoofMildWeatherMild$

$= 0.1653 \times 0.13 \times 0.65$

$= 0.013967$

$ViterbiMild_2[1] = ViterbiMild[0] \times Transition\_RoofMildMild \times$

$Emission\_RoofMildWeatherMild$

$= 0.182 \times 0.61 \times 0.65$

$= 0.072163$

$ViterbiMild_3[1] = ViterbiLow[0] \times Transition\_RoofLowMild \times$

$Emission\_RoofMildWeatherMild$

$= 0.0658 \times 0.29 \times 0.65$

$= 0.012403$

$ViterbiMild[1] = ViterbiMild_2[1] = 0.072163$

$ViterbiLow_1[1] = ViterbiHigh[0] \times Transition\_RoofHighLow \times$

$Emission\_RoofLowWeatherMild$

$= 0.1653 \times 0.1 \times 0.47$

$= 0.007769$

$ViterbiLow_2[1] = ViterbiMild[0] \times Transition\_RoofMildLow \times$

$Emission\_RoofLowWeatherMild$

$= 0.182 \times 0.09 \times 0.47$

$= 0.0076985$

$ViterbiLow_3[1] = ViterbiLow[0] \times Transition\_RoofLowLow \times Emission\_RoofLowWeatherMild$

$= 0.0658 \times 0.35 \times 0.47$

$= 0.010824$

$ViterbiLow[1] = ViterbiLow_2[1] = 0.010824$

Highest value ViterbiHigh[1] is the same as the recorder status for 11 Jan 2018 as illustrated in Figure 5.44. The probability is a High roof temperature.

$Viterbi[1] = ViterbiMild[1] = 0.072163$

Calculating the roof temperature probabilities for Day[2] to Day[4] are done in the same manner as Day[1].

5 Day Prediction Viterbi Algorithm

Weather

Date	2018/2/13	2018/2/14	2018/2/15	2018/2/16	2018/2/17
Weather	H	M	Cl	H	H

Roof Temperature

High	0.4047	0.09037	0	0.000539	0.000295
Mild	0.0588	0.034197	0.001877	0.000132	3E-06
Low	0.0084	0.019021	0.002169	4.6E-05	3E-06
Recorded	H	H	L	H	H

Figure 5.45 Five day roof temperature Viterbi prediction 13 to 17 February 2018

5 Day Prediction Viterbi Algorithm

Weather

Date	2018/3/20	2018/3/21	2018/3/22	2018/3/23	2018/3/24
Weather	H	H	Cl	Cl	M

Roof Temperature

High	0.4047	0.221249	0	0	4.5E-05
Mild	0.0588	0.000512	0.002589	0.000142	8.4E-05
Low	0.0084	0.002428	0.00531	0.000446	7.3E-05
Recorded	M	H	L	L	M

Figure 5.46 Five day roof temperature Viterbi prediction 20 to 24 March 2018

5 Day Prediction Viterbi Algorithm

Weather					
Date	2018/4/12	2018/4/13	2018/4/14	2018/4/15	2018/4/16
Weather	L	L	M	M	M
Roof Temperature					
High	0	0	0.000286	6.4E-05	1.8E-05
Mild	0.0168	0.000615	0.000532	0.000211	8.4E-05
Low	0.0336	0.002822	0.000464	7.6E-05	1.3E-05
Recorded	L	M	M	M	L

Figure 5.47 Five day roof temperature Viterbi prediction 12 to 16 April 2018

The HMM values accuracy since January to April is shown in Table 5.18. Calculating the accuracy, two points were given for the correct roof temperate collection and one point if the neighbouring temperature collection were calculated.

Table 5.18 Rainy Season: High - Roof temperature probability accuracy

	Month			
Accuracy (%)	Jan	Feb	Mar	Apr
Day [0]	100	100	50	100
Day [1]	100	100	100	50
Day [0 - 4]	90	100	90	80

The accuracy has been 100% correct for the first predicted day except for March when High was calculated but Mild was recorded. Only during April the following day, Day [1] where incorrect calculating the neighbour temperature collection. The 5-day probability where April the lowest and February 100% correct.

5.6.2 Rainy season: Low

Figure 5.48 illustrates the HMM transition and emission probability state diagram for the weather conditions and roof temperatures from 1 October to 31 December 2018.

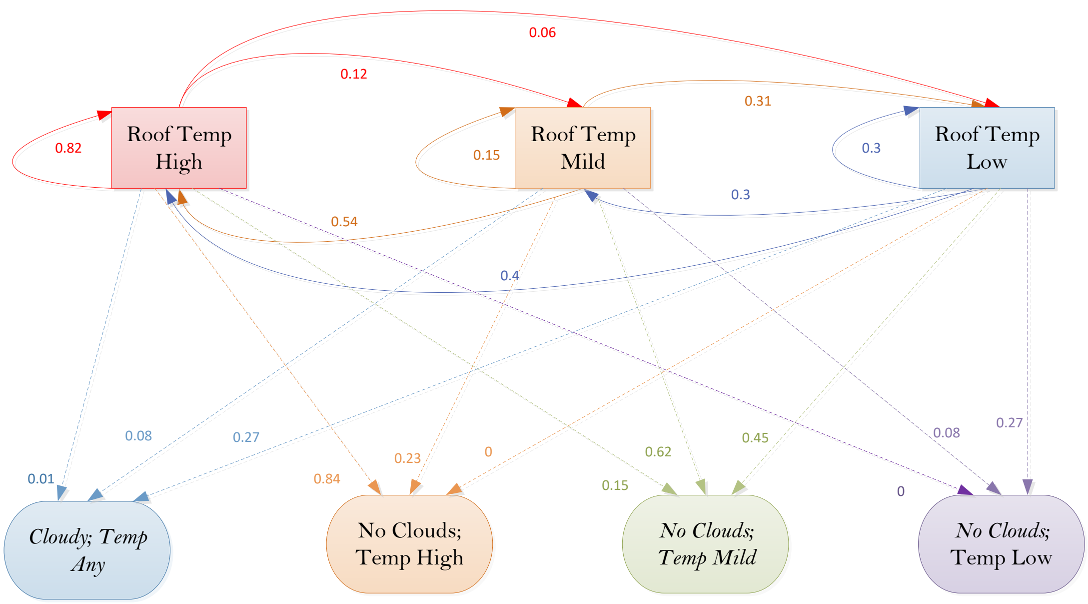


Figure 5.48 Rainy season Low: Normalised roof and weather temperatures

Table 5.19 illustrates the initial state probability distribution matrix, Table 5.20 the HMM Transition probability distribution matrix and Table 5.21 the HMM Emission probability distribution matrix for the roof temperatures. The process calculating the values were discussed in section 5.3.

Table 5.19 Initial state probability distribution for Roof. Rainy Season: Low

Current	Roof High	Roof Mild	Roof Low
Start	0.74	0.14	0.12

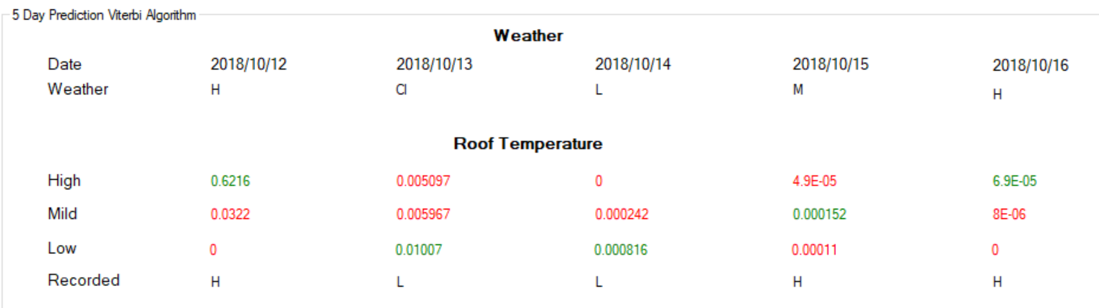
Table 5.20 HMM Transition probability distribution for Roof. Rainy Season: Low

	Next		
Current	Roof High	Roof Mild	Roof Low
Roof High	0.82	0.12	0.06
Roof Mild	0.54	0.15	0.31
Roof Low	0.4	0.3	0.3

Table 5.21 HMM Emission probability distribution for Roof. Rainy Season: Low

	Observation / Emission			
State	Temp: Any Cloudy	Temp: High No Clouds	Temp: Mild No Clouds	Temp: Low No Clouds
Roof High	0.01	0.84	0.15	0
Roof Mild	0.08	0.23	0.62	0.08
Roof Low	0.27	0	0.45	0.27

Figure 5.49 to Figure 5.51 illustrates the Viterbi results for the selected five-days during October to December 2018. The method calculating the results is the same as during the high rainy season described in section 5.6.1 example 5.1, except that this seasons HMM transition and emission values will be used.


5 Day Prediction Viterbi Algorithm

Weather

Date	2018/10/12	2018/10/13	2018/10/14	2018/10/15	2018/10/16
Weather	H	Cl	L	M	H

Roof Temperature

High	0.6216	0.005097	0	4.9E-05	6.9E-05
Mild	0.0322	0.005967	0.000242	0.000152	8E-06
Low	0	0.01007	0.000816	0.00011	0
Recorded	H	L	L	H	H

Figure 5.49 Five day roof temperature Viterbi prediction 12 to 16 October 2018

5 Day Prediction Viterbi Algorithm

Weather

Date	2018/11/18	2018/11/19	2018/11/20	2018/11/21	2018/11/22
Weather	H	H	M	H	H

Roof Temperature

High	0.6216	0.428158	0.052663	0.036274	0.024986
Mild	0.0322	0.017156	0.031855	0.000798	0.001001
Low	0	0	0.01156	0	0
Recorded	H	H	M	H	H

Figure 5.50 Five day roof temperature Viterbi prediction 18 to 22 November 2018

5 Day Prediction Viterbi Algorithm

Weather

Date	2018/12/15	2018/12/16	2018/12/17	2018/12/18	2018/12/19
Weather	H	H	H	H	H

Roof Temperature

High	0.6216	0.428158	0.294915	0.203137	0.139921
Mild	0.0322	0.017156	0.011817	0.00814	0.005607
Low	0	0	0	0	0
Recorded	H	H	H	H	H

Figure 5.51 Five day roof temperature Viterbi prediction 15 to 19 December 2018

The HMM values accuracy for October to December is shown in Table 5.22. Calculating the accuracy, two points were given for the correct roof temperate collection and one point if the neighbouring temperature collection were calculated.

Table 5.22 Rainy Season: Low - Pool temperature probability accuracy

	Month		
Accuracy (%)	Oct	Nov	Dec
Day 0	100	100	100
Day1	100	100	100
Day 0 to 4	90	90	100

The accuracy has been 100% correct for the first predicted two day's Day[0] and Day[1]. The 5-day probability where 100% correct for December and 90% in October and November.

5.6.3 **Rainy season: None**

Figure 5.52 illustrates the HMM transition and emission probability state diagram for the weather conditions and roof temperatures from 1 May to 30 September 2018.

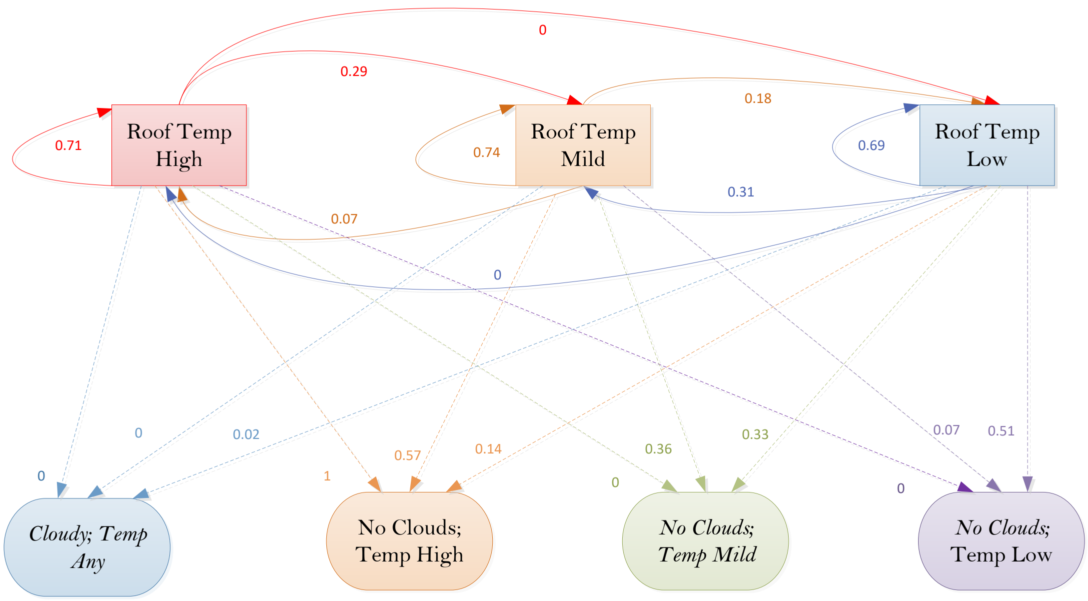


Figure 5.52 Rainy season None: Normalised roof and weather temperatures

Table 5.23 illustrates the initial state probability distribution matrix, Table 5.24 the HMM Transition probability distribution matrix and Table 5.25 the HMM Emission probability distribution matrix for the roof temperatures. The process calculating the values were discussed in section 5.3.

Table 5.23 Initial state probability distribution for Roof. Rainy Season: None

Current	Roof High	Roof Mild	Roof Low
Start	0.14	0.54	0.32

Table 5.24 HMM Transition probability distribution for Roof. Rainy Season: None

	Next		
Current	Roof High	Roof Mild	Roof Low
Roof High	0.71	0.29	0
Roof Mild	0.07	0.74	0.18
Roof Low	0	0.31	0.69

Table 5.25 HMM Emission probability distribution for Roof. Rainy Season: None

	Observation / Emission			
State	Temp: Any Cloudy	Temp: High No Clouds	Temp: Mild No Clouds	Temp: Low No Clouds
Roof High	0	1	0	0
Roof Mild	0	0.57	0.36	0.07
Roof Low	0.02	0.14	0.33	0.51

Figure 5.53 to Figure 5.57 illustrates the Viterbi results for the selected five-days during May to September 2018. The method calculating the results is the same as during the high rainy season described in section 5.6.1 example 5.1, except that this seasons HMM transition and emission values will be used.

5 Day Prediction Viterbi Algorithm

Weather

Date	2018/5/15	2018/5/16	2018/5/17	2018/5/18	2018/5/19
Weather	L	L	M	H	M

Roof Temperature

High	0	0	0	0.000449	0
Mild	0.0378	0.003541	0.006409	0.002311	0.000616
Low	0.1632	0.05743	0.013077	0.001263	0.000288
Recorded	L	L	L	L	L

Figure 5.53 Five day roof temperature Viterbi prediction 15 to 19 May 2018

5 Day Prediction Viterbi Algorithm

Weather

Date	2018/6/14	2018/6/15	2018/6/16	2018/6/17	2018/6/18
Weather	M	H	L	M	M

Roof Temperature

High	0	0.013608	0	0	0
Mild	0.1944	0.01866	0.000967	0.000401	0.000107
Low	0.1056	0.010201	0.00359	0.000817	0.000186
Recorded	L	M	L	L	M

Figure 5.54 Five day roof temperature Viterbi prediction 14 to 18 June 2018

5 Day Prediction Viterbi Algorithm

Weather

Date	2018/7/15	2018/7/16	2018/7/17	2018/7/18	2018/7/19
Weather	L	L	L	L	M

Roof Temperature

High	0	0	0	0	0
Mild	0.0378	0.003541	0.001246	0.000439	0.000794
Low	0.1632	0.05743	0.02021	0.007112	0.001619
Recorded	L	L	L	L	L

Figure 5.55 Five day roof temperature Viterbi prediction 15 to 19 July 2018

5 Day Prediction Viterbi Algorithm

Weather

Date	2018/8/17	2018/8/18	2018/8/19	2018/8/20	2018/8/21
Weather	H	H	M	H	H

Roof Temperature

High	0.14	0.0994	0	0.000726	0.000515
Mild	0.3078	0.007916	0.010377	0.000312	4.6E-05
Low	0.0448	0.007757	0.001766	0.000262	2.5E-05
Recorded	M	L	M	M	M

Figure 5.56 Five day roof temperature Viterbi prediction 17 to 21 August 2018

5 Day Prediction Viterbi Algorithm

Weather

Date	2018/9/9	2018/9/10	2018/9/11	2018/9/12	2018/9/13
Weather	L	M	H	H	H

Roof Temperature

High	0	0	0.001275	0.000905	0.000643
Mild	0.0378	0.018213	0.006566	0.000634	6.1E-05
Low	0.1632	0.037161	0.00359	0.000347	3.4E-05
Recorded	L	M	H	H	H

Figure 5.57 Five day roof temperature Viterbi prediction 9 to 13 Septemper 2018

The HMM values accuracy from May to September are shown in Table 5.26. Calculating the accuracy, two points were given for the correct roof temperate collection and one point if the neighbouring temperature collection were calculated.

Table 5.26 Rainy Season: Low - Pool temperature probability accuracy

	Month				
Accuracy (%)	May	Jun	Jul	Aug	Sep
Day [0]	100	50	100	100	100
Day [1]	100	100	100	0	50
Day [0 - 4]	80	80	100	60	80

The accuracy has been 100% correct for the first predicted day except for June when Mild was calculated but Low was recorded. During May, June and July the following day, Day [1] were correct. The 5-day probability August where the lowest and July the highest.

5.6.4 Baum-Welch algorithm training one HMM model

As proven with the pool prediction the latest seasonal HMM model will be used training only one model. The geyser prediction model is more complex than the swimming pool. The first phase is based on the same principle heating the swimming pool. The prediction model must decide in the morning what the probability is heating the geyser water via the solar panel. As the second modifications discussed at equation 5.14 proven insufficient, it was excluded. Figure 5.58 indicates the same pattern as with heating the pool water that the best predictions are obtained with an observation sequence between 7 and 14 days with modification one. The modification again results in a higher accuracy prediction as with the normal Baum-Welch algorithm. Seven day’s observations was chosen as it was used by the pool predictions.

Calculating the initial, transition and emission probabilities are the same as discussed with flow charts illustrated in Figure 5.28 to Figure 5.31.

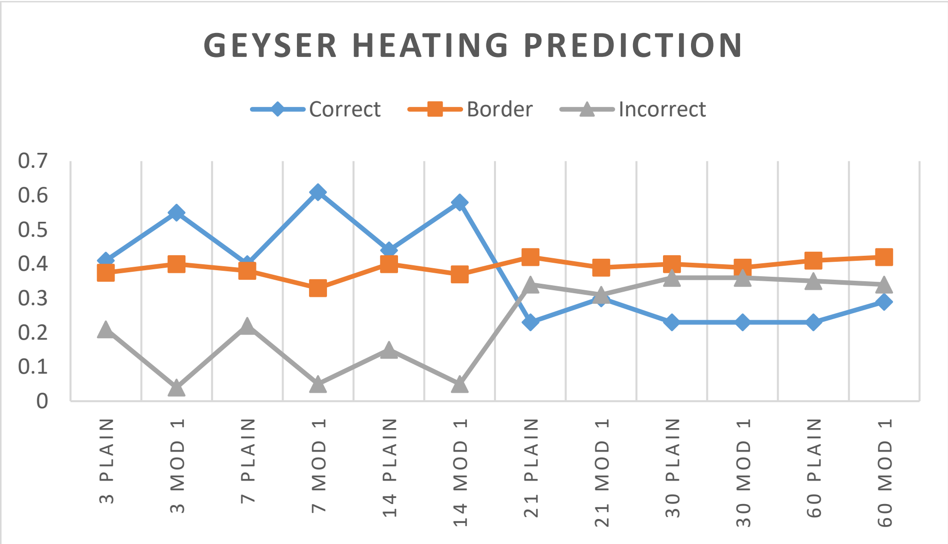


Figure 5.58 Observation lengths accurate comparison

The following diagrams illustrate the Baum Welch algorithm output from January to December 2018 algorithm changing the HMM probability parameters over four seasons. It is important, as discussed in section 4.4, that the roof temperature is used instead of the geyser temperature as the geyser could be heated with an electrical element create unreliable solar heating recordings.

Figure 5.59 illustrates the change in initial probabilities whereas predicted probability to heat the roof to High is the best during the summer and Mild during spring. During winter it is noticeable that the probabilities are almost identical for High, Mild, and Low.

Appendix N illustrates the Baum-Welch output and Figure 5.60 to Figure 5.62, the HMM model training the transition probabilities from high to any, mild to any and low to any.

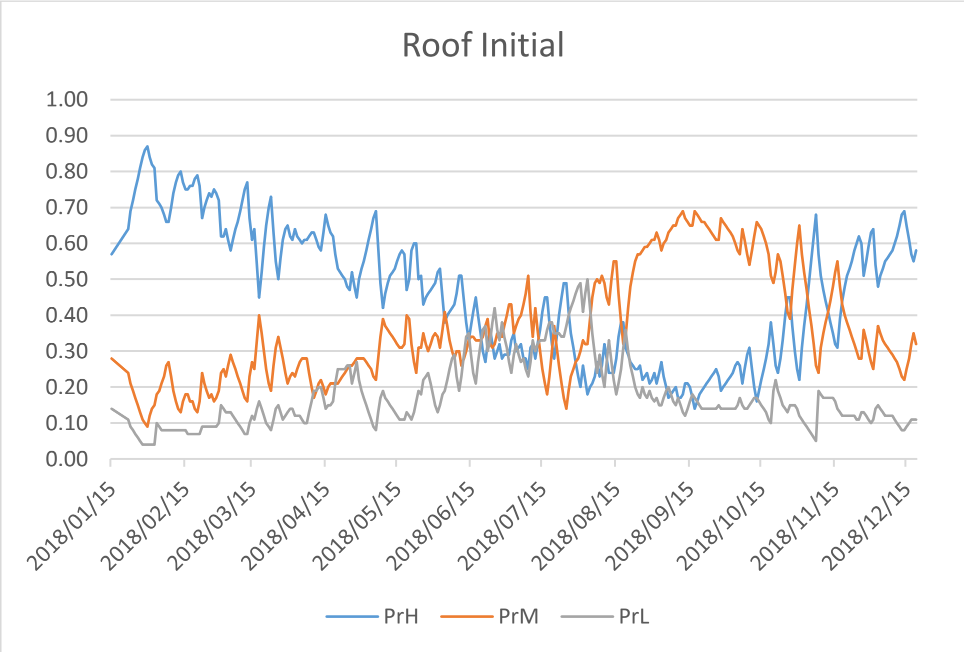


Figure 5.59 *Baum-Welch training Roof HMM model initial probabilities*

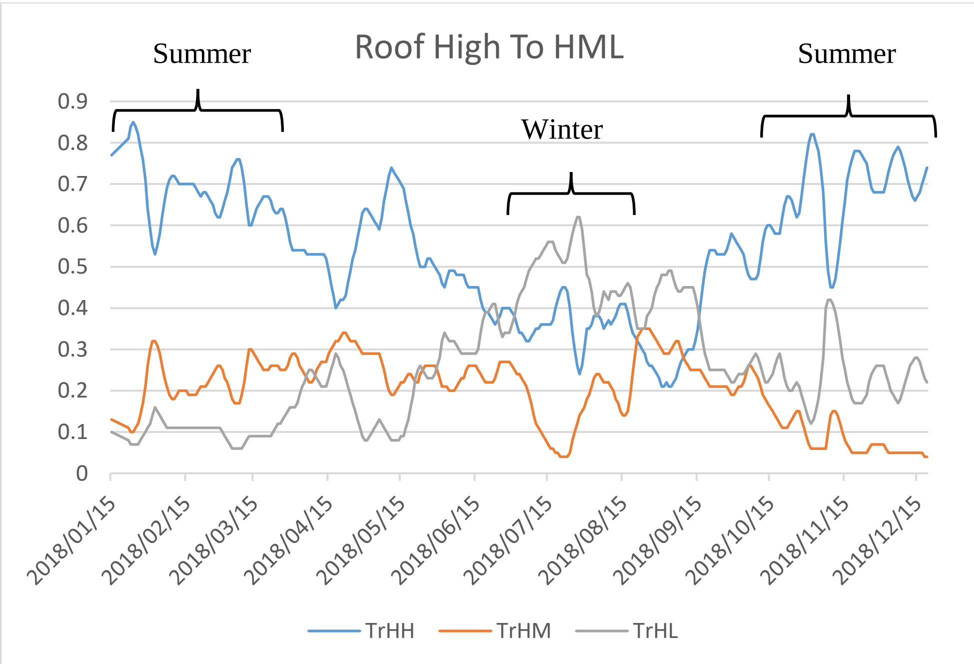


Figure 5.60 *Baum-Welch training Roof High HMM model transition probabilities*

The high transition diagram illustrates the probability for a high following with a high is the best during the summer and high following by a low during the winter.

The Mild transition diagram point A, illustrates the probabilities are low the entire year for a mild temperature to be followed by a low temperature.

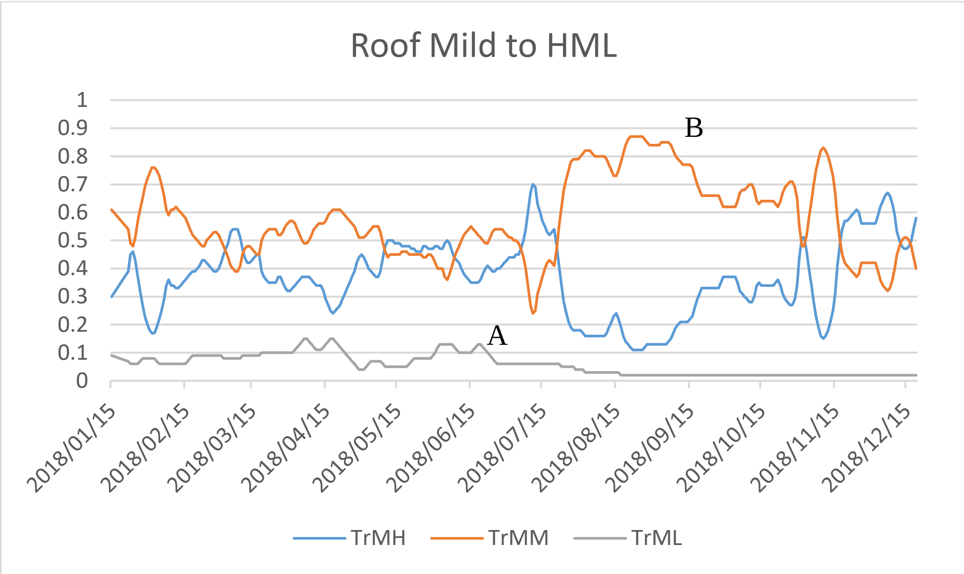


Figure 5.61 Baum-Welch training Roof Mild HMM model transition probabilities

The Mild and Low transition diagram point B, highlights that during spring and early summer the probability is the highest for a mild roof temperature. During winter, June and July point C, there are periods where a Low temperature is the best, followed by a Low at point D, but around start July the probability is better that a Mild temperature will follow a Low temperature at point C. An explanation for that is when the last cold fronts of the winter season passed through the area monitored.

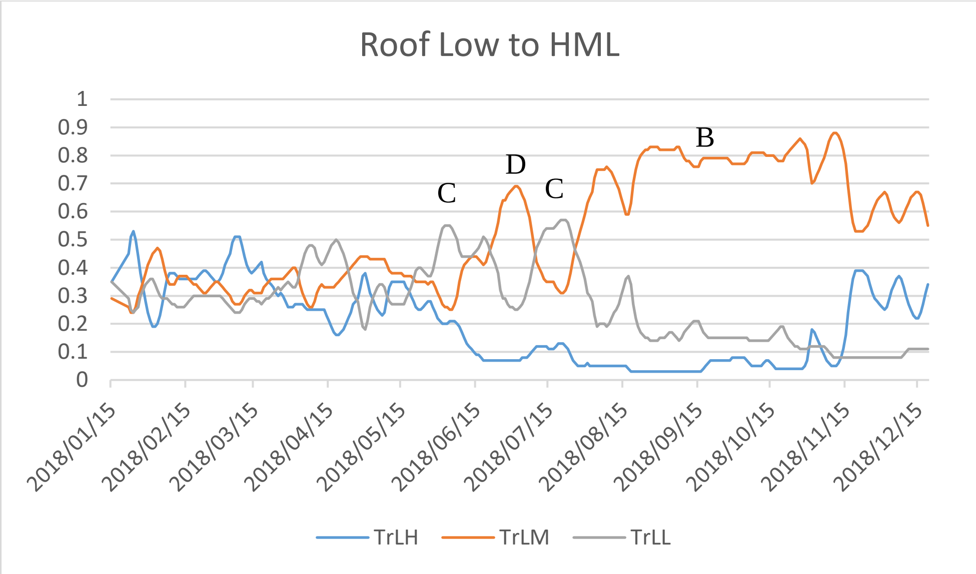


Figure 5.62 *Baum-Welch training Roof Low HMM model transition probabilities*

Figure 5.63 to Figure 5.65 illustrates the emission probabilities between the weather conditions and the roof temperatures. The diagrams illustrate the HMM model train correctly between seasons and that at points A, the best scenarios for high are during spring and early summer months which are between September and December.

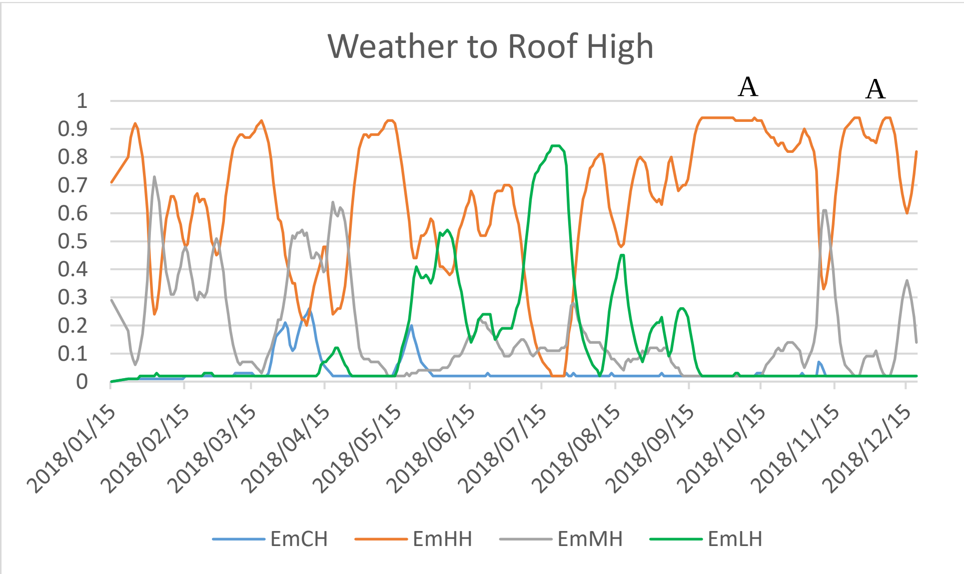


Figure 5.63 *Baum-Welch training Roof High HMM model emission probabilities*

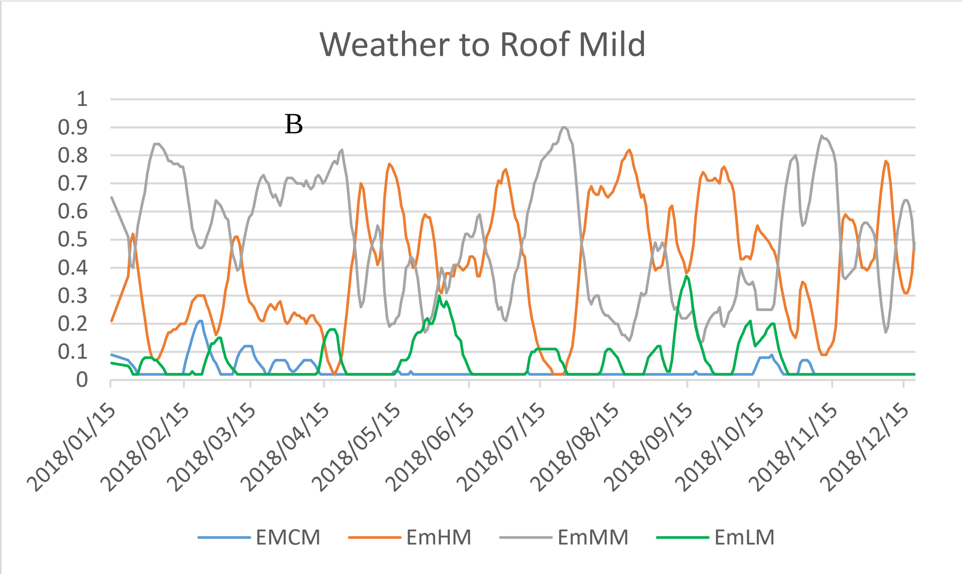


Figure 5.64 Baum-Welch training Roof Mild HMM model emission probabilities

During the rainy season, February to April, the probability where the best in a mild roof's temperature shown at point B, a mild followed by a low shown at point C and low temperature during winter shown at point D.

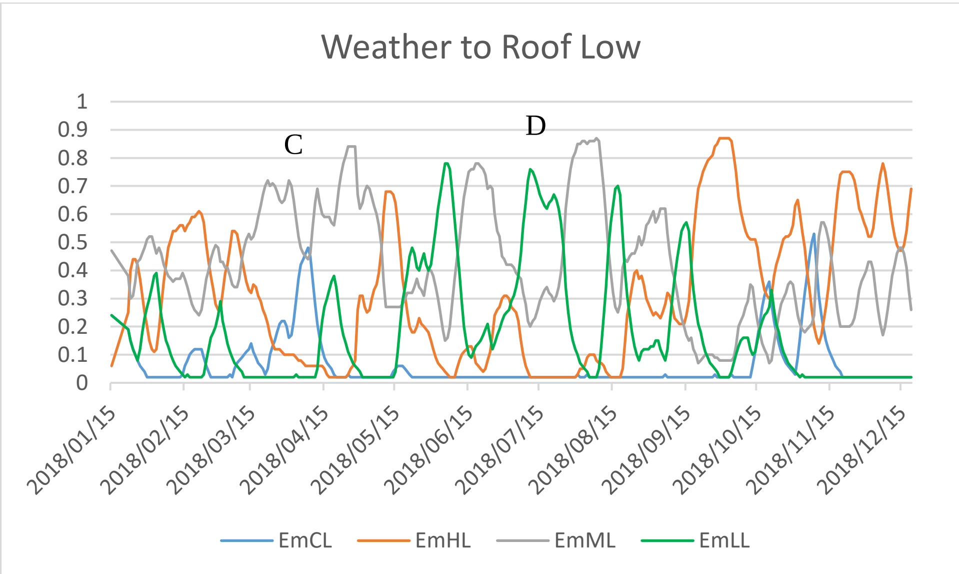


Figure 5.65 Baum-Welch training Roof Low HMM model emission probabilities

As the geyser water temperatures must be at certain temperatures during the day, it is important to predict the usage before a decision are made to when and how hot to heat the water with an electrical element.

5.7 *Predicting hot water usage*

The fuzzy logic output develops in section 4.1 will now be used to predict the hot water usage per hour.

As all the data were already processed via fuzzy logic and with no hidden component the probability that the water usage is None, Small, Medium or Large was calculated by the sum per usage, divided by the sum of samples as defined by Bayes' theorem as discussed at equation 2.72 where $p(A|B) = [p(A).p(B|A)] / p(B)$.

Figure 5.66 and Figure 5.67 illustrates the highest usage per hour for 2017 and 2018 per hour. The outcome is almost the same for one year or two year's samples.

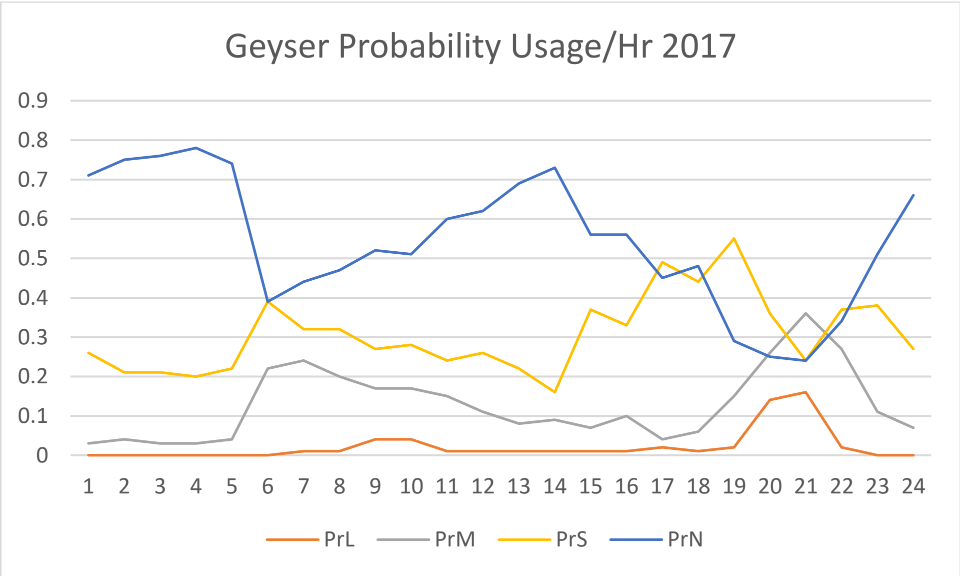


Figure 5.66 Hot water usages per hour for 2017

Keeping track of usage per hour is the most accurate, but also requires the most processing and memory resources. Investigating Figure 5.67 further by dividing the usage in three hour slots. First, timeslots where created starting from 0h00 to 2h59, 3h00 to 5h59, …., until 21h00 to 23h59 as indicated by the green lines, marked as A. Investigating Figure 5.67 it was noticeable, the hot water usage only changing from None too Small between 4h00 and 5h00.

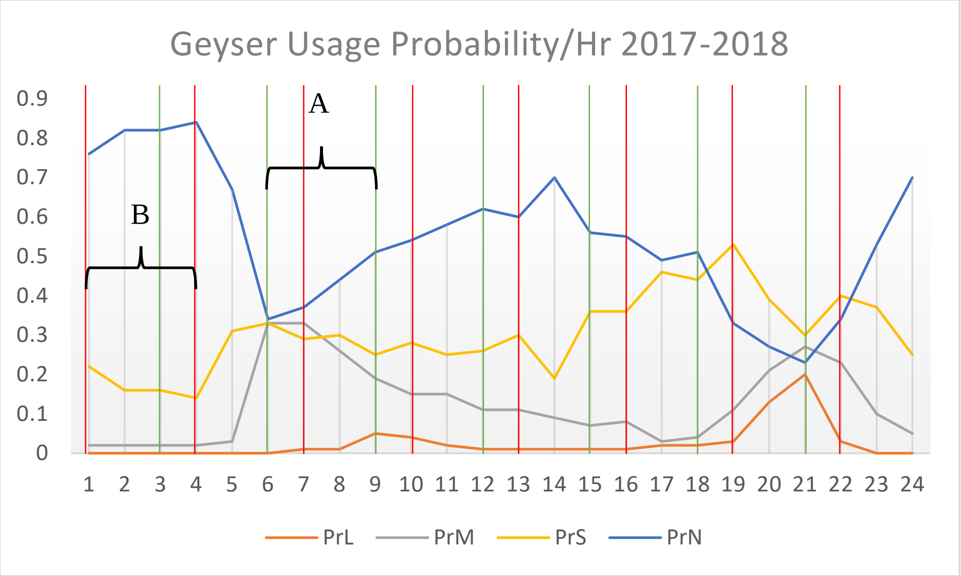


Figure 5.67 Hot water usages per hour for 2017 and 2018

The system must start heating the water before 3h00 to ensure correct water temperature starting timeslot 3h00 to 5h59. If the time slots are forwarded with an hour, as illustrated by the red lines and marked as B, the time slots will be from 1h00 to 3h59, 4h00 to 6h59, …., until 22h00 to 0h59 presentation the hot water usage better. The Large usages also fall in the timeslot 19h00 to 21h59, otherwise the system would require heating the water for Large usage also between 21h00 and 23h00.

Figure 5.68 illustrated the newly computed hot water usage per three hour time slots. It is important that the trend of usage per hour and three hours did stay the same with little loss in weight per category per time slot.

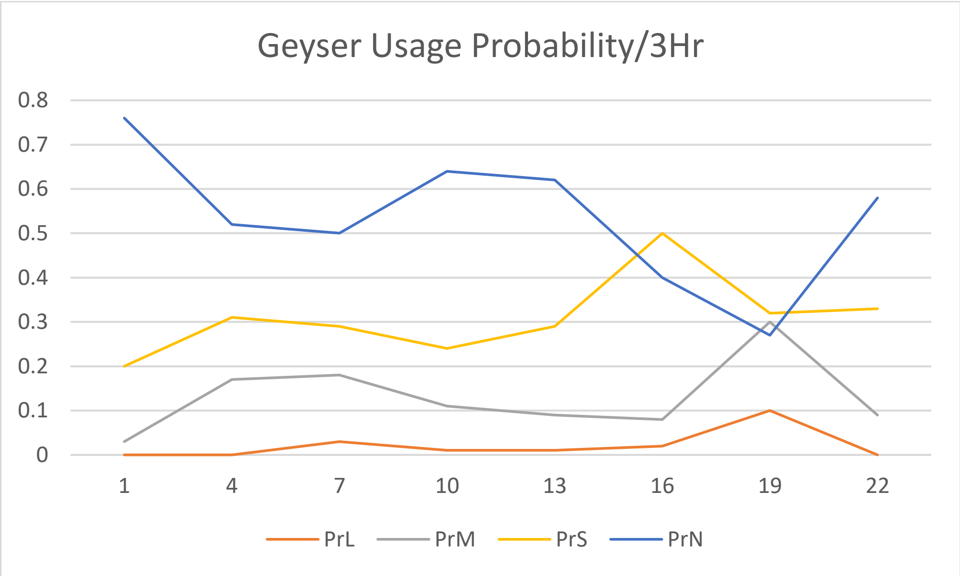


Figure 5.68 Hot water usage per three hour timeslots for 2017 and 2018

Figure 5.69 and Figure 5.70. First, historical data used must be the same as for solar geyser heating as discussed in section 5.6.4. Seven days was used to train the prediction model.

Calculating the number of usages “None, Small, Medium and Large” are illustrated in Figure 5.69. For each hour the SQL statement needs to be updated to retrieve a specific hour’s usage data for the past seven days. This data is used in Figure 5.70 to split the usages per time slot. The time slot selection was discussed earlier in this section. Normalise the usages by dividing them with the sum of all usages for a specific time slot.

The same principle discussed in section 5.6.4 calculating the training model, using a specific weight newly calculated hot water usage with historical usage, was used. The percentage used will be discussed in Figure 5.71.

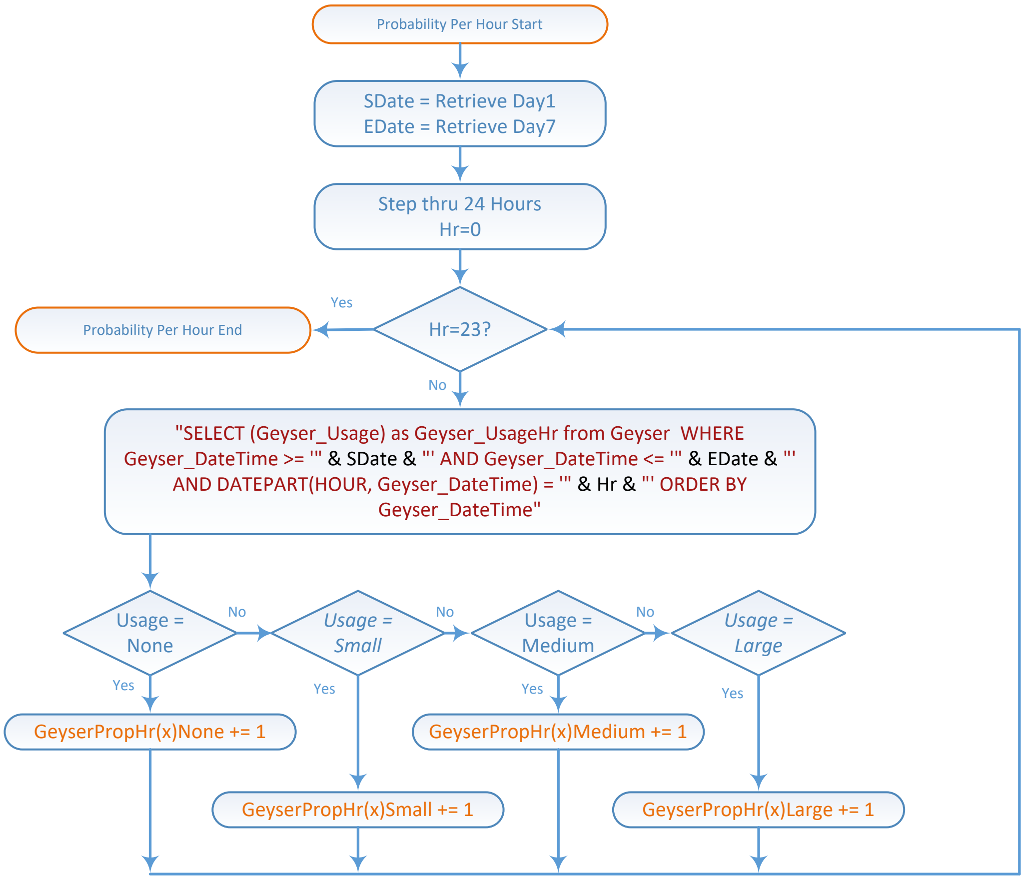


Figure 5.69 Flow chart one predicting hot water usage

Newly calculated usages need to be re-normalised to ensure the sum of usages per hour is one. Through simulations it was determined that the highest percentage usage per time slot cannot be obtained by using data of the 14th January 2018 Timeslot 6 for example:

None 45%, Small 42%, Medium 8% and Large 4%.

The system cannot produce a result of usage was None. A new variable was included determine from UsageHigh to UsageLow the minimum percentage used, determining the usage selection. For example, if the cut-off is 5% the prediction will be Medium but if the cut-off is 4% the prediction will be Large.

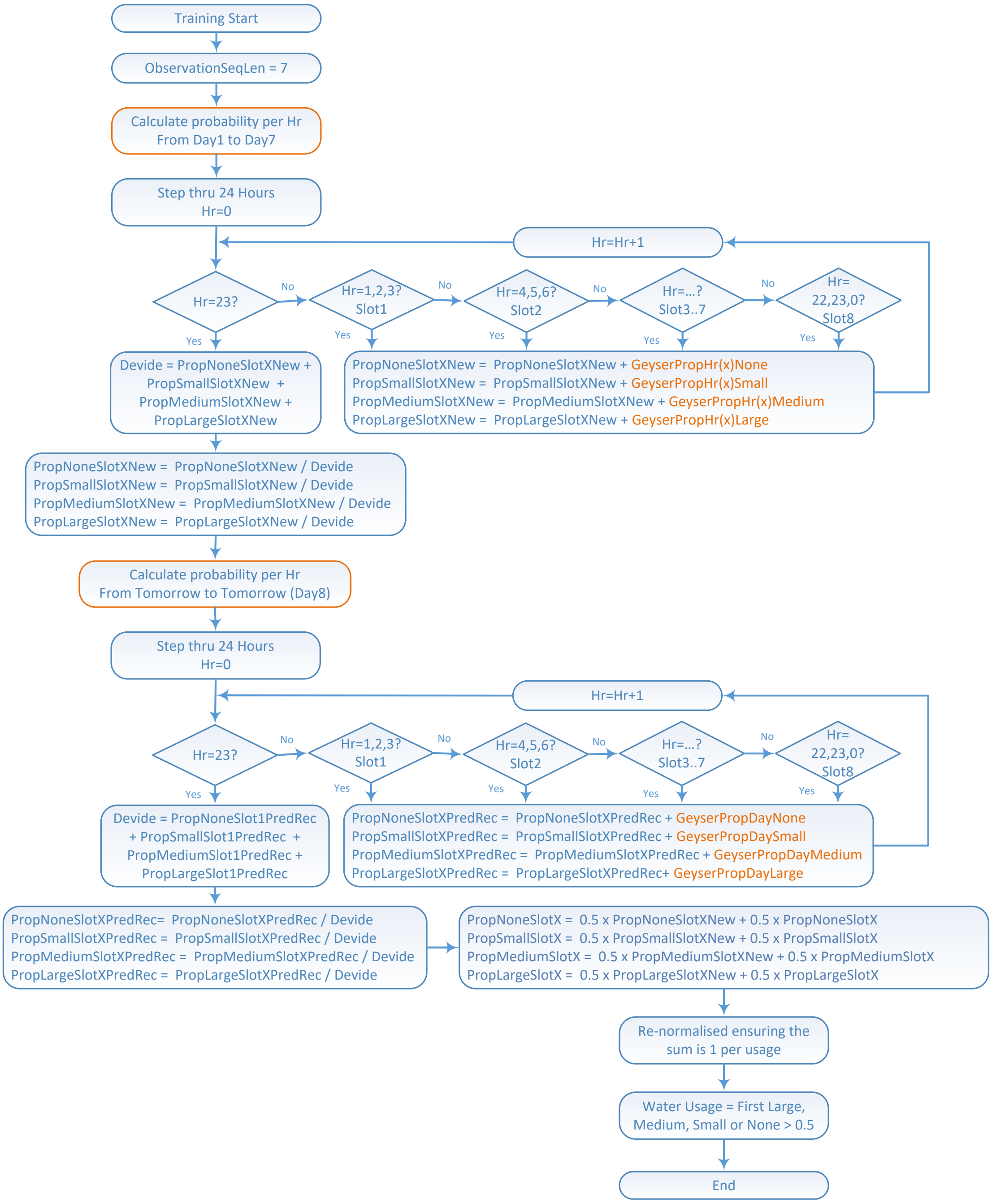


Figure 5.70 Flow chart two predicting hot water usage

Figure 5.71 illustrates the influence on prediction through two factors. First, the percentage historical with new seven days calculated prediction. The x-axis label

1-5% 10-90 means the following: Timeslot 1, highest weight usage 5%, 10% seven day and 90% historical data used. Appendix O, illustrates a partial output for the 50%-50% weight probabilities calculated for 2018.

The different lines are:

C	Correct;
H1	Heat one Usage class higher
H2	Heat two Usage classes higher
H3	Heat three Usage classes higher
I	Water heat to cold

The 1-8 indicate the eight time slots

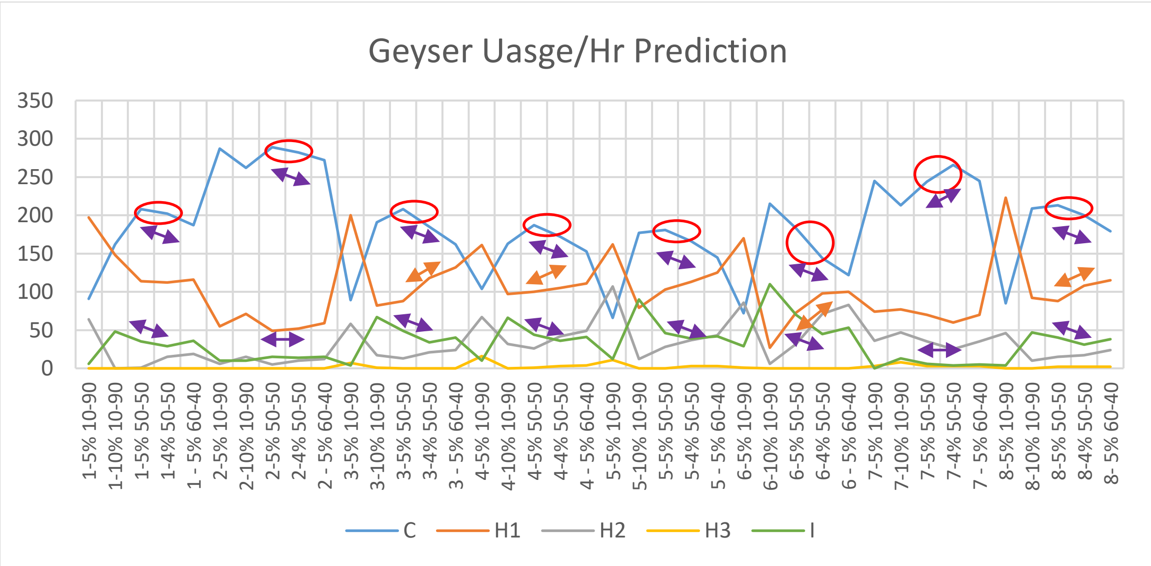


Figure 5.71 Different weights influence hot water predictions

On investigating Figure 5.71 the following was noted:

- Between 10%-90%, 50%-50% and 60%-40% weight that the highest number correct predictions were the sum of 50% historical and 50% previous seven day's prediction as indicated with the red highlights.
- With 50%-50% given the highest correct predictions changing the highest used weight from 5% to 4% the number correct predictions decrease and also the number of incorrect predictions as indicated by the purple arrows. A steep increase in H1 (water heated one class higher than required) orange arrows is noted. A small inline is also noted at H2.
- A higher "highest used weight" will produce better power savings, but lower human satisfaction, where lower "highest used weight" will produce lower power savings with higher human satisfaction.

5.8 *Final geyser solutions*

The final stage is combining the solar geyser heating and hot water usage together, to determine when and how much electrical energy must be added to the geyser water with an electrical standby element. This will be done by implementing the prediction results calculated from section 5.6.4 predicting heating the water with solar power and section 5.7 predicting the hot water usage.

5.8.1 **Simulator parameter**

Two new functions were added to the software. Firstly, the software needs to calculate the duration required to heat the water to a specific temperature and to which temperature the water will cool down over a specific time period.

As discussed in section 4.2, formula 4.20 explains the parameters required heating the water as:

$Q = mc\Delta T$

and formula 4.18 the cooldown parameters as:

$q_{loses} = (T_h - T_{ambient}) / (\Delta x / k + 1/h).$

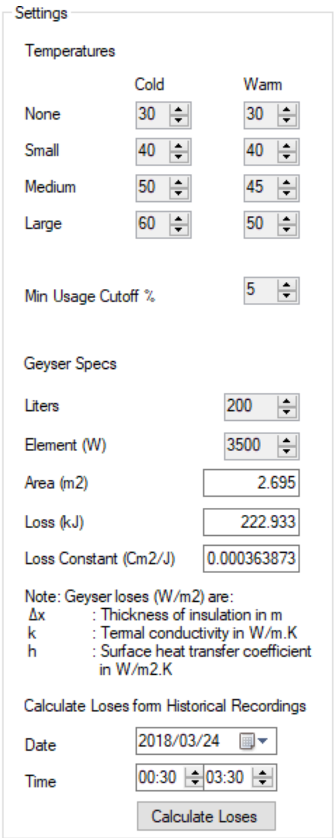


Figure 5.72 Geyser Parameters

Some geyser parameters are difficult to collect as: thickness of insulation, thermal conductivity and surface heat transfer coefficient. A function was added to calculate the loss parameter $\Delta x / k + 1/h$ in Cm^2/J from historical recordings. Investigate the recordings a suitable condition was found between 2018/03/24 0h30 to 3h30 where the ambient temperature dropped from 17.2°C to 17.1°C. All the other parameters are known as illustrated in Figure 5.72 and by calculating the loss parameter as $3.638x10^{-4}$ resulting in 222.9333kJ losses per hour. The same process will be used to simulate the water cool down periods at specific water and roof temperatures.

Simulating geyser heating and cooling will be based on actual recordings except when the cooling required to be calculated when the water where heated, but the simulation calculate no heating required or when the algorithm required heating the water but at that time period the actual element was switched off.

The simulator must ensure that the solar heating does not heat the water higher than 15°C of the roof temperature. The value is based on the average highest medoid indicated in Table 4.17. Roof temperature values lower than the hot water temperatures was excluded and come to 16.48°C and was rounded to 15°C.

Proofing the algorithm efficiency, the simulation will use historical recordings for hot water usage and weather predictions, predicting usage and solar heating. The simulator only has knowledge of historical data as it will have with an actual implementation.

The algorithm can predict if the usage will be none, small, medium or large and if the predicted day will be cool or hot. The end user must decide which minimum temperatures will be suitable for their environment as illustrated in Figure 5.72.

5.8.2 All prediction models together.

The flow charts in Figure 5.74 and Figure 5.75 illustrates the process calculating the geyser element switch at times based on newly develop prediction models. Predictions will be made at the end of a cycle, which is 0h59 each morning.

Firstly, the model will calculate the solar heating probability, as discussed in section 5.6.4 and the water usage per hour, as discussed in section 5.7.

Next, the weather prediction is used determining the minimum geyser temperatures per time slot as illustrated in Figure 5.73. The user can still override the algorithm and disable the geyser element for specific timeslots ensuring the best efficiency.

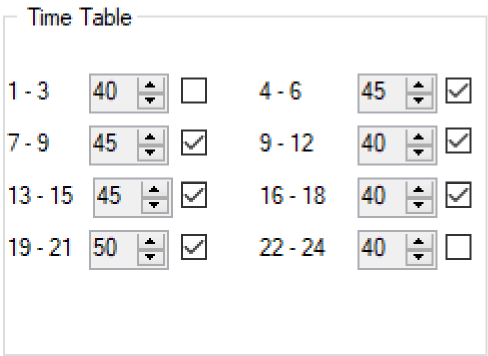


Figure 5.73 Geyser minimum temperatures per timeslot

Next, the simulation algorithm will do calculations every 5 minutes for 24 hours. Calculations in a physical device will be done with any change in hot water pipe or geyser water temperatures. It is important to note that for time periods know where the solar influence is none to minimum solar predictions will be excluded. The time periods were discussed in section 4.4 at Figure 4.23.

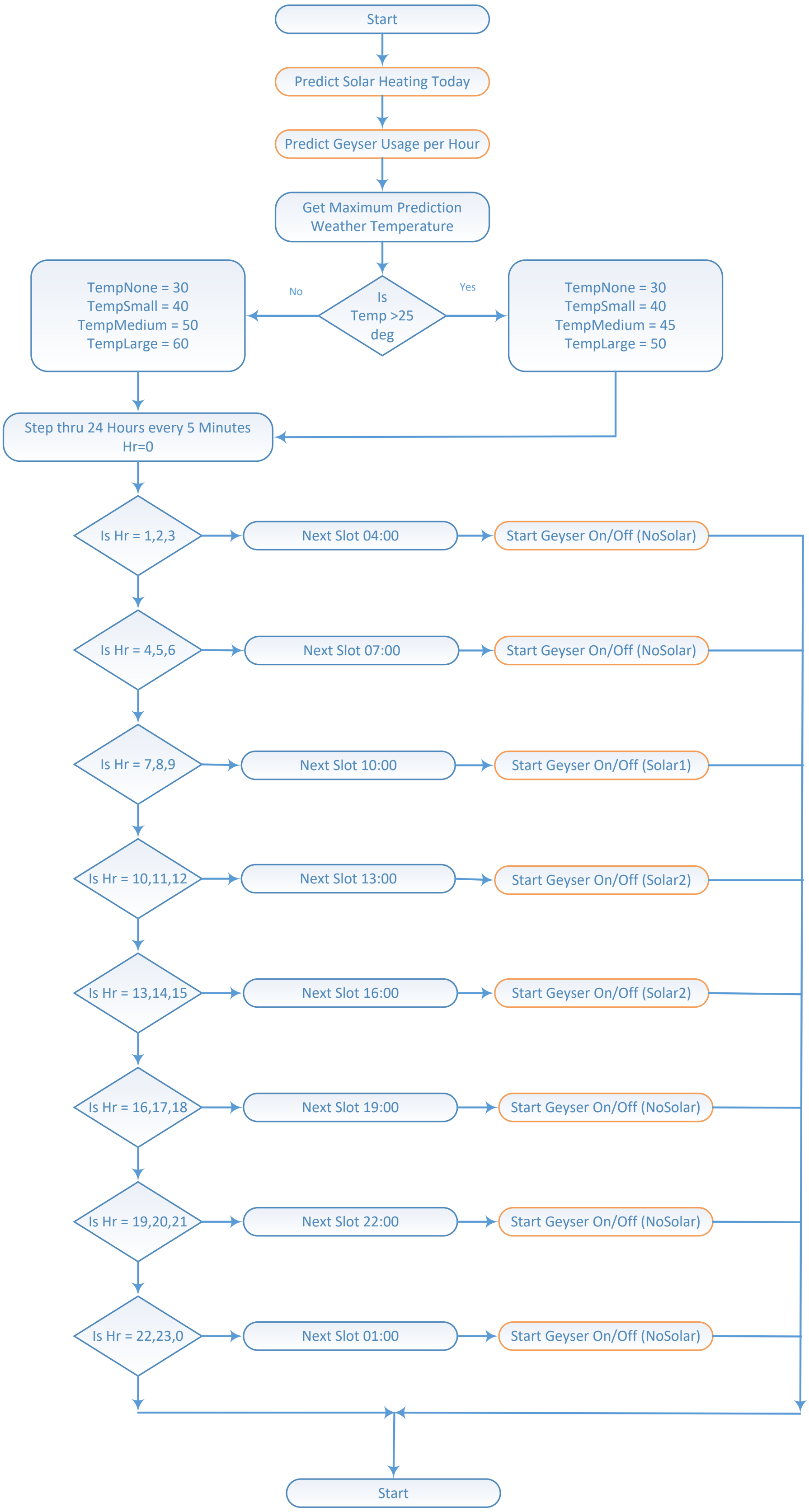


Figure 5.74 Flow chart for final prediction to enable geyser element

The algorithm required knowledge when the next time slot will start. Time based controllers will start heating water at a specific time, but this algorithm already has the knowledge of geyser parameter and will calculate the start time to heat the water, having the correct water temperature entering a specific time slot.

The following is rules in the prediction model: Initial, Transition and emission values:

- Except for time slot four to six, no solar heating will be included determine if geyser element must be enabled.
- For each time slot the current and next time slot minimum temperatures are retrieved as calculated earlier in this section.
- Every five minutes or when temperature changes, the time period left in the time slot and time required heat water from current temperature to next time slot are calculated.

If the geyser water temperature is lower than the required temperature and the water is not busy with cooling down period the element is switched on. The cool down period is enabled when the required water temperature is reached and disabled when water is cooled down for 5°C eliminating geyser switch on/off the entire time if the temperature drops 0.1°C and switch of seconds later when required temperature is reached.

The geyser will also be switched on if the water temperature is correct for the required time slot, but the time left at the current time slot is less or equal than the time needed to heat the water to a minimum temperature for the next time slot.

During time slot four, the next time slot which starts at 10h00, the process changes heating water for the next timeslot. As it is still early in the morning the element will heat up the water except if the solar prediction is High.

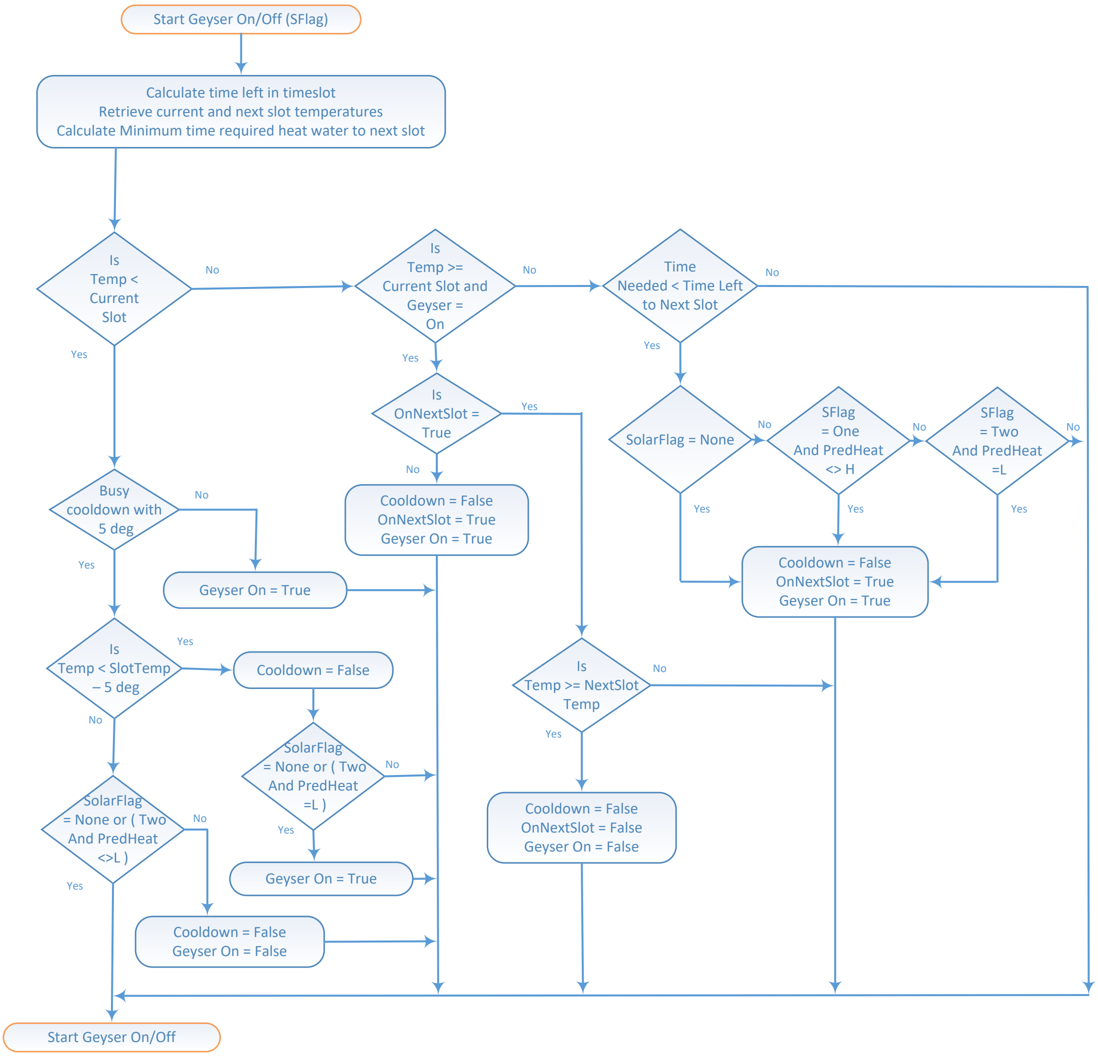


Figure 5.75 Flow chart for final prediction to enable geyser element

During the middle of the day, with is time slot five and six, the element will only be enabled for current and next time slot temperatures if the solar prediction is Low.

5.8.3 Results

The difference between controlling the geyser element with a time-based controller against a controller with solar heating and hot water usage that can adapt to external conditions are shown in this section with the same hot water usage. The simulation was started from January to December 2018 to investigate what the impact would be over the period. Figure 5.76 to 5.80 highlights certain events during this period.

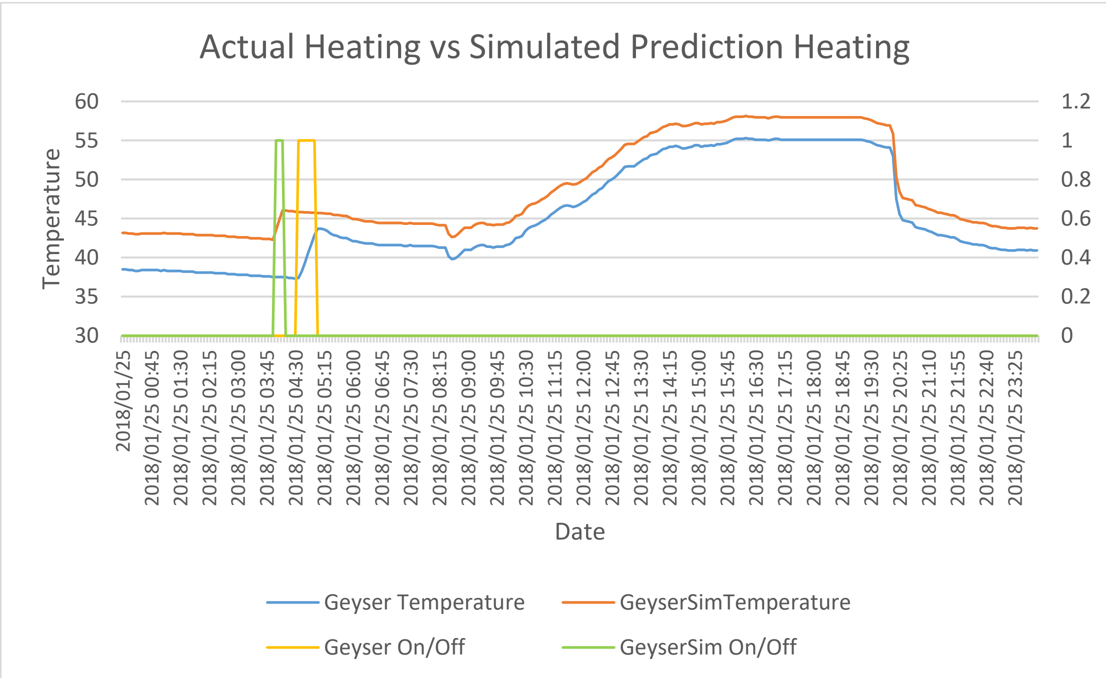


Figure 5.76 Geyser simulation 2018/01/25

Figure 5.76 illustrates the difference in controlling the element on a hot summer's day. The simulation will switch the element on at four and heats the water to 45 °C and the controller at 4h30 to heat water to 43°C. With the hotter water a smaller drop in temperature is noted at 05h00 when there was a small usage of water. This let that the water was hotter when the solar collector start heating the water. This in return, theoretically led to a hotter water temperature before the large usage later in the afternoon. The simulation activates the element for 15 minutes against the actual 30 minutes.

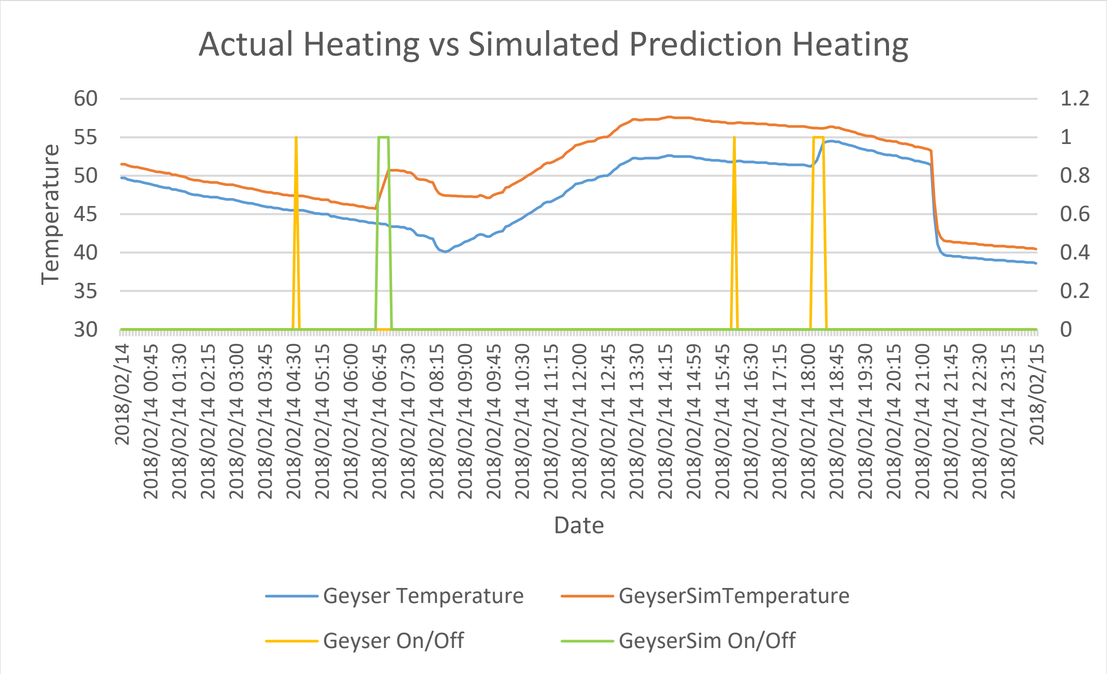


Figure 5.77 *Geyser simulation 2018/02/14*

Figure 5.77 illustrates the comparison for 14 February 2018. At 4h00 the recordings indicate the element was switched on for only 5 minutes. At 6h45 the simulation starts heating the water to 50°C. On investigation, it is noted that the algorithm did predict a large hot water usage between 7h00 and 10h00. At 8h15 a medium usage did occur. Starting with a higher temperature, it is possible for the solar collector to heat the water to a higher temperature, eliminate heating the water later the afternoon with the element for the larger usage. The simulator did activate element for 20 minutes against the actual 30 minutes. The January and February comparisons were made on hot summer days with no clouds, indicate that more power savings are possible and still have hot water the entire day.

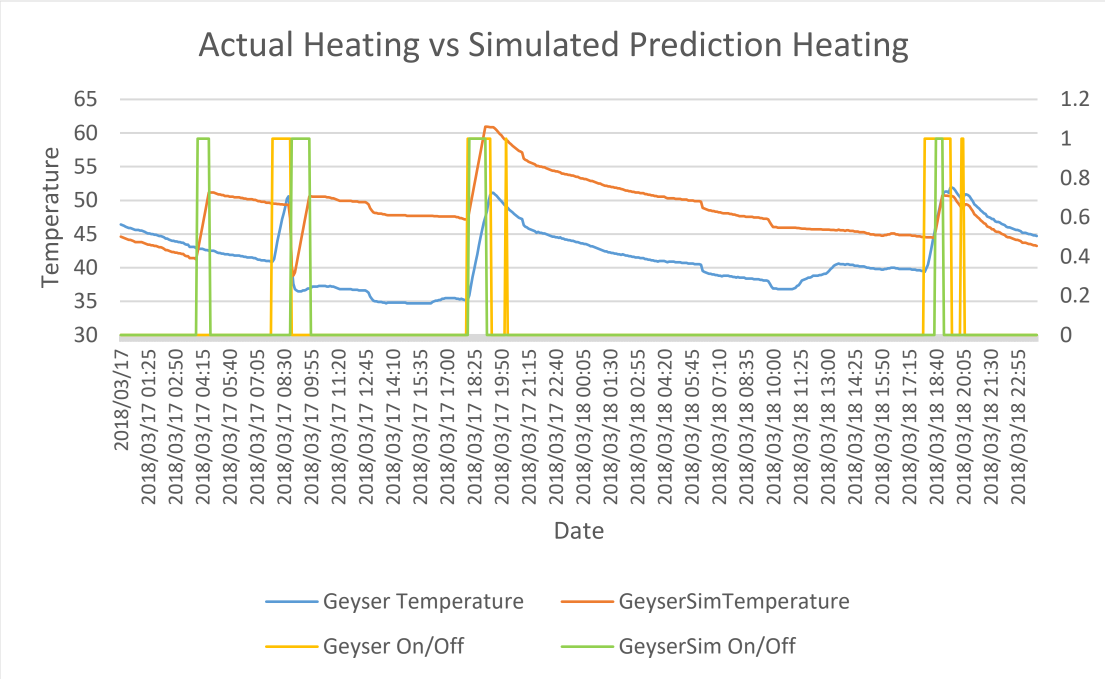


Figure 5.78 *Geyser simulation 2018/03/17*

Figure 5.78 illustrates the comparison over a weekend starting Saturday with cloud cover up to 90% during the solar heating time periods. Investigating the effects over two days illustrates a more accurate representation if solar heating is inefficient. Investigating the 17th, it is shown that the algorithm did predict a medium use on a cool day. At 8h30 there were a large usage and no heating was required before the time. Compare this with the recording it is noticed the water was heated and used just afterwards, indicating that the controller was overwritten as there was no hot water to use. Afterwards, the water was not heated, leaving the household with cold water the entire day. The simulator did have knowledge of the weather prediction and heated to water with the element.

Both controller and simulator incorrectly heated the water in the afternoon for a large usage, as it was already taken earlier in the day. On the Sunday it is noticeable that the recordings shown between 10h00 and 13h00 an increase in temperature indicate it was done through the solar collector as the element was not activated. The simulation did use electrical energy instead of solar energy by heating through the collector, but there was hotter water available during the entire day without the need to heat the water with the element.

Starting at the 17th, the recordings indicate the water is 2 °C higher than the simulation and at midnight, the 18th, the recorded temperature was 1 °C higher than the simulator with the simulator activated the element for 3 hours against the recorded 3.91 hours.

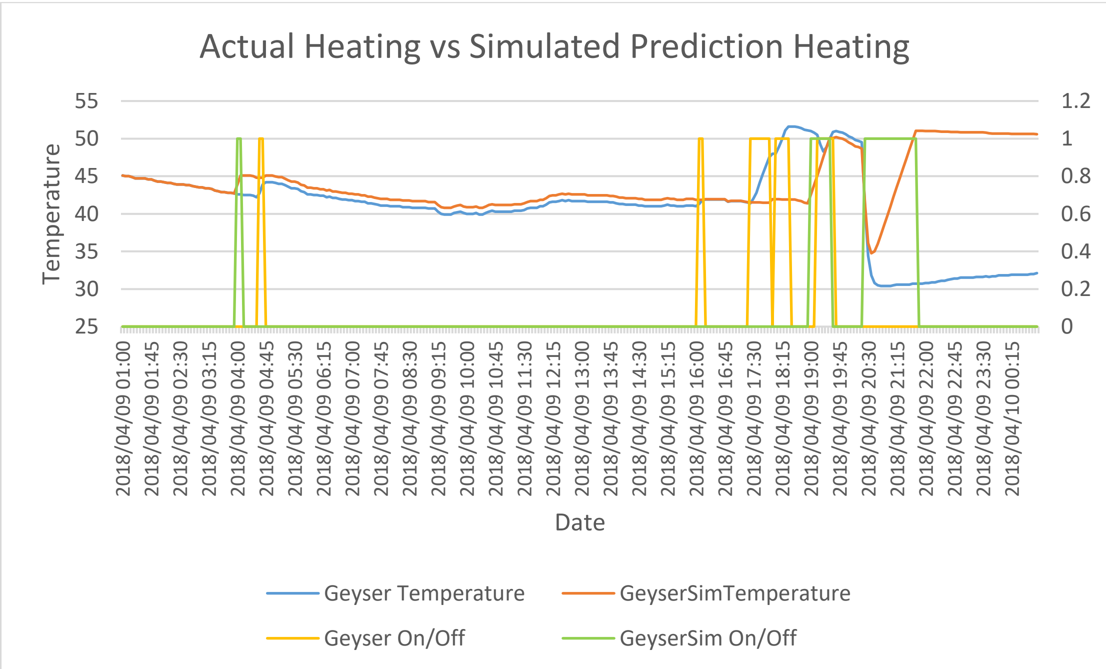


Figure 5.79 Geyser simulation 2018/04/09

Figure 5.79 illustrates another overcast day with only one large usage with insufficient solar heating. The recordings did heat water too early, but the simulator heats the water after the larger usage. It was already being shown that the next morning there will be less power required. The simulator did activate the element for 2 hours an 10 minute, but zero hours the next morning against. The recording illustrates the element was activated for 1 hour and 50 minutes, but 40 minutes the next morning indicate the simulator was activated 20 minutes less.

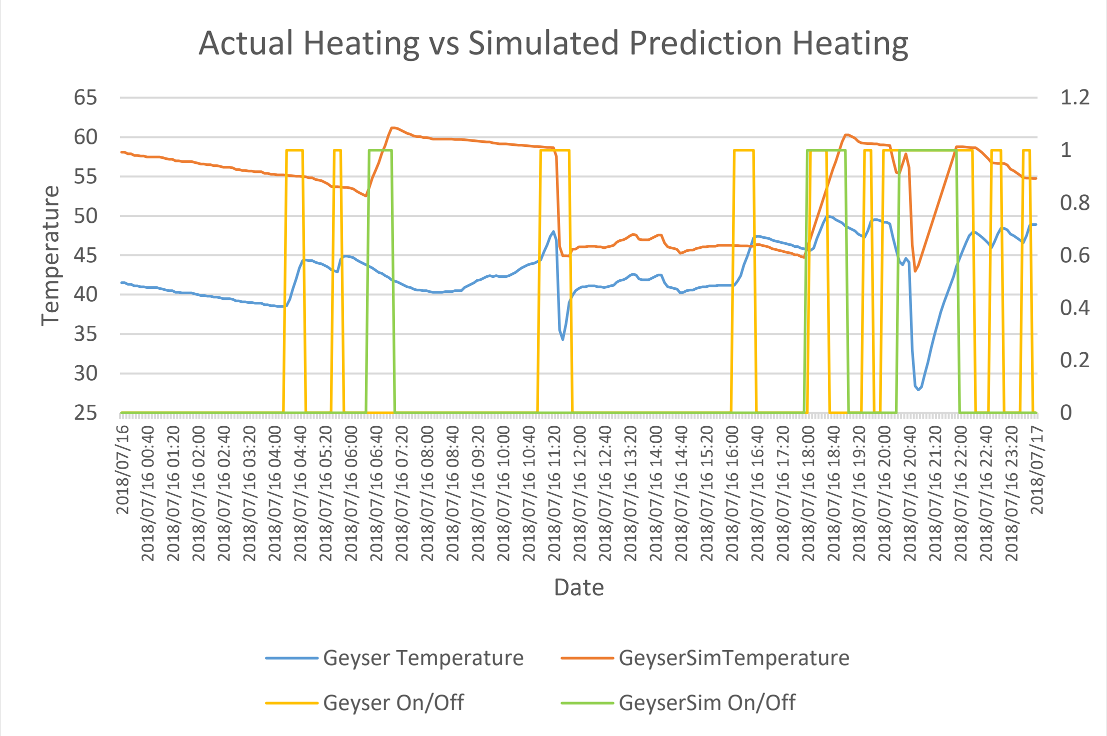


Figure 5.80 Geyser simulation 2018/07/17

Figure 5.80 illustrates the comparison between the simulator and recording for a cold clear winter's day. There were two large usages at 11h20 and again at 20h40. Noted that the simulator did heat the water with higher temperatures as the recordings with the element activated for 2 hours and 10 minutes against the recorded 5 hours and 55 minutes.

Figure 5.81 shown a yearly comparison between the algorithm simulation and the recordings. As expected, both did activate the element more during the winter, with July, the most and the dry summer month December and January the least.

The algorithm did activate the element less for all month except for September and October.

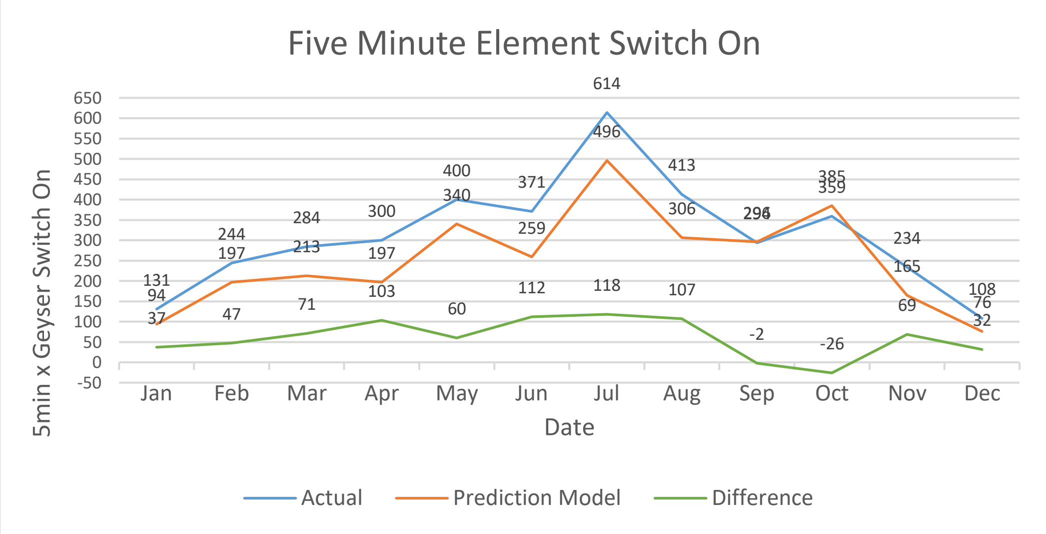


Figure 5.81 *Element Simulator vs Recordings*

The simulator was activated 3024 times, compared to the actual 3752 times. Each active count represents a five-minute interval. That results in 728 times less or a saving of 61 hours per year. As most of the savings in heating water were already accomplished by installing a solar collector and a hot water time-based controller. There is no major saving on the electricity bill, but it is proved that a controller that have knowledge of external conditions can be up to 20% more efficient than a time-based controller with higher hot water temperatures during the day. The biggest advantage is a higher customer satisfaction as there is fewer occasions that the controller must manually activate or re-programmed as seen from recorded data.

All of these results let to a conclusion being discussed in chapter 6.

6 Conclusion

To the development of a prediction model that could control electricity consumption in a traditional house adapting to external environmental conditions and occupation required a platform to test it on, a monitoring system and an evaluation concept. For this reason, a normal household was selected as platform, as the same principles can be implemented in other smart home or office adaptations.

Multiple developed home control systems were investigated, but none had the capabilities to access the required raw data to be monitored. As analysing historical data were critical formulating a new prediction model, a system was built utilizing the ESP8266 range IoT devices.

Controlling household lights with multiple sensors did not make financial sense as the most efficient method is to replace all light fittings with energy saving light bulbs and thus was excluded from the prediction model.

The study was initially conducted on a time-based controlled swimming pool incorporating solar heating. The developed model was then applied to a time-based controlled solar heated geyser environment in evaluation of the adaptability of the prediction model to other smart home or office solutions. With the solar geyser, human behaviour was also incorporated as a criteria as sensor readings change with the withdrawal of hot water from the system,

Contributions made in this study were, but not limited to; a newly develop prediction model which needed a developed process to simplify multiple analogue inputs into linguistic terms. This was done by fuzzification model for all the sensor recordings into a single linguistic term for usage as None, Low, Medium and High.

Addressing event classification correctly, for the model to be able to differentiate between sensor reading fluctuations by an event or environmental conditions. For example, the drop-in temperature during night-time had the same behaviour on the pipe temperature as the use of a small amount of hot water. Membership functions were correctly created to categorise changes in water pipe temperatures as water usage or increase or decrease in ambient temperatures.

Partial partitioning with the CLARANS technique were implemented to analyse the data, creating medoids, simplifying and analysing thousands of recorded entries. This produced a result of a certain pattern associated per season. Implementing yearly or monthly historical data did not indicate accurate prediction probabilities for the newly develop prediction model. Analysing the data, clarified that a combination of all historical data (yearly, monthly, weekly and daily) must be used with a different weight, to increase the prediction model accuracy.

The Hidden Markov model was used to predict solar behaviour but needed to be altered to adapt faster to human behavioural changes. In this study a weather prediction model were used as the known observation and the water heating as the hidden component.

Implementing the standard Baum-Welch algorithm to resolve Hidden Markov model’s training, it was found that if the observation sequence was too long, the probabilities could become zero, resulting in local maxima appearance. This scenario was the opposite outcome as expected from the literature as the Hidden Markov model in other applications the predictions become more accurate with larger observation sequences as with robotics or speech recognition applications. The standard Hidden Markov model with Baum-Welch was found not to be suitable to train the model for sudden environmental and human behavioural changes. If the standard Hidden Markov model was used multiple HMM models where required as each season observation was different and did decrease the accuracy of one global model. Another shortcoming encountered with the standard Baum-Welch backwards algorithm was that it did not include the current training forward (α) calculations. This was another factor that made the standard HMM slow in adopting in sudden environmental or human behavioural changes.

Thru inspection it was notice that by altering the model not to allow any normalised value to be a 0 or 1 the HMM did not get stuck into local maxima appearance. Combining historic and seven days historical recordings, with a weight of 90% all historic recordings and 10% past seven days, made the model sensitive to sudden changes in weather conditions without the need to develop a new training model. The modified Baum-Welch algorithm resulted in the use of only one HMM model with no season specific knowledge with only five incorrect Hidden Markov predictions for the year 2019.

The newly created prediction model do allow for specific factors as requirements differs per solution. For example, if wind is present, it will speed up the drop in temperature and subsequently the pool pump should be deactivated. The user has the ability to enable the pool pump when the wind speed is higher than the set value, to enable the automated pool cleaner. The results did indicate that installing an intelligent pool controller use less electricity, without compromising the pool water temperature and filtration of the pool water.

The same process was implemented to predict the geyser solar efficiency. A problem was that the historical maximum water temperature could not be brought into account, as the temperature that was reached by solar or electrical power was unknown to the system. The maximum roof temperature was rather incorporated, as indication level of solar geyser heating.

The Viterbi algorithm was used to predict both, pool and geyser heating probabilities. The observation sequence consists of the previous four days, current and trained condition, calculated by the modified Baum-Welch algorithm.

Simulations on the overall system did indicate that the geyser controller was up to 20% more efficient than a time-based controller. However, though savings seemed minimal the biggest advantage was more reliable hot water when needed. The achievement was that solar heating and a proper type of controller for the standby element does create the most electrical power savings.

New contributions made in this study by implementing the developed prediction model are:

- Converting sensor readings into linguistic terms simplified the prediction model and decrease calculation time required,
- Using another model prediction as the observations for Hidden Markov Model,

- Altering Baum-Welch algorithm as Markov Hidden model training algorithm by decreasing the length of the observations is making the predictions more accurate thru adding different weight to different age emission and observation data for prediction models where sudden behavioural changes can occur, and
- Proving that an artificial algorithm can be implemented in a smart home or office environment with the same comfort levels, but also reducing the electrical power usage.

References

1. Martinez-Miranda J, Alvarado A and Banares-Alcantara R, **Modelling human behaviour at work using fuzzy logic: The Challenges of Work Teams Configuration**, Journal of Organizational Behaviour, January 2014.
2. Rea MS, Figueiro MG. and Bullough JD, **Circadian light,** Journal of Circadian Rhythms, Volume 8, Ubiquity Press, 13 February 2010.
3. Huang Q, Cohen D and Komarzynski S, **Hidden Markov models for monitoring circadian rhythmicity in telemetric activity data**, Journal of the Royal Society February 2018 Volume 15, Issue 139.
4. Otto SP and Day T, A biologist guide to mathematical modelling in ecology and evolution. Princeton Predd. New Jersey, 2007.
5. Galindo J, **Fuzzy Databases: Modelling, Design and Implementation**, London, Idea Group Publishing, 2006.
6. Chen G and Pham TT, **Introduction to Fuzzy Sets, Fuzzy Logic and Fuzzy Control Systems**, Houston, CRC Press, 2001.
7. Lee KH, **First Course on Fuzzy Theory and Applications**, New York, Springer, 2005.
8. Voskoglou MG, **Methods for Assessing Human-Machine Performance under Fuzzy Conditions**, Journal of MDPI Mathematics 2019 volume 7.
9. Maimon O, **Clustering Methods**, Tel-Aviv, Tel-Aviv University,1998.
10. Jain AK and Dubes RC, **Algorithms for Clustering Data**, New Jersey, Prentice-Hall, 1988.
11. Tan P-N, Steinback M and Karpatne A, **Introduction to Data Mining 2nd edition**, Pearson, 4 January 2018.
12. Han J, Kamber M and Pei J, **Data Mining Concepts and Techniques 3rd edition**, AmsterSwarndeeo Saketdam, Elsevier, 2012.

13. Swarndeep SJ and l Pandya S, **An Overview of Partitioning Algorithms in Clustering Techniques**, International Journal of Advanced Research in Computer Engineering & Technology June 2016, Vol 5, Issue 6.

14. Downey AB, **Think Bayes**, Green Tea Press, Needham, 2012.

15. Schliep A, **A Bayesian approach to learning Hidden Markov model topology with biological sequence analysis**, Thesis for Faculty of mathematics and science at University of Cologne, Düsseldorf, 2002.

16. Cappé A, Moulines E and Rydén T, **Inference in Hidden Markov Models**, Berlin, Springer, 2007.

17. Rezaei V, Pezeshk H and Perez-Sa'nchez H, **Generalized Baum-Welch Algorithm Based on the Similarity between Sequences**, PLOS one December 2013, Vol. 8 Issue 12.

18. Rabiner LR, **A tutorial on Hidden Markov Models and selected applications in speech recognition,** Proceedings of the IEEE, Vol. 77 No. 2, February 1989.

19. Liu T and Lemeire J, **Efficient and Effective Learning of HMMs Based on Identification of Hidden States**, Hindawi: Research Article at Mathematical Problems in Engineering, 2017.

20. Rabiner LR and Juang BH, **An introduction to Hidden Markov Models**, IEEE ASSP Magazine, January 1986.

21. Maas A, **Spoken Language Processing**, Stanford University, 2017.

22. Pietrzykowski M and Salabun M, **Applications of Hidden Markov Model: state-of-the-art**, Wojciech Salabun et al, Int. J. Computer Technology & Application 2014, Vol. 5 No. 4.

23. Soiman SI, Rusu A an Pentiuc SG, **Optimizing the Forward Algorithm for Hidden Markov Model on IBM Roadrunner clusters**, Advances in Electrical and Computer Engineering 2015, Vol 12 No.2.

24. Pullford GW, **Multihypthesis Viterbi Data Association**: Algorithm Development and Assessment, IEEE Transactions on Aerospace and Electronic Systems May 2010, Vol. 46 No. 2.

25. Levy R, **HMM Viberti mini example**, Winter 2009

26. **Speech Recognition and HMM Learning** [online], Available from https:/slideplayer.co/slide/3359798, [Access 1 April 2019].

27. Dintchev OD, **Solar water heating as a mechanism to reduce domestic electrical energy consumption and its contribution towards south African maximum demand**, Proceedings of the 15th domestic use of electrical energy conference, Cape Town, 2006.

28. Sauer FJ, **Predicting solar radiation based on available weather indications**, Thesis for Missouri university of science and technology, Scholars' Mine, 2013.

29. Delport J, **The influence of temperature on residential hot water load control**, Proceedings of the 15th domestic use of electrical energy conference, Cape Town, 2006.

30. Great Britain Health and Safety Executive, **Legionnaires' disease Fourth edition**, London, 2013.

31. World Health Organisation, **Legionella and the prevention of legionellosis**, Geneva, WHO Press, 2007.

32. Delport D, **Achieving energy efficiency for hot water cylinders (geysers) with the new dual element system**, Proceedings of the 15th domestic use of electrical energy conference, Cape Town, 2006.

33. Thomas RSM, Xia X and Zhang JF, **Energy Efficient Geyser**, Proceedings of the 17th domestic use of electrical energy conference, Cape Town, 2008.

34. Tyler L, **Power utility urges pool pump enrgy saving plans**, Engeneering News, Creamer Media, Johannesburg, 24 April 2007.

35. Tarad AH, **Heating Mechanism and Energy Analuses for Over-Ground Outdoor Swimming Pool Technology**, Asian Journal of Applied Science and Technology, Volume1, Issue 6, Pages 8-22, July 2017.

36. Shorr G, **Do motion sensor light switches save energy** [online], Available from https:/www.saveonenergy.com/energy-saving-tips/do-motion-sensor-light-switch-save-energy, 8 Jan 2014, [Access 21 July 2018].

37. Hrasnica H and Lehnert R, **Broadband powerline communications**, Chichester, John Wiley & Sons Ltd, 2004.

38. Palmer M, **Hands-On Networking Fundamentals Second Edition**, Canada, 2013

39. Tooley M and Tooley R, **Exploring the Arduino Part 12: Wi-Fi and the Internet of Things**, Everyday Practical Electronics, January 2017, Wimborne Publishing.

40. **Fuzzylite** [online], Available from https://Fuzzylite.com, [Access 19 May 2017].

41. Shaily Thaker and Viral Nagori, **Analysis of Fuzzification Process in Fuzzy Expert Systems**, Proceedings of International Conference on Computer Intelligence and Data Science, 2018.

42. Delport J, **The geyser gadgets that work / do not work**, Proceedings of the 14th domestic use of electrical energy conference, Cape Town, 2005.

43. Kwiklot, **Electric water heaters domestic range**, Benoni, 2015.

44. Melikyan Z andEgnatosyan S, **Developing of New Structure of Flat Plate Solar Water and Method for Calculation and Design**, International Journal of Energy and Power Engineering, July 2016, Vol. 5 No.4..

45. Li C, Biswas G, Dale M, and Dale P, **Building models of ecological dynamics using hmm based temporal data clustering,** In procedding. of the Fourth International Conference on Intelligent Data Analysis, 2001.

46. Reynolds A.P, Richards G, de la Iglesias B and Rayward-Smith V. J. **Clustering Rules: A Comparison of Partitioning and Hierarchical Clustering Algorithm**. Journal of Mathematical Modelling and Algorithms 2006, Volume 5.
47. Eye Witness News [online], Availible from https://ewn.co.za/2019/10/23/heat-is-on-gauteng-residents-warned-to-expect-weekend-heatwave, [Access 24 Nov 2019]
48. Sachs P, **Wind forces in engineering 2nd edition**, Oxdord, Pergamon Press, 1978.
49. Cilla R, Paricio MA and Berlanga A, **Recogmizing Human Activities from Sensors Using Hidden Markov Models Constructed by Feature Selection Techniques**, Journal of algorithms 2009 Volume 2.
50. **ESP Basic** [online], Available from www.esp8266basic.com, [Access 3 June 2016].

Appendix A: Flashing Node MCU with ESP8266 Basic

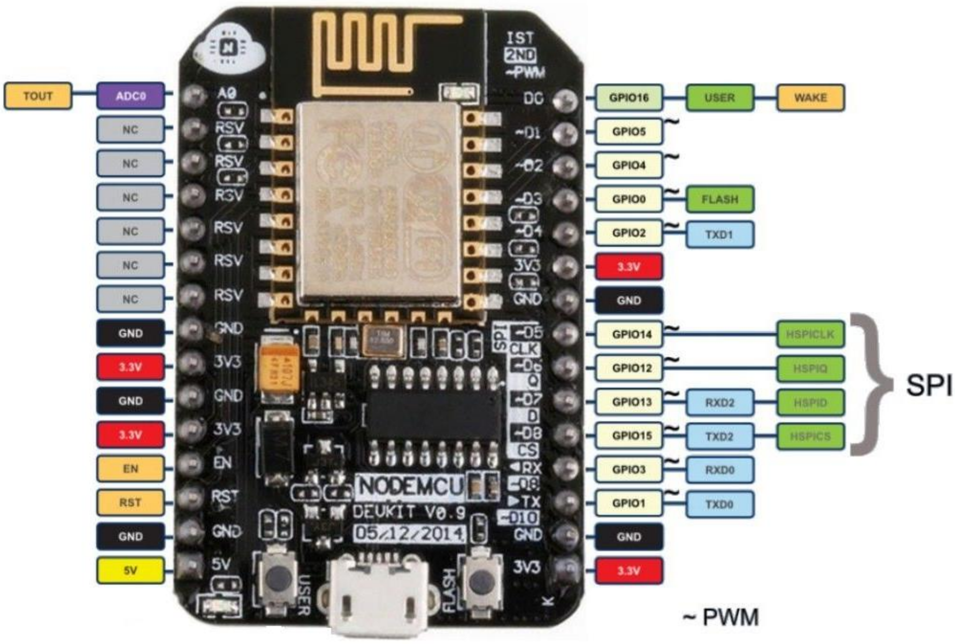


Figure A.1 NodeMCU ESP8266

By default, die NodeMCU ESP8266 as illustrated in Figure A.1 is loaded with Lua scripting language. An on-board USB to TTL CP2102 chip allows for serial communications to a computer needed to flash the ESP8266 module. Version 2.0 build 24 was used.

The following files is required to flash the module;

- CP2102 Driver: CP210xVCPInstaller_x64.exe from www.silabs.com
- ESP8266 Basic flasher: ESP_Basic_Flasher.exe from www.esp8266basic.com

After installing both executables, the ESP module needs to be connected via the USB port. Open Device manager under Ports (COM & LPT) and look for the Silcon Labs CP210x entry for the COM port number. Figure A.2 indicate that port 10 was used.

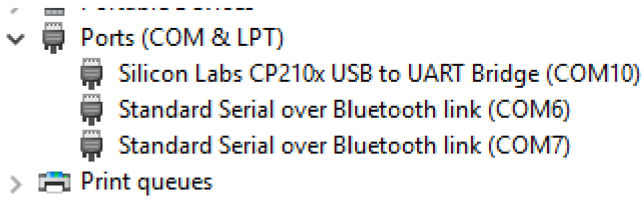


Figure A.2 Device Manager Ports

Run the Basic flashing software and select the Communication Port as 10 and the flash size as 4Meg as shown in Figure A.3. The next step is to place the ESP8266-12E in bootloader mode by tie GPIO0 and GPIO15 to ground before powering up the module. The on-board blue light emitting diode (LED) will flash once indicating that the chip is in bootloader mode and GPIO0 and GPIO15 can now be in any state. The NodeMCU board make this much easier. If the device is powered on, you need to press the reset and flash button together, then only release the reset button after the blue on board LED flashed. Now click the Firmware Flash button on the NodeMCU.

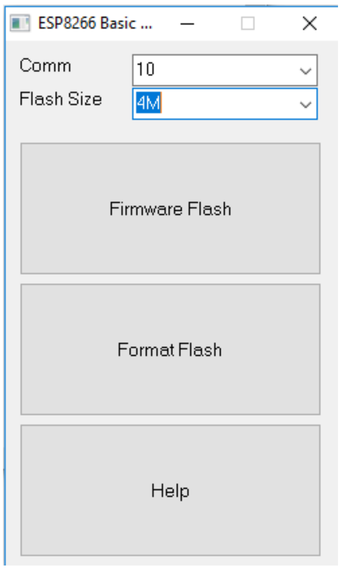


Figure A.3 ESP8266 Basic flasher

The status window will indicate the flashing progress as shown in Figure A.4 During the flashing period, the blue on board led will flash until it is completed in about

30 seconds a pop-up window will inform that the flashing of the ESP8266 was successful.

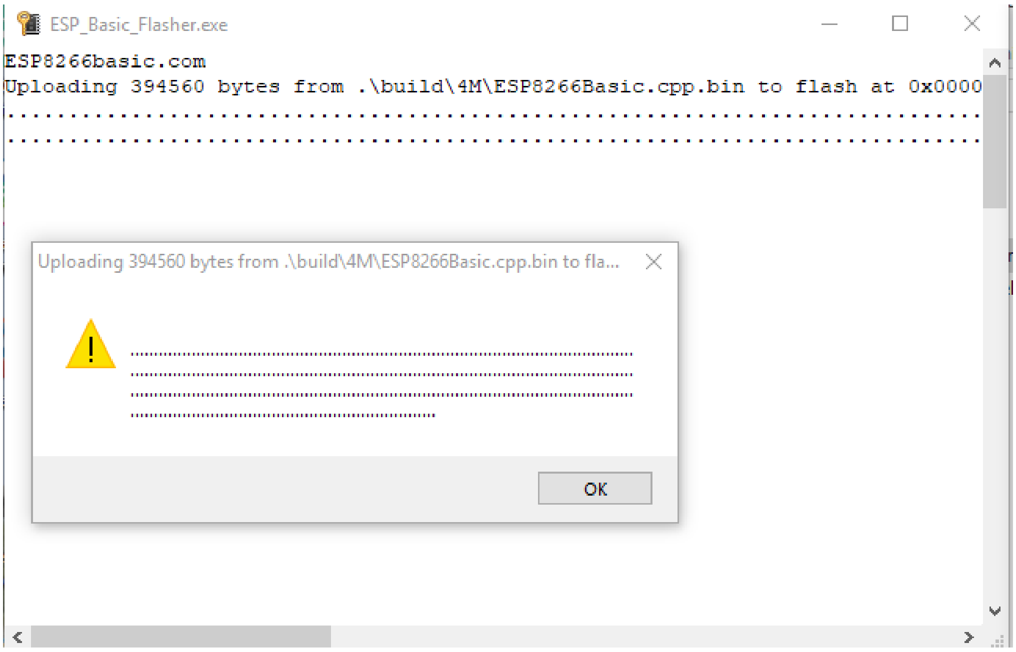


Figure A.4 *ESP8266 Basic flashing status*

ESP8266 Basic boots the Wi-Fi module in AP mode with a SSID of ESP. The example that was used, the SSID "ESP1" was already setup. The final step is to connect a computer to the SSID and to open an internet browser and go to http://192.168.4.1 as illustrated in Figure A.5. Connecting a PC to the ESP8266 via a USB cable, the boot sequence can be viewed on a terminal eliminator like putty as illustrated in Figure A.6. [50].

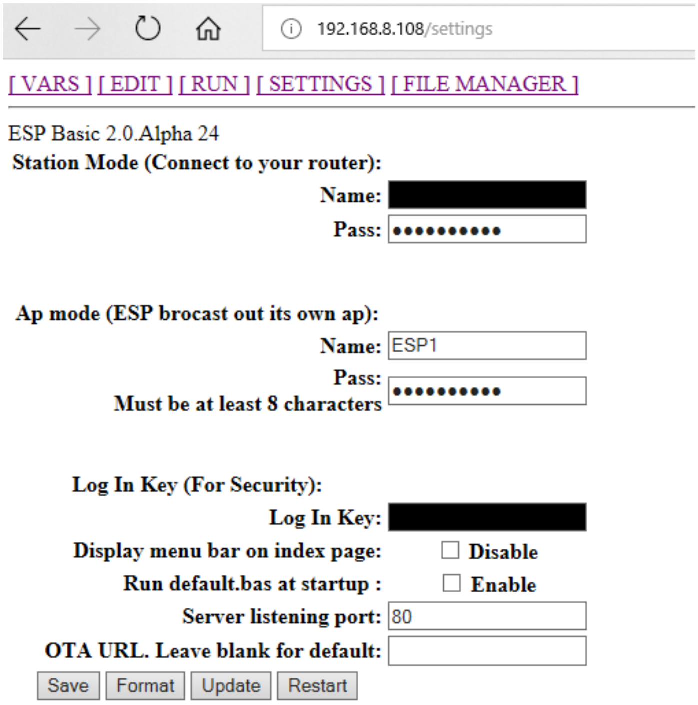


Figure A.5 *ESP8266 Basic development window*

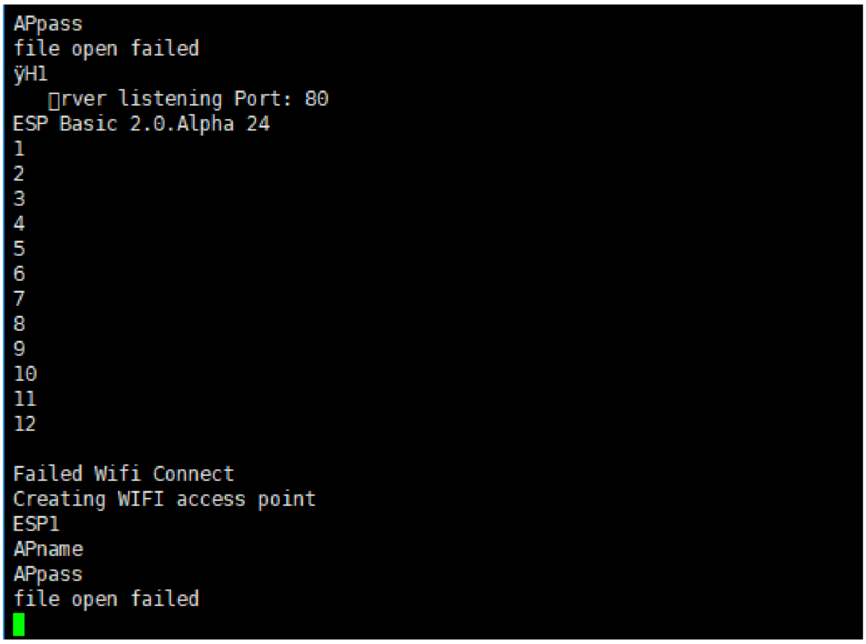


Figure A.6 *ESP8266 Basic boot sequence via USB*

Appendix B: IoT Hardware design

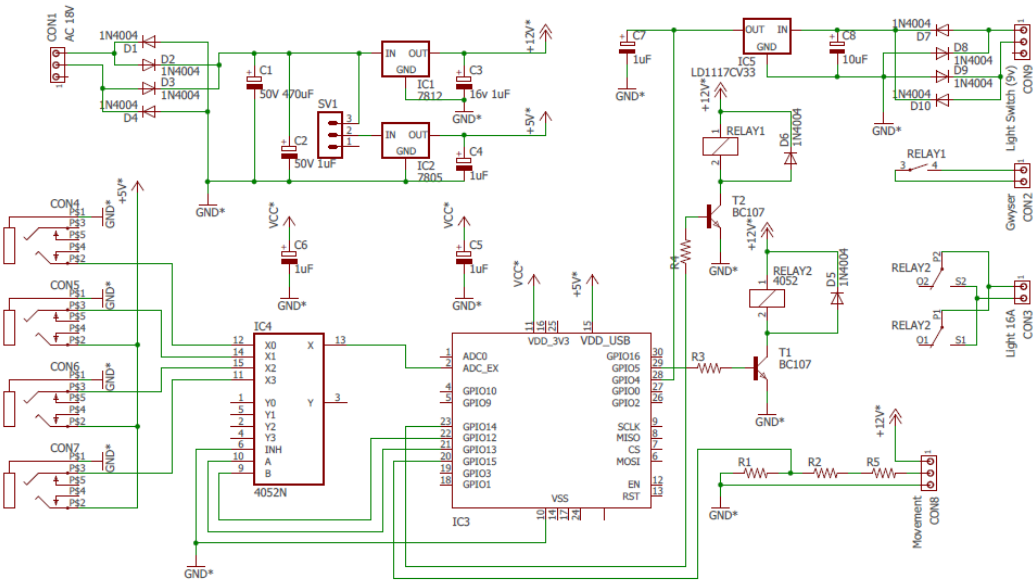


Figure B.1 *Sensor Schmatic*

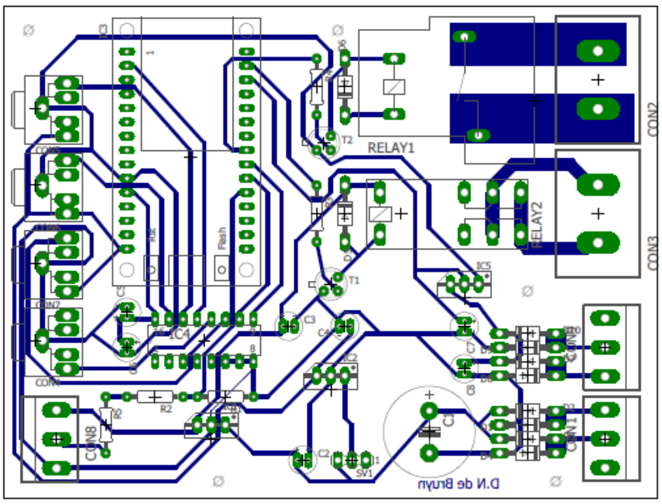


Figure B.2 *Sensor PCB*

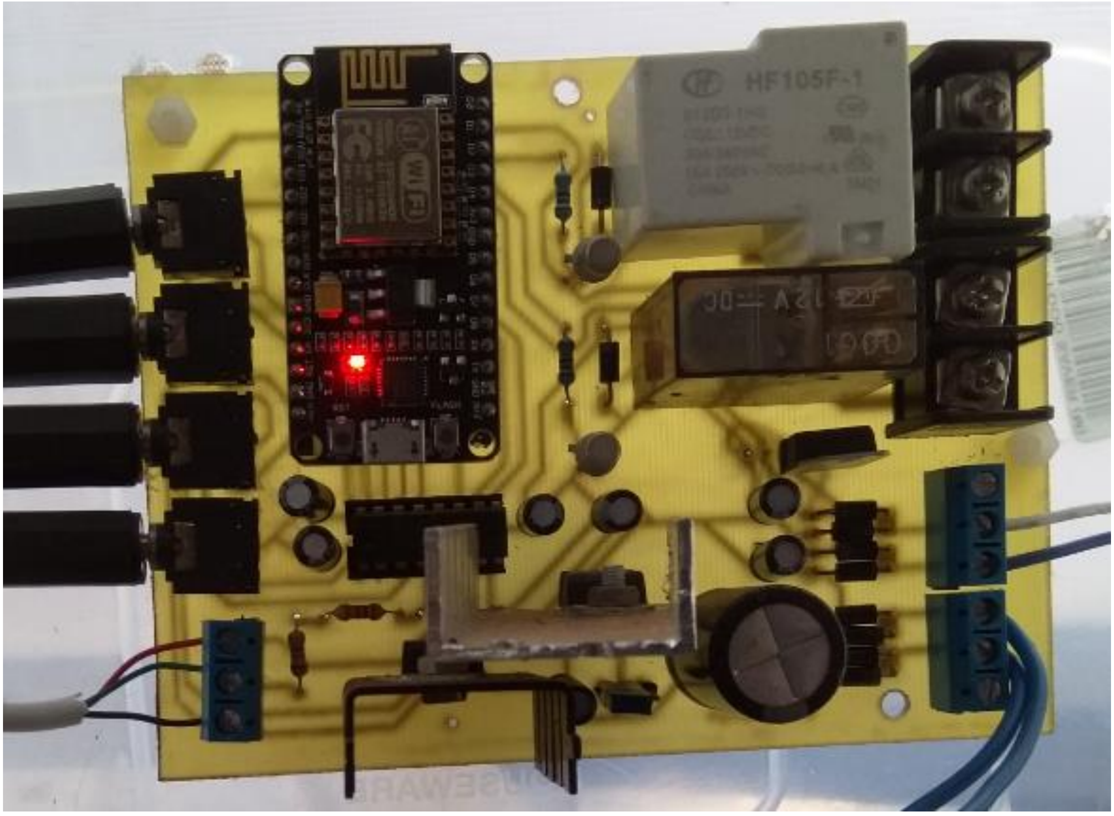


Figure B.3 Physical IoT device

Figure B.4 IoT Current Sensor

Figure B.5 IoT Temp Sensor

Appendix C: Registering sensors to the controller and update endpoints

The device must be flexible and the GPIO pin's including its feature must be registered as an endpoint when the sensor registers with the controller.

Table C.1 Endpoint Types

Description	Type ID	Direction	AnaDig	Group
TempGeyserInside	1	I	A	Geyser
TempGeyserOutside	2	I	A	Geyser
TempGeyserPipe	3	I	A	Geyser
TempPoolInside	4	I	A	Pool
TempPoolHeat	5	I	A	Pool
TempInside	7	I	A	Enviroment
TempOutside	8	I	A	Enviroment
Humidity	9	I	A	Enviroment
DayNight	15	I	A	Enviroment
SwitchLight	20	I	D	Light
SwitchMovement	21	I	D	Movement
SwitchPlug	22	I	D	Plug
CurrentGeyser	30	I	A	Geyser
CurrentPump	31	I	A	Pool
RelayGeyser	100	O	A	Geyser
RelayLight	101	O	D	Light
RelayPlug	102	O	D	Plug
RelayPool	110	O	D	Pool

Table C.1 indicate the controllers Endpoint Types characteristics. This scalable design enables the controller to add any future type sensor.

When the ESP8266 registers, it requires a unique identification. ESP8266 Basic can read the module's unique serial which was used to identify the unit. The designed communication between the controller and sensors uses UDP/IP. The device ID, amount of IO's, name and IP address are registered into the Device table as indicate Figure C.1. The IP does not need to be included into the command as it can be retrieved from the UDP/IP header. The table will also keep track if the device is reachable and the number of Endpoints registered.

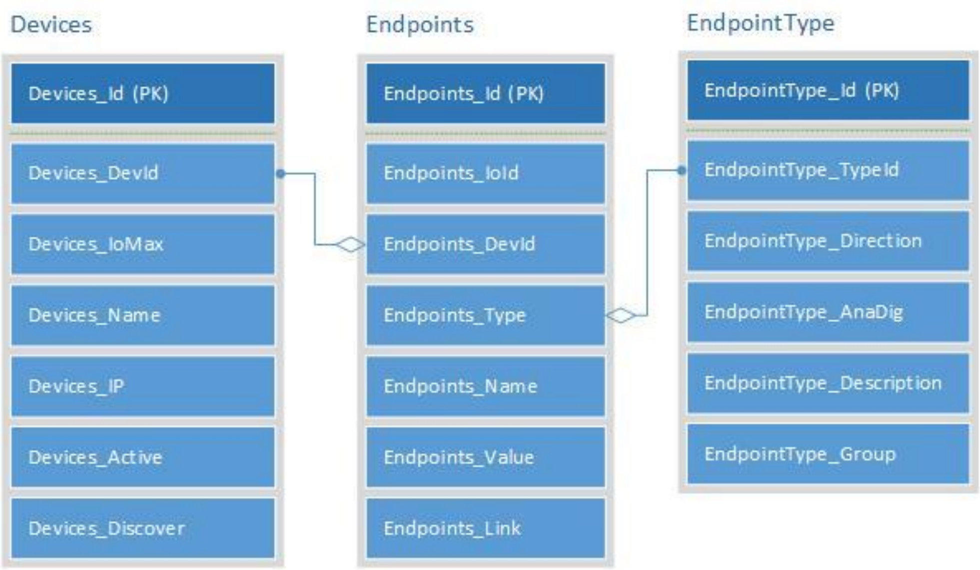


Figure C.1 SQL Tables for Device and Endpoint registration

After a device registers, an acknowledge is sent from the controller. The controller requests, every third second, the information of an endpoint. The endpoint ID, with the devices ID, endpoint type and endpoint name are stored in the Endpoints table until all the endpoints is discovered. The value of the endpoint and linked controller feature are also stored as discussed in section 3.2. The endpoint type features shown in Table C.1 are stored into the EndpointType table as illustrated in Figure C.1.

To ensure that the device is available a keep alive is required. To reduce costs of the sensor, a Real Time Clock (RTC) was not added to the sensor, but to ensure that the time is in synchronisation with the controller the date and time was included into the keep alive command as shown in Table C.1.

Three keep alive requests, will be send every minute, in three second intervals between then 10s and 20s keep alive timer interval, until an acknowledge is received. If a keep alive is not received after three requests, it is marked as not live but if an acknowledge is received the counter is reset to 60 seconds. To ensure that every device timer is not identical the number of device in the database are added to the keep alive counter. The registration process flowchart is illustrated in Figure C.2.

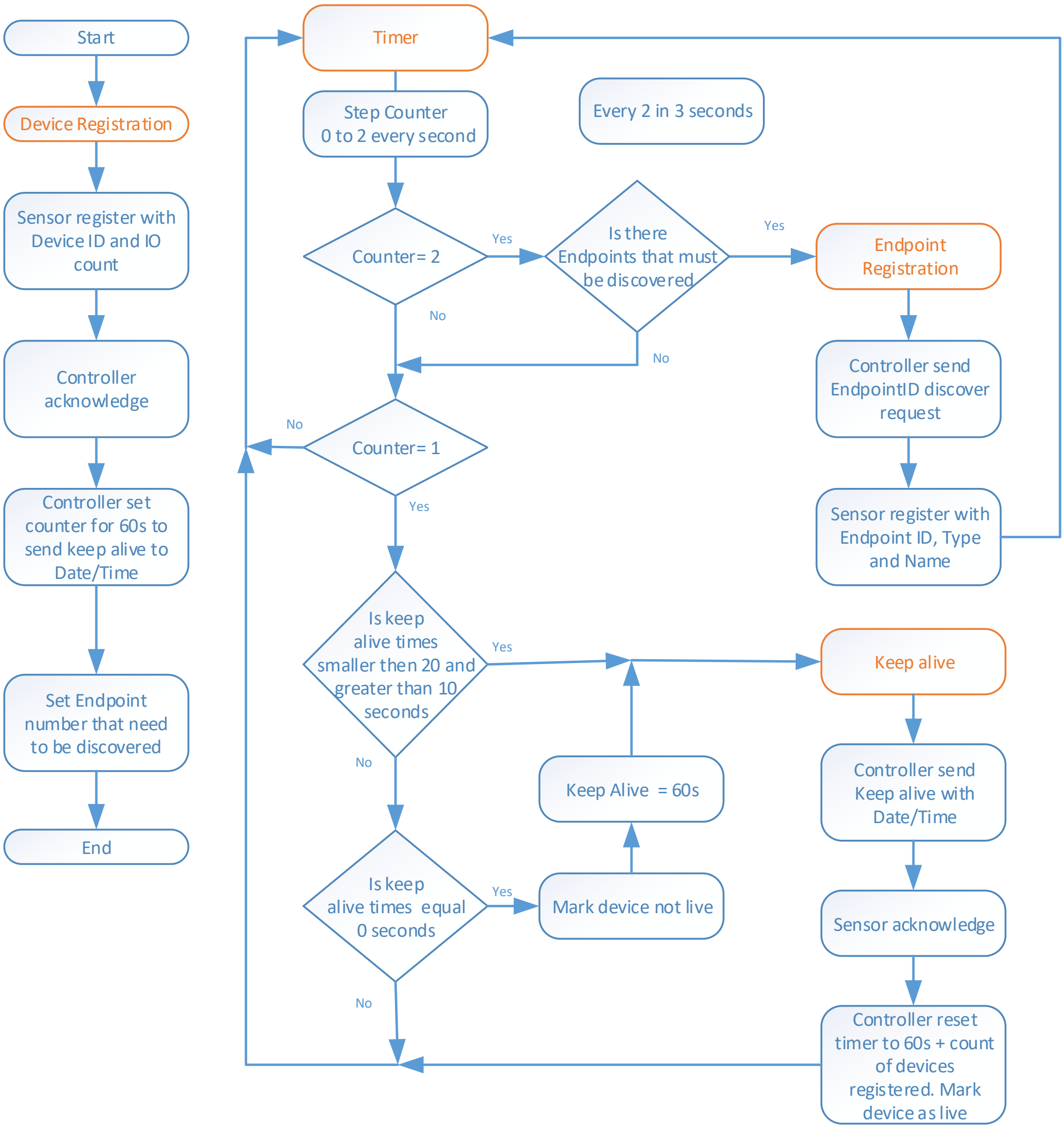


Figure C.2 Flowchart for Device and Endpoint registration

After the registration process, the controller will request the sensor to update its endpoint values every 60 seconds. If any of the endpoint values changes on a sensor it will also send an update of all the values. When the controller changes a value on an endpoint, the sensor will update the controller with all the endpoint values.

Table C.2 illustrates all the communication commands that are used between the controller and the sensors.

Table C.2 Commands between Controller and Sensors

Description	Command	ID	Value							
Registration										
Request Server Connect Device	RSCnDev	xxxxx	Name	Maximum Endpoints						
Acknowledge Client Connect Device	ACCnDev	xxxxx								
Request Client Connect IO	RCCnIO	xxxxx	IOId							
Acknowlege Server Connect IO	ASCnIO	xxxxx	IOId	Type	Name					
Keep Alive										
Request Client Write DateTime	RCWrDT	xxxxx	Yr	Mnd	Day	Hr	Min	Sec		
Acknowledge Server Write DateTime	ASWrDT	xxxxx								
Update Endpoint values										
Request Client Read IO	RCRdIO	xxxxx								
Acknowledge Server Read IO	ASRdIO	xxxxx	1	2	3	4	5	6	7	8
Request Client Write IO	RCWrIO	xxxxx	IOId	Value						

Figure C.3 is graphical illustration of the communication between the sensors and controller. This was captured at the controller and is displayed in Appendix E. The following occurs at step:

1. Sensor device with ID 8851570, request to registers with controller indicating the Name is Geyser & Bedroom and have 8 endpoint sensors. The device name, Controller IP address and the request is set via the IoT GUI interface.
2. The controller acknowledges the request.
3. The controller will now start to request the sensor device to register its first endpoint.
4. The device acknowledges indicating that the first sensor endpoint type is 1, with name Geyser Temp. The controller knows that Type 1 is TempGeyserInside with an Analogue Input and belongs to the Geyser Suite.

5. To Step 18. Endpoint 2 to 8 are registered in the same manner as endpoint 1.
19. After all the endpoints are successfully registered. The controller will send a keep alive to the device including the current date and time.
20. The device acknowledges the keep alive.
21. The controller requests the device an update of all the endpoint values.
22. The device acknowledges including all eight endpoint values, starting with the first sensor.
23. The geyser temperature changes from 32.5°C to 32.6 °C and it send an update to the controller in the same format as in step 22. It does not require an acknowledgement from the controller. Step 22 will be repeated every minute and that with the keep alive is believe is enough to ensure that their us reliable communication between the devices.
24. The bedroom light is switch on from the controller. The controller sends a request to the controller to switch the bedroom light on.
25. The controller acknowledges in the same format as in step 22 indicating that the sixth endpoint, the bedroom light as been switch on.

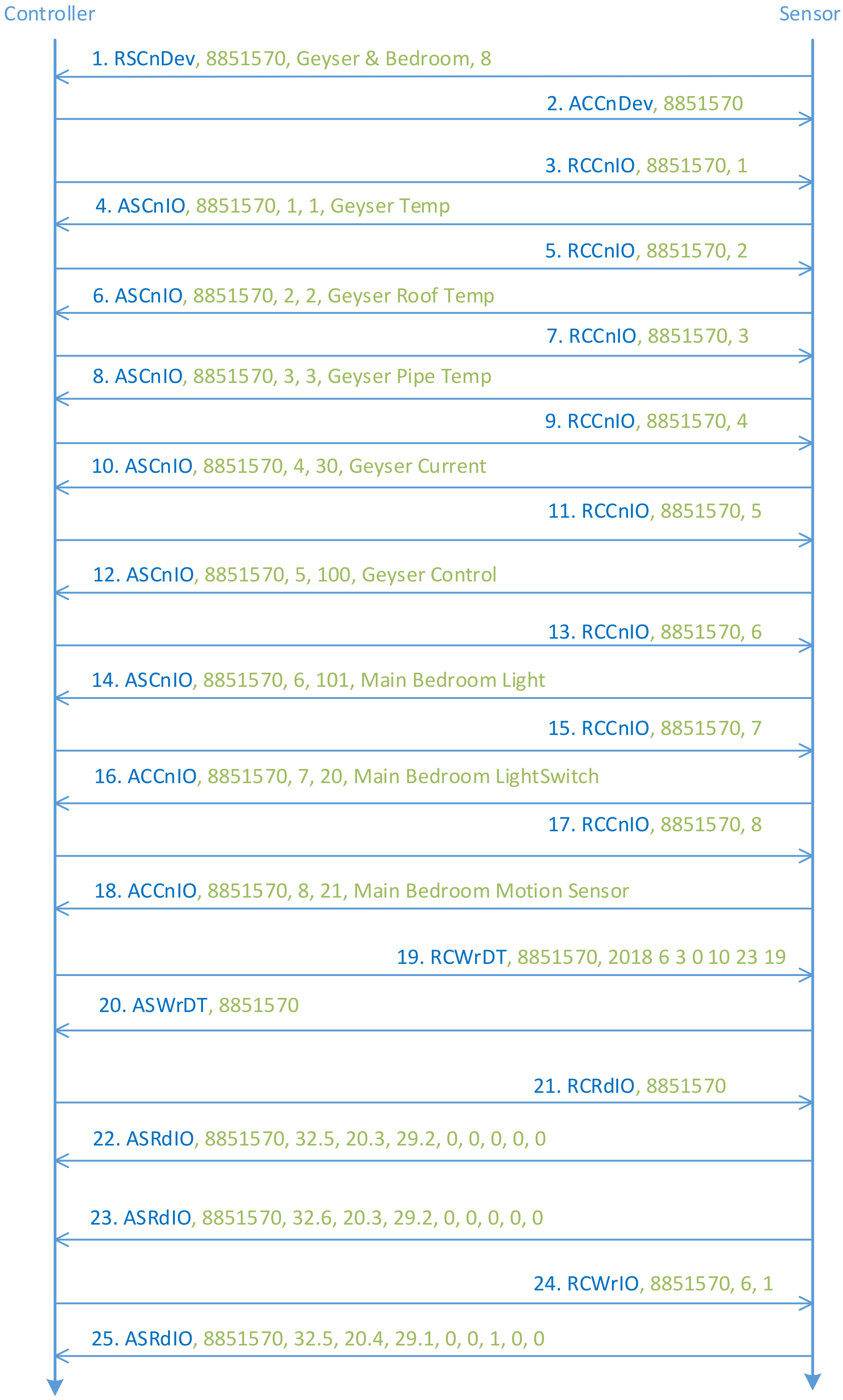


Figure C.3 Communication between controller and sensor

Figure C.4 illustrates a screenshot from the controller after Device 8851570 have been registered.

Figure C.4 *Controller Sensor registration*

Figure C.5 illustrates a screenshot from device 8851570 third endpoint.

Figure C.5 *Controller Endpoints registration*

Appendix D: Associate sensor to controller instance

After successful device registration to the controller, the endpoints need to be linked to a controller instance for example a light and light switch. To illustrate the link process two devices was registered to the controller. Figure D.1 illustrates that two devices was registered. The red icon of device with name Geyser & Bedroom implicate that it is not reachable. All the endpoints icon's showing blue indicates that it is not linked to any controller instance.

The second device registered have is identified as Geyser & Bedroom1 with an identification of 8852551. The green icon implicate that the device is switched on and reachable. The blue icons of the endpoints, indicating that none is linked to a controller instance. It is important to notice that both devices endpoints are the same, allowing the user to implement the solution with one or multiple devices depending to which device the input or output was connected to.

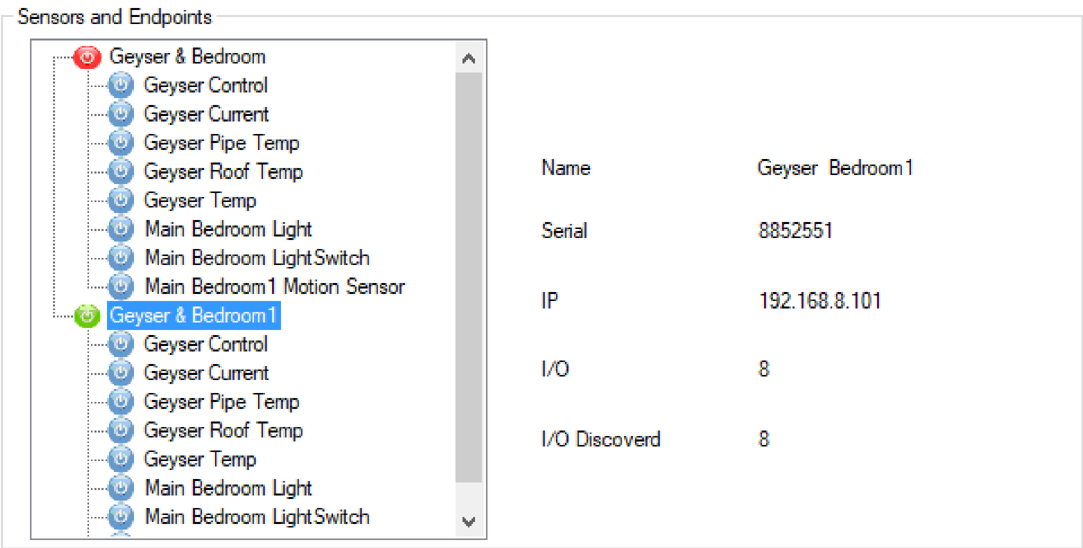


Figure D.1 Devices registered

Figure D.2 illustrates the controller's control window for lights.. For the link process three instances need to be linked to device endpoints.

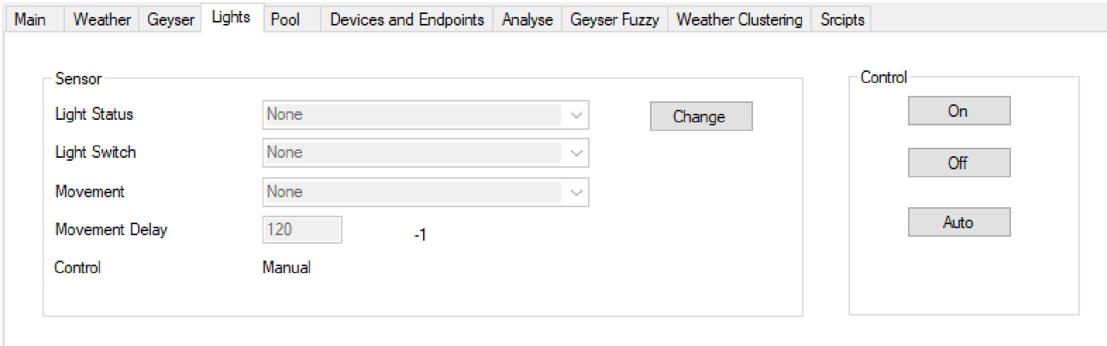


Figure D.2 Light control

After the change button was clicked, the dropdown boxes are enabled. When clicked on, the controller will only show the user the available endpoints that can control this type of endpoint. This is done by binding the same type to an instance as shown in Figure D.3. When the three instances were selected, the submit button is clicked.

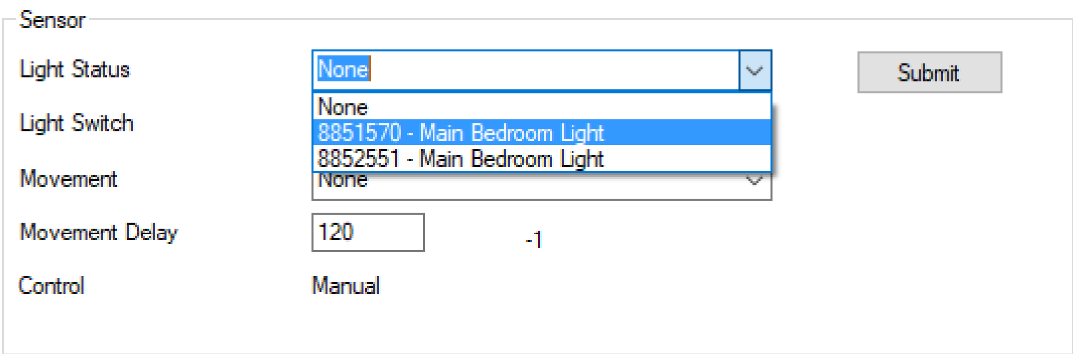


Figure D.3 Link endpoint

Figure D.4 illustrates the devices registered after endpoints was linked to a controller instance. The icon colour now changes from blue to red if the endpoint value is 0 and to green if the value is greater than 0.

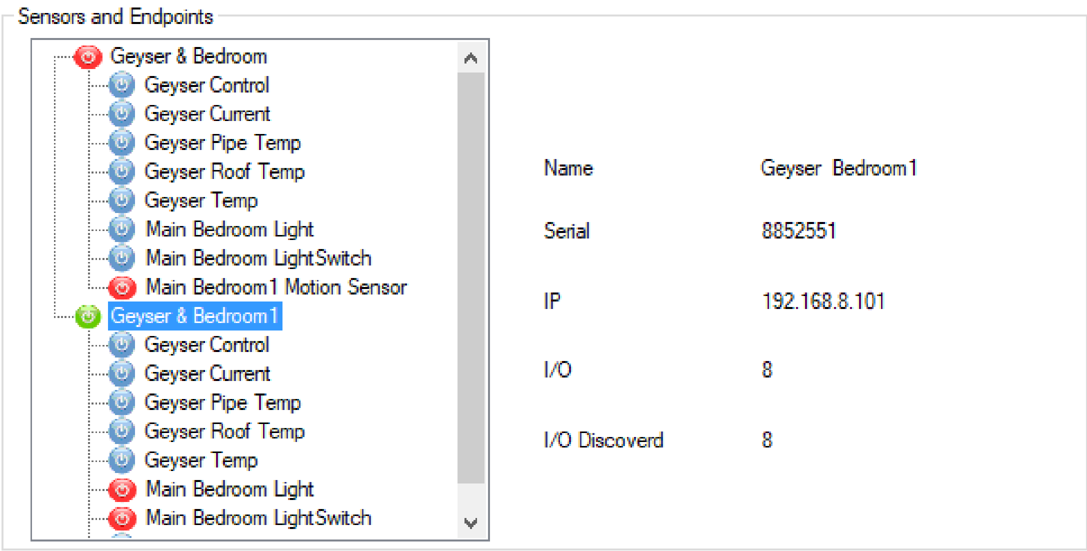


Figure D.4 Linked devices

When the "On" button is clicked the status indication of the Light changes from red to green as illustrated in Figure D.5.

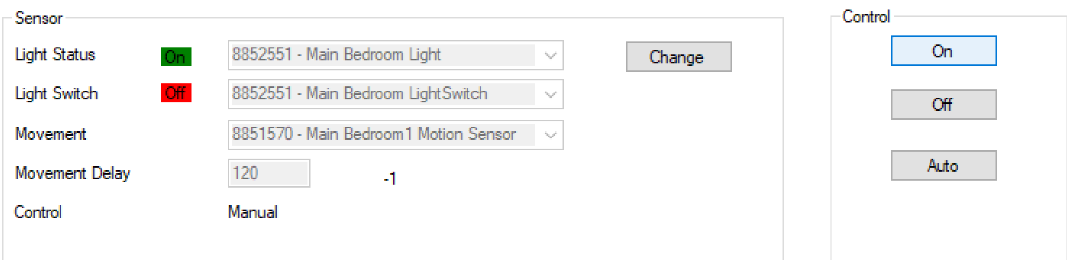


Figure D.5 Light switch on

Appendix E: Raw communication data between Sensor and controller

Table E.1 Communication capture between sensor and controller.

00047,10:25:00,192.168.8.10,192.168.8.100,RCRdIO 8851570,
00046,10:24:54,192.168.8.100,192.168.8.10,ASRdIO 8851570 32.5 20.4 29.6 0 0 0 0 0,
00045,10:24:42,192.168.8.100,192.168.8.10,ASRdIO 8851570 32.5 20.4 27.7 0 0 0 0 0,
00044,10:24:30,192.168.8.100,192.168.8.10,ASRdIO 8851570 32.5 20.4 29.3 0 0 0 0 0,
00043,10:24:26,192.168.8.100,192.168.8.10,ASRdIO 8851570 32.5 20.4 29.1 0 0 0 0 0,
00042,10:24:25,192.168.8.10,192.168.8.100,RCWrIO 8851570 6 0,
00041,10:24:22,192.168.8.100,192.168.8.10,ASRdIO 8851570 32.5 20.4 29.1 0 0 1 0 0,
00040,10:24:22,192.168.8.10,192.168.8.100,RCWrIO 8851570 6 1,
00039,10:24:21,192.168.8.100,192.168.8.10,ASWrDT 8851570,
00038,10:24:21,192.168.8.10,192.168.8.100,RCWrDT 8851570 2018 6 3 0 10 24 20,
00037,10:24:18,192.168.8.100,192.168.8.10,ASRdIO 8851570 32.5 20.4 29.1 0 0 0 0 0,
00036,10:24:08,192.168.8.100,192.168.8.10,ASWrDT 8851570,
00035,10:24:08,192.168.8.10,192.168.8.100,RCWrDT 8851570 2018 6 3 0 10 24 08,
00034,10:24:06,192.168.8.100,192.168.8.10,ASRdIO 8851570 32.5 20.4 28.2 0 0 0 0 0,
00033,10:24:05,192.168.8.100,192.168.8.10,ASWrDT 8851570,
00032,10:24:05,192.168.8.10,192.168.8.100,RCWrDT 8851570 2018 6 3 0 10 24 05,
00031,10:24:02,192.168.8.100,192.168.8.10,ASWrDT 8851570,
00030,10:24:02,192.168.8.10,192.168.8.100,RCWrDT 8851570 2018 6 3 0 10 24 02,
00029,10:24:01,192.168.8.100,192.168.8.10,ASRdIO 8851570 32.5 20.4 29.2 0 0 0 0 0,
00028,10:24:00,192.168.8.10,192.168.8.100,RCRdIO 8851570,
00027,10:23:54,192.168.8.100,192.168.8.10,ASRdIO 8851570 32.5 20.3 29.2 0 0 0 0 0,
00026,10:23:42,192.168.8.100,192.168.8.10,ASRdIO 8851570 32.5 20.3 28 0 0 0 0 0,
00025,10:23:30,192.168.8.100,192.168.8.10,ASRdIO 8851570 32.5 20.3 29.1 0 0 0 0 0,
00024,10:23:29,192.168.8.100,192.168.8.10,ASCnIO 8851570 8 21
Main_Bedroom1_Motion_Sensor,
00023,10:23:29,192.168.8.10,192.168.8.100,RCCnIO 8851570 8,
00022,10:23:26,192.168.8.100,192.168.8.10,ASCnIO 8851570 7 20
Main_Bedroom_LightSwitch,
00021,10:23:26,192.168.8.10,192.168.8.100,RCCnIO 8851570 7,
00020,10:23:23,192.168.8.100,192.168.8.10,ASCnIO 8851570 6 101 Main_Bedroom_Light,
00019,10:23:23,192.168.8.10,192.168.8.100,RCCnIO 8851570 6,
00018,10:23:20,192.168.8.100,192.168.8.10,ASCnIO 8851570 5 100 Geyser_Control,
00017,10:23:20,192.168.8.10,192.168.8.100,RCCnIO 8851570 5,
00016,10:23:19,192.168.8.100,192.168.8.10,ASWrDT 8851570,
00015,10:23:19,192.168.8.10,192.168.8.100,RCWrDT 8851570 2018 6 3 0 10 23 19,
00014,10:23:18,192.168.8.100,192.168.8.10,ASRdIO 8851570 32.5 20.3 27.7 0 0 0 0 0,
00013,10:23:17,192.168.8.100,192.168.8.10,ASCnIO 8851570 4 30 Geyser_Current,
00012,10:23:17,192.168.8.10,192.168.8.100,RCCnIO 8851570 4,
00011,10:23:14,192.168.8.100,192.168.8.10,ASCnIO 8851570 3 3 Geyser_Pipe_Temp,
00010,10:23:13,192.168.8.10,192.168.8.100,RCCnIO 8851570 3,
00009,10:23:11,192.168.8.100,192.168.8.10,ASCnIO 8851570 2 2 Geyser_Roof_Temp,
00008,10:23:10,192.168.8.10,192.168.8.100,RCCnIO 8851570 2,
00007,10:23:08,192.168.8.100,192.168.8.10,ASCnIO 8851570 1 1 Geyser_Temp,
00006,10:23:07,192.168.8.10,192.168.8.100,RCCnIO 8851570 1,
00005,10:23:07,192.168.8.10,192.168.8.100,ACCnDev 8851570,
00004,10:23:07,192.168.8.100,192.168.8.10,RSCnDev 8851570 Geyser_&_Bedroom 8,
00003,10:23:07,192.168.8.100,192.168.8.10,ASWrDT 8851570,
00002,10:23:06,192.168.8.10,192.168.8.100,RCWrDT 8851570 2018 6 3 0 10 23 06,
00001,10:23:05,192.168.8.100,192.168.8.10,ASRdIO 8851570 32.4 20.3 28 0 0 0 0 0,

Appendix F: Raw sensor data for 2017/06/13

Table F.1 Raw sensor data for 2017/06/13

Date Time	Geyser Temp	Roof Temp	Pipe Temp	Geyser Current	Geyser kWh	Weather Temprature	Weather Humidity	Move
2017/06/13 00:00	39.5	12.5	14.1	0	0			0
2017/06/13 00:05	39.4	12.3	13.7	0	0			0
2017/06/13 00:10	39.4	12.3	13.7	0	0			0
2017/06/13 00:15	39.3	12.3	13.8	0	0			0
2017/06/13 00:20	39.2	12.2	13.8	0	0			0
2017/06/13 00:25	39.1	12.2	13.8	0	0			0
2017/06/13 00:30	39	12.2	13.6	0	0			0
2017/06/13 00:35	39	12.1	13.6	0	0			0
2017/06/13 00:40	39	12.1	13.5	0	0			0
2017/06/13 00:45	38.9	12.1	13.5	0	0			0
2017/06/13 00:50	38.9	12	13.5	0	0			0
2017/06/13 00:55	38.8	12	13.4	0	0			0
2017/06/13 01:00	38.7	12	13.3	0	0			0
2017/06/13 01:05	38.6	12	12.9	0	0			0
2017/06/13 01:10	38.5	11.9	13.3	0	0			0
2017/06/13 01:15	38.4	11.8	13.2	0	0			0
2017/06/13 01:20	38.3	11.8	13.2	0	0			0
2017/06/13 01:25	38.3	11.8	13.1	0	0			0
2017/06/13 01:30	38.3	11.8	13.1	0	0			0
2017/06/13 01:35	38.2	11.7	13.1	0	0			0
2017/06/13 01:40	38.2	11.7	13	0	0			0
2017/06/13 01:45	38.1	11.7	13	0	0			0
2017/06/13 01:50	38.1	11.7	13	0	0			0
2017/06/13 01:55	37.9	11.7	12.7	0	0			0
2017/06/13 02:00	37.9	11.6	12.9	0	0	1.3	72	0
2017/06/13 02:05	37.8	11.6	12.8	0	0			0
2017/06/13 02:10	37.8	11.6	12.9	0	0			0
2017/06/13 02:15	37.8	11.6	12.8	0	0			0
2017/06/13 02:20	37.8	11.6	12.8	0	0			0
2017/06/13 02:25	37.7	11.5	12.8	0	0			0
2017/06/13 02:30	37.7	11.5	12.8	0	0			0
2017/06/13 02:35	37.7	11.5	12.7	0	0			1
2017/06/13 02:40	37.7	11.5	12.7	0	0			0
2017/06/13 02:45	37.6	11.4	12.5	0	0			0
2017/06/13 02:50	37.6	11.4	12.5	0	0			0
2017/06/13 02:55	37.6	11.4	12.7	0	0			0
2017/06/13 03:00	37.5	11.4	12.6	0	0			0
2017/06/13 03:05	37.4	11.3	12.5	0	0			0
2017/06/13 03:10	37.3	11.3	12.5	0	0			0
2017/06/13 03:15	37	11.2	12.5	0	0			0
2017/06/13 03:20	36.9	11.2	12.5	0	0			0
2017/06/13 03:25	36.6	11.1	12.5	0	0			0
2017/06/13 03:30	36.6	11.1	12.4	0	0			0
2017/06/13 03:35	36.5	11.1	12.5	0	0			0
2017/06/13 03:40	36.5	11.1	12.2	0	0			0
2017/06/13 03:45	36.4	11	12.3	0	0			0
2017/06/13 03:50	36.4	11	12.3	0	0			0
2017/06/13 03:55	36.3	11	12.4	0	0			0
2017/06/13 04:00	36.3	11	12.4	0	0			0
2017/06/13 04:05	36.2	11	12.2	0	0			0
2017/06/13 04:10	36	10.9	12.3	0	0			0
2017/06/13 04:15	35.9	10.9	12.1	0	0			0
2017/06/13 04:20	35.9	10.9	12.2	0	0			0
2017/06/13 04:25	35.8	10.8	12.2	0	0			0
2017/06/13 04:30	35.8	10.8	12.2	16	0.0165			0
2017/06/13 04:35	35.8	10.7	11.9	16.1	0.0459			0
2017/06/13 04:40	35.7	10.8	12.2	0	0.0459			0
2017/06/13 04:45	35.7	10.7	12.2	0	0.0459			0
2017/06/13 04:50	35.7	10.7	12.1	0	0.0459			0
2017/06/13 04:55	35.6	10.6	12.1	0	0.0459			0
2017/06/13 05:00	35.6	10.6	12	0	0.0459	3.81	65	0
2017/06/13 05:05	35.4	10.5	11.9	0	0			0
2017/06/13 05:10	35.4	10.5	11.9	0	0			0
2017/06/13 05:15	35.4	10.5	12	0	0			0

2017/06/13 05:20	35.3	10.5	12	0	0			0
2017/06/13 05:25	35.2	10.5	12	0	0			0
2017/06/13 05:30	35.2	10.5	11.9	0	0			0
2017/06/13 05:35	35.1	10.5	11.9	0	0			0
2017/06/13 05:40	35.1	10.5	11.8	0	0			1
2017/06/13 05:45	35	10.5	30	0	0			1
2017/06/13 05:50	35	10.5	30.5	0	0			1
2017/06/13 05:55	35	10.6	28.7	0	0			1
2017/06/13 06:00	35	10.6	26.2	0	0			1
2017/06/13 06:05	35	10.6	24.8	0	0			1
2017/06/13 06:10	34.9	10.6	30.8	0	0			1
2017/06/13 06:15	34.9	10.7	35.2	0	0			1
2017/06/13 06:20	34.9	10.7	37.5	0	0			1
2017/06/13 06:25	34.8	10.7	38.6	0	0			1
2017/06/13 06:30	34.7	10.8	37	0	0			1
2017/06/13 06:35	34.7	10.8	35.1	0	0			1
2017/06/13 06:40	34.7	10.8	32.5	0	0			0
2017/06/13 06:45	34.7	10.8	28.8	0	0			0
2017/06/13 06:50	34.7	10.8	27.1	0	0			1
2017/06/13 06:55	34.5	10.8	25.2	0	0			1
2017/06/13 07:00	34.4	10.7	23.8	0	0			1
2017/06/13 07:05	34.3	10.7	22	0	0			1
2017/06/13 07:10	34.3	10.7	20.7	0	0			0
2017/06/13 07:15	34.3	10.8	19.6	0	0			0
2017/06/13 07:20	34.2	10.8	18.6	0	0			1
2017/06/13 07:25	34.4	10.8	17.9	0	0			1
2017/06/13 07:30	34.3	10.9	17.2	0	0			1
2017/06/13 07:35	34.3	10.9	16.9	0	0			1
2017/06/13 07:40	34.3	11	16.5	0	0			1
2017/06/13 07:45	34.3	11.1	16.2	0	0			1
2017/06/13 07:50	34.3	11.2	15.7	0	0			1
2017/06/13 07:55	34.3	11.3	15.4	0	0			1
2017/06/13 08:00	34.3	11.4	37.3	0	0	21.5	24	1
2017/06/13 08:05	33.9	11.7	38.8	0	0			1
2017/06/13 08:10	33.7	11.9	39.9	0	0			1
2017/06/13 08:15	33.4	12	39.4	0	0			1
2017/06/13 08:20	33.1	12.2	38.3	0	0			1
2017/06/13 08:25	33	12.6	35.3	0	0			1
2017/06/13 08:30	33	12.9	32.1	0	0			1
2017/06/13 08:35	33.1	13.2	30.4	0	0			0
2017/06/13 08:40	33.2	13.4	28.5	0	0			0
2017/06/13 08:45	33.2	13.6	26.3	0	0			0
2017/06/13 08:50	33.2	13.9	25.3	0	0			0
2017/06/13 08:55	33.4	14.2	24.1	0	0			0
2017/06/13 09:00	33.5	14.4	23	0	0			0
2017/06/13 09:05	33.9	14.8	22.4	0	0			0
2017/06/13 09:10	34	15.1	21.2	0	0			0
2017/06/13 09:15	34.2	15.3	20.8	0	0			0
2017/06/13 09:20	34.4	15.8	20.5	0	0			0
2017/06/13 09:25	34.8	16.1	20.1	0	0			0
2017/06/13 09:30	35	16.5	19.9	0	0			0
2017/06/13 09:35	35.2	16.7	19.7	0	0			0
2017/06/13 09:40	35.5	17	19.7	0	0			0
2017/06/13 09:45	35.7	17.5	19.7	0	0			0
2017/06/13 09:50	35.7	17.9	19.6	0	0			0
2017/06/13 09:55	35.8	18.3	19.6	0	0			1
2017/06/13 10:00	35.9	18.7	19.7	0	0			1
2017/06/13 10:05	36.2	19.3	19.8	0	0			1
2017/06/13 10:10	36.6	19.9	20.1	0	0			1
2017/06/13 10:15	36.8	20.3	20.2	0	0			1
2017/06/13 10:20	37.3	20.8	20.5	0	0			1
2017/06/13 10:25	37.6	21.2	20.5	0	0			1
2017/06/13 10:30	38.1	21.8	20.9	0	0			0
2017/06/13 10:35	38.4	22.2	21.1	0	0			1
2017/06/13 10:40	38.6	22.7	21.2	0	0			1
2017/06/13 10:45	38.9	23.2	21.7	0	0			1
2017/06/13 10:50	39.2	23.7	22.1	0	0			0
2017/06/13 10:55	39.4	24.1	22.3	0	0			1
2017/06/13 11:00	39.7	24.5	22.6	0	0	24.8	18	0
2017/06/13 11:05	40	24.7	22.8	0	0			1
2017/06/13 11:10	40.3	25.2	23.3	0	0			0
2017/06/13 11:15	40.7	25.7	23.7	0	0			0
2017/06/13 11:20	41	26	24	0	0			1

2017/06/13 11:25	41.1	26.4	24.5	0	0			1
2017/06/13 11:30	41.6	26.9	25	0	0			0
2017/06/13 11:35	41.9	27.3	25.4	0	0			0
2017/06/13 11:40	42.1	27.6	26	0	0			0
2017/06/13 11:45	42.4	27.9	26.1	0	0			1
2017/06/13 11:50	42.7	28.4	26.8	0	0			1
2017/06/13 11:55	43.1	28.7	27.2	0	0			0
2017/06/13 12:00	43.5	29	27.3	0	0			0
2017/06/13 12:05	43.9	29.2	28.1	0	0			0
2017/06/13 12:10	44.1	29.5	28.2	0	0			0
2017/06/13 12:15	44.4	29.8	28.6	0	0			0
2017/06/13 12:20	44.6	30.1	28.9	0	0			0
2017/06/13 12:25	44.9	30.3	29.1	0	0			0
2017/06/13 12:30	45.1	30.7	29.8	0	0			0
2017/06/13 12:35	45.5	30.9	30	0	0			0
2017/06/13 12:40	45.7	31.1	30.4	0	0			0
2017/06/13 12:45	45.9	31.2	30.7	0	0			0
2017/06/13 12:50	46.1	31.6	31.1	0	0			1
2017/06/13 12:55	46.4	31.8	30.9	0	0			0
2017/06/13 13:00	46.7	32	31.5	0	0			0
2017/06/13 13:05	46.8	32	31.8	0	0			0
2017/06/13 13:10	47.2	32.4	31.5	0	0			0
2017/06/13 13:15	47.2	32.4	31.9	0	0			0
2017/06/13 13:20	47.4	32.7	43.3	0	0			0
2017/06/13 13:25	47.6	32.7	43.3	0	0			0
2017/06/13 13:30	47.8	32.9	42.9	0	0			0
2017/06/13 13:35	48	33	41.8	0	0			0
2017/06/13 13:40	48.2	33.2	41.2	0	0			0
2017/06/13 13:45	48.4	33.2	40.9	0	0			0
2017/06/13 13:50	48.5	33.2	40.3	0	0			0
2017/06/13 13:55	48.6	33.4	45.1	0	0			1
2017/06/13 14:00	48.8	33.5	45.5	0	0	21.2	21	0
2017/06/13 14:05	48.8	33.6	49.7	0	0			1
2017/06/13 14:10	48.4	33.6	50.4	0	0			1
2017/06/13 14:15	48.1	33.6	49.1	0	0			0
2017/06/13 14:20	47.9	33.7	47.2	0	0			1
2017/06/13 14:25	47.9	33.6	45.6	0	0			1
2017/06/13 14:30	48	33.6	44.4	0	0			1
2017/06/13 14:35	48.1	33.5	43.7	0	0			0
2017/06/13 14:40	48.1	33.6	42.5	0	0			0
2017/06/13 14:45	48.1	33.6	41.8	0	0			0
2017/06/13 14:50	48.1	33.6	40.8	0	0			0
2017/06/13 14:55	48.2	33.5	40.4	0	0			0
2017/06/13 15:00	48.2	33.5	39.6	0	0			0
2017/06/13 15:05	48.2	33.5	38.7	0	0			0
2017/06/13 15:10	48.2	33.3	37.7	0	0			0
2017/06/13 15:15	48.2	33.3	37.8	0	0			0
2017/06/13 15:20	48.2	33.1	37.4	0	0			0
2017/06/13 15:25	48.2	32.9	37	0	0			0
2017/06/13 15:30	48.2	32.8	36.5	0	0			0
2017/06/13 15:35	48.2	32.7	36.4	0	0			1
2017/06/13 15:40	48.2	32.6	36	0	0			1
2017/06/13 15:45	48	32.5	35.3	0	0			0
2017/06/13 15:50	47.9	32.2	35.4	0	0			0
2017/06/13 15:55	47.9	32	34.9	0	0			0
2017/06/13 16:00	47.6	31.7	34.5	0	0			0
2017/06/13 16:05	47.5	31.6	34.3	16.3	0.2309			0
2017/06/13 16:10	47.6	31.3	33.9	16.2	0.5182			0
2017/06/13 16:15	48.1	31.2	32.7	16.1	0.7467			0
2017/06/13 16:20	48.3	30.9	30	0	0.7467			0
2017/06/13 16:25	48.3	30.5	29.7	0	0.7467			0
2017/06/13 16:30	48.1	30.3	29.4	0	0.7467			0
2017/06/13 16:35	47.9	30	29.1	0	0.7467			1
2017/06/13 16:40	47.7	29.8	28.9	0	0.7467			0
2017/06/13 16:45	47.5	29.4	28.7	0	0.7467			0
2017/06/13 16:50	47.3	29.1	28.4	0	0.7467			0
2017/06/13 16:55	47.2	28.7	27.9	0	0.7467			0
2017/06/13 17:00	47	28.5	27.7	0	0.7467	9.42	53	0
2017/06/13 17:05	46.8	28.2	27.4	0	0			0
2017/06/13 17:10	46.6	27.9	27.1	0	0			0
2017/06/13 17:15	46.6	27.6	31	0	0			0
2017/06/13 17:20	46.5	27.4	39.4	0	0			0
2017/06/13 17:25	46.2	27	39.5	0	0			0

2017/06/13 17:30	46.1	26.8	38.2	0	0			0
2017/06/13 17:35	46	26.5	36.5	0	0			0
2017/06/13 17:40	45.9	26.2	35.2	0	0			0
2017/06/13 17:45	45.8	26	34	0	0			0
2017/06/13 17:50	45.7	25.7	33	0	0			0
2017/06/13 17:55	45.6	25.4	31.9	0	0			0
2017/06/13 18:00	45.5	25.1	30.9	0	0			0
2017/06/13 18:05	45.3	24.9	31.7	16.3	0.1186			0
2017/06/13 18:10	45.5	24.6	31.7	16.4	0.4103			0
2017/06/13 18:15	46	24.4	30.7	16.3	0.701			0
2017/06/13 18:20	46.7	24.1	30.1	16.4	0.9927			0
2017/06/13 18:25	47.2	23.9	28.7	16.4	1.1561			0
2017/06/13 18:30	47.3	23.7	25.9	0	1.1561			0
2017/06/13 18:35	47.2	23.5	25.3	0	1.1561			0
2017/06/13 18:40	47.2	23.3	24.7	0	1.1561			0
2017/06/13 18:45	47	23	24.2	0	1.1561			0
2017/06/13 18:50	46.8	22.8	26	0	1.1561			1
2017/06/13 18:55	46.5	22.5	32.9	0	1.1561			0
2017/06/13 19:00	46.1	22.2	44.5	0	1.1561			1
2017/06/13 19:05	45.9	22	44.5	0	0			1
2017/06/13 19:10	45.7	21.8	41.4	0	0			1
2017/06/13 19:15	45.6	21.5	38.9	0	0			0
2017/06/13 19:20	45.3	21.3	37	0	0			0
2017/06/13 19:25	45.1	21.1	45	0	0			0
2017/06/13 19:30	44.7	20.9	49.7	0	0			0
2017/06/13 19:35	44.6	20.8	45.9	0	0			0
2017/06/13 19:40	44.4	20.6	43.6	0	0			0
2017/06/13 19:45	44.3	20.4	49	16.3	0.1633			1
2017/06/13 19:50	42.7	20.2	50	16.4	0.456			0
2017/06/13 19:55	36.8	19.9	44.1	16.2	0.7458			0
2017/06/13 20:00	35.8	19.7	41.2	16.2	1.0372	5.31	63	0
2017/06/13 20:05	36.7	19.5	37.5	16.2	0.003			0
2017/06/13 20:10	36.8	19.3	33.6	0	0.003			0
2017/06/13 20:15	36.8	19.2	31.9	0	0.003			0
2017/06/13 20:20	36.7	19	30.2	0	0.003			0
2017/06/13 20:25	36.5	18.8	28.5	0	0.003			0
2017/06/13 20:30	36.1	18.6	27.1	0	0.003			1
2017/06/13 20:35	35.9	18.5	25.8	0	0.003			1
2017/06/13 20:40	35.7	18.3	24.7	0	0.003			1
2017/06/13 20:45	35.5	18.1	37.9	0	0.003			0
2017/06/13 20:50	34.9	18	39.1	0	0.003			0
2017/06/13 20:55	34.6	17.9	37.7	0	0.003			1
2017/06/13 21:00	34.5	17.8	35.1	0	0.003			0
2017/06/13 21:05	34.3	17.6	32.5	0	0			1
2017/06/13 21:10	34.2	17.5	30.5	0	0			1
2017/06/13 21:15	34.1	17.3	36	0	0			0
2017/06/13 21:20	33.6	17.1	38.3	0	0			1
2017/06/13 21:25	33.2	17	37.4	0	0			0
2017/06/13 21:30	33.2	16.9	34.7	0	0			1
2017/06/13 21:35	33.1	16.8	32.5	0	0			1
2017/06/13 21:40	33	16.7	30.3	0	0			0
2017/06/13 21:45	32.9	16.6	28.5	0	0			0
2017/06/13 21:50	32.8	16.6	26.6	0	0			0
2017/06/13 21:55	32.6	16.5	25.1	0	0			0
2017/06/13 22:00	32.5	16.3	24.5	0	0			0
2017/06/13 22:05	32.4	16.3	25	0	0			1
2017/06/13 22:10	32.3	16.2	24	0	0			0
2017/06/13 22:15	32.2	16.1	22.6	0	0			0
2017/06/13 22:20	32.2	16	21.7	0	0			0
2017/06/13 22:25	32.1	15.9	21.2	0	0			0
2017/06/13 22:30	32.1	15.8	20.6	0	0			0
2017/06/13 22:35	32.1	15.8	20.1	0	0			0
2017/06/13 22:40	32	15.7	19.5	0	0			0
2017/06/13 22:45	31.9	15.6	19.1	0	0			0
2017/06/13 22:50	31.8	15.5	18.8	0	0			1
2017/06/13 22:55	31.8	15.4	18.4	0	0			0
2017/06/13 23:00	31.7	15.3	18.1	0	0	3.48	67	0
2017/06/13 23:05	31.6	15.3	17.9	0	0			0
2017/06/13 23:10	31.5	15.2	17.6	0	0			0
2017/06/13 23:15	31.4	15.1	17.4	0	0			0
2017/06/13 23:20	31.4	15.1	17.2	0	0			0
2017/06/13 23:25	31.3	15	16.9	0	0			0
2017/06/13 23:30	31.2	15	16.8	0	0			0

Appendix G: Raw sensor data for 2017/07/28

Table G.1 Raw sensor data for 2017/07/28

Date Time	Geyser Temp	Roof Temp	Pipe Temp	Geyser Current	Geyser kWh	Move	Weather Temprature	Weather Humidity
2017/07/28 00:00	39.2	13.4	14.7	0	0	1		
2017/07/28 00:05	39.2	13.3	14.4	0	0	1		
2017/07/28 00:10	39.1	13.3	14.6	0	0	0		
2017/07/28 00:15	39.1	13.2	14.5	0	0	0		
2017/07/28 00:20	39	13.1	14.4	0	0	0		
2017/07/28 00:25	39	13	14.2	0	0	0		
2017/07/28 00:30	38.9	13	13.9	0	0	0		
2017/07/28 00:35	38.9	12.8	13.9	0	0	0		
2017/07/28 00:40	38.9	12.7	13.9	0	0	0		
2017/07/28 00:45	38.7	12.7	13.9	0	0	0		
2017/07/28 00:50	38.7	12.5	13.8	0	0	0		
2017/07/28 00:55	38.7	12.5	13.7	0	0	0		
2017/07/28 01:00	38.6	12.4	13.7	0	0	0		
2017/07/28 01:05	38.6	12.3	13.3	0	0	0		
2017/07/28 01:10	38.5	12.2	13.5	0	0	0		
2017/07/28 01:15	38.5	12.1	13.2	0	0	0		
2017/07/28 01:20	38.5	12.1	13.4	0	0	0		
2017/07/28 01:25	38.5	12	13.4	0	0	0		
2017/07/28 01:30	38.4	12	13.3	0	0	0		
2017/07/28 01:35	38.3	11.9	13.2	0	0	0		
2017/07/28 01:40	38.3	11.9	13	0	0	0		
2017/07/28 01:45	38.2	11.8	12.9	0	0	0		
2017/07/28 01:50	38.2	11.7	12.9	0	0	0		
2017/07/28 01:55	38.1	11.6	12.8	0	0	0		
2017/07/28 02:00	38	11.5	12.9	0	0	0	0.93	72
2017/07/28 02:05	38	11.4	12.7	0	0	1		
2017/07/28 02:10	38	11.4	12.5	0	0	0		
2017/07/28 02:15	37.9	11.4	12.5	0	0	0		
2017/07/28 02:20	37.9	11.3	12.6	0	0	0		
2017/07/28 02:25	37.9	11.3	12.6	0	0	0		
2017/07/28 02:30	37.8	11.3	12.6	0	0	0		
2017/07/28 02:35	37.8	11.3	12.6	0	0	0		
2017/07/28 02:40	37.8	11.3	12.5	0	0	0		
2017/07/28 02:45	37.8	11.2	12.5	0	0	0		
2017/07/28 02:50	37.7	11.2	12.5	0	0	0		
2017/07/28 02:55	37.6	11.2	12.5	0	0	0		
2017/07/28 03:00	37.6	11.2	12.5	0	0	1		
2017/07/28 03:05	37.6	11.2	12.1	0	0	0		
2017/07/28 03:10	37.5	11.2	12.2	0	0	0		
2017/07/28 03:15	37.5	11.1	12.5	0	0	0		
2017/07/28 03:20	37.5	11.1	12.5	0	0	0		
2017/07/28 03:25	37.5	11.1	12.2	0	0	0		
2017/07/28 03:30	37.4	11.1	12.2	0	0	0		
2017/07/28 03:35	37.4	11.1	12.3	0	0	0		
2017/07/28 03:40	37.4	11.1	12.2	0	0	0		
2017/07/28 03:45	37.4	11.1	12.1	0	0	0		
2017/07/28 03:50	37.4	11.1	12.2	0	0	0		
2017/07/28 03:55	37.3	11	12.3	0	0	0		
2017/07/28 04:00	37.3	11	12.3	0	0	0		
2017/07/28 04:05	37.2	11	12.4	0	0	0		
2017/07/28 04:10	37.1	11	12.2	0	0	0		
2017/07/28 04:15	37.1	11	12.2	0	0	0		
2017/07/28 04:20	37	11	12.2	0	0	0		
2017/07/28 04:25	36.9	11	11.9	0	0	0		
2017/07/28 04:30	36.9	11	12.3	0	0	0		
2017/07/28 04:35	36.9	11	11.8	15.9	0.15	1		
2017/07/28 04:40	37.9	10.9	12.1	15.7	0.42	1		
2017/07/28 04:45	39.1	10.9	12.1	15.8	0.7	1		
2017/07/28 04:50	40.6	11	12.2	16	0.98	1		
2017/07/28 04:55	41.1	10.9	12.2	15.8	0.99	1		
2017/07/28 05:00	41.2	10.9	12.2	0	0.99	1	3.3	72
2017/07/28 05:05	41.2	10.9	19.6	0	0	1		
2017/07/28 05:10	41	10.9	19.8	0	0	1		
2017/07/28 05:15	40.9	10.9	19.5	0	0	0		

2017/07/28 05:20	40.8	10.9	18.9	0	0	0		
2017/07/28 05:25	40.7	10.8	18	0	0	0		
2017/07/28 05:30	40.6	10.9	17.4	0	0	0		
2017/07/28 05:35	40.4	10.8	16.6	0	0	0		
2017/07/28 05:40	40.3	10.8	16.2	0	0	0		
2017/07/28 05:45	40.1	10.7	15.7	0	0	0		
2017/07/28 05:50	40.1	10.7	15.7	0	0	0		
2017/07/28 05:55	40	10.7	15.1	0	0	1		
2017/07/28 06:00	39.9	10.7	14.8	0	0	1		
2017/07/28 06:05	39.8	10.8	14.6	0	0	1		
2017/07/28 06:10	39.7	10.8	14.4	0	0	1		
2017/07/28 06:15	39.6	10.9	15.9	0	0	1		
2017/07/28 06:20	39.4	10.8	30	0	0	1		
2017/07/28 06:25	39.1	10.9	35	0	0	1		
2017/07/28 06:30	38.9	10.9	33.3	0	0	1		
2017/07/28 06:35	38.9	10.9	31.8	0	0	1		
2017/07/28 06:40	38.8	10.9	29	0	0	1		
2017/07/28 06:45	38.7	10.9	26.7	0	0	1		
2017/07/28 06:50	38.6	10.9	25	0	0	1		
2017/07/28 06:55	38.6	10.9	22.9	16.3	0.17	1		
2017/07/28 07:00	39.7	11	21.5	16.2	0.46	1		
2017/07/28 07:05	41.1	11.1	20.3	16.2	0.23	1		
2017/07/28 07:10	42.4	11.1	19.6	16.3	0.52	1		
2017/07/28 07:15	43.9	11.1	18.6	16.3	0.81	0		
2017/07/28 07:20	45	11.1	18	16.1	0.98	0		
2017/07/28 07:25	45.1	11.2	17.6	0	0.98	0		
2017/07/28 07:30	45.1	11.3	16.9	0	0.98	1		
2017/07/28 07:35	45.2	11.4	16.5	15.9	1.13	0		
2017/07/28 07:40	45.6	11.4	16.2	0	1.13	0		
2017/07/28 07:45	45.6	11.5	15.8	0	1.13	0		
2017/07/28 07:50	45.5	11.6	37.2	15.9	1.15	0		
2017/07/28 07:55	44.8	11.8	38.8	15.9	1.37	0		
2017/07/28 08:00	35.6	11.9	32.6	0	1.37	1	18.44	33
2017/07/28 08:05	30.4	12	32.6	0	0	1		
2017/07/28 08:10	28.5	12.2	32.8	0	0	0		
2017/07/28 08:15	27	12.4	30.8	0	0	0		
2017/07/28 08:20	26.4	12.8	31.5	0	0	0		
2017/07/28 08:25	26	13	31.9	0	0	0		
2017/07/28 08:30	25.6	13.3	31.7	0	0	0		
2017/07/28 08:35	25.1	13.5	31.6	0	0	1		
2017/07/28 08:40	25	13.7	32.6	0	0	1		
2017/07/28 08:45	25.2	14	32.1	0	0	1		
2017/07/28 08:50	25.3	14.3	31.4	0	0	1		
2017/07/28 08:55	25.5	14.6	28.9	0	0	1		
2017/07/28 09:00	25.8	14.8	27.4	0	0	1		
2017/07/28 09:05	26	15.1	26.7	0	0	1		
2017/07/28 09:10	26.2	15.3	25	0	0	0		
2017/07/28 09:15	26.5	15.7	24.3	0	0	1		
2017/07/28 09:20	26.6	15.9	23.6	0	0	1		
2017/07/28 09:25	27	16.2	22.6	0	0	1		
2017/07/28 09:30	27.3	16.5	21.8	0	0	1		
2017/07/28 09:35	27.5	16.7	21.4	0	0	1		
2017/07/28 09:40	27.8	17	27.1	0	0	1		
2017/07/28 09:45	28.2	17.3	29.8	0	0	1		
2017/07/28 09:50	28.5	17.6	29.7	0	0	1		
2017/07/28 09:55	28.9	17.9	29	0	0	1		
2017/07/28 10:00	29.2	18.2	28.1	0	0	1		
2017/07/28 10:05	29.7	18.6	27.4	0	0	1		
2017/07/28 10:10	30.2	18.8	26.7	0	0	0		
2017/07/28 10:15	30.6	19.2	25.9	0	0	0		
2017/07/28 10:20	31.1	19.5	25.4	0	0	0		
2017/07/28 10:25	31.5	19.9	25	0	0	0		
2017/07/28 10:30	32.2	20.3	24.6	0	0	0		
2017/07/28 10:35	32.5	20.5	24.4	0	0	0		
2017/07/28 10:40	33.1	20.8	23.9	0	0	1		
2017/07/28 10:45	33.5	21.1	23.7	0	0	1		
2017/07/28 10:50	34	21.4	23.7	0	0	1		
2017/07/28 10:55	34.6	21.8	23.6	0	0	1		
2017/07/28 11:00	34.8	22.1	23.7	0	0	1	21.66	24
2017/07/28 11:05	35.1	22.4	23.7	0	0	1		
2017/07/28 11:10	35.6	22.8	23.7	0	0	1		
2017/07/28 11:15	35.8	23.1	23.8	0	0	0		
2017/07/28 11:20	36.1	23.4	23.9	0	0	1		

2017/07/28 11:25	36.2	23.7	23.9	0	0	0		
2017/07/28 11:30	36.4	24	23.9	0	0	1		
2017/07/28 11:35	36.6	24.5	24	0	0	1		
2017/07/28 11:40	36.8	24.8	24.5	0	0	1		
2017/07/28 11:45	37.2	25.1	24.6	0	0	1		
2017/07/28 11:50	37.3	25.3	25	0	0	1		
2017/07/28 11:55	37.6	25.7	25.1	0	0	1		
2017/07/28 12:00	38	25.9	25.1	0	0	1		
2017/07/28 12:05	38.2	26.1	25.3	0	0	1		
2017/07/28 12:10	38.5	26.3	25.7	0	0	1		
2017/07/28 12:15	38.7	26.4	25.9	0	0	1		
2017/07/28 12:20	39	26.7	25.9	0	0	0		
2017/07/28 12:25	39.2	27	26.1	0	0	0		
2017/07/28 12:30	39.5	27.1	26.2	0	0	0		
2017/07/28 12:35	39.8	27.4	26.2	0	0	0		
2017/07/28 12:40	40.1	27.6	26.8	0	0	0		
2017/07/28 12:45	40.4	27.8	27.1	0	0	0		
2017/07/28 12:50	40.7	27.9	27.3	0	0	0		
2017/07/28 12:55	41	28.1	27.5	0	0	0		
2017/07/28 13:00	41.2	28.1	27.6	0	0	0		
2017/07/28 13:05	41.5	28.4	28	0	0	0		
2017/07/28 13:10	41.7	28.6	28	0	0	0		
2017/07/28 13:15	42	28.6	28.3	0	0	0		
2017/07/28 13:20	42.2	28.8	28.5	0	0	0		
2017/07/28 13:25	42.5	28.9	28.5	0	0	0		
2017/07/28 13:30	42.7	28.9	28.7	0	0	0		
2017/07/28 13:35	42.9	28.9	28.8	0	0	0		
2017/07/28 13:40	43.1	29	28.9	0	0	0		
2017/07/28 13:45	43.5	29.1	29	0	0	0		
2017/07/28 13:50	43.8	29.1	29	0	0	0		
2017/07/28 13:55	44	29.1	29	0	0	1		
2017/07/28 14:00	44.1	29.2	29	0	0	1	19.99	20
2017/07/28 14:05	44.4	29.2	29.1	0	0	1		
2017/07/28 14:10	44.6	29.2	29.1	0	0	1		
2017/07/28 14:15	44.7	29.2	33	0	0	1		
2017/07/28 14:20	44.8	29.3	37.5	0	0	0		
2017/07/28 14:25	44.9	29.4	39	0	0	0		
2017/07/28 14:30	45.1	29.5	40	0	0	0		
2017/07/28 14:35	45.2	29.5	40	0	0	0		
2017/07/28 14:40	45.2	29.4	39.4	0	0	0		
2017/07/28 14:45	45.3	29.4	38.3	0	0	0		
2017/07/28 14:50	45.4	29.3	37.2	0	0	1		
2017/07/28 14:55	45.5	29.2	37.7	0	0	1		
2017/07/28 15:00	45.5	29.1	42.3	0	0	1		
2017/07/28 15:05	44.9	29.1	41.9	0	0	0		
2017/07/28 15:10	44.3	29	37.2	0	0	0		
2017/07/28 15:15	43.1	28.8	37.1	0	0	0		
2017/07/28 15:20	40.1	28.8	37	0	0	0		
2017/07/28 15:25	34.9	28.7	36.6	0	0	1		
2017/07/28 15:30	31.8	28.6	35.6	0	0	0		
2017/07/28 15:35	30.8	28.6	35.5	0	0	0		
2017/07/28 15:40	30.1	28.4	35.1	0	0	0		
2017/07/28 15:45	29.7	28.3	34.5	0	0	1		
2017/07/28 15:50	29.9	28	33.4	0	0	1		
2017/07/28 15:55	30.2	27.8	33	0	0	1		
2017/07/28 16:00	30.1	27.6	33.9	0	0	0		
2017/07/28 16:05	29.5	27.5	33.7	16.1	0.13	1		
2017/07/28 16:10	31.1	27.5	36.1	16.1	0.42	1		
2017/07/28 16:15	33.3	27.4	34.2	16.1	0.7	1		
2017/07/28 16:20	35.6	27.2	35	16.1	0.98	1		
2017/07/28 16:25	37.4	27	34.6	16.1	1.26	1		
2017/07/28 16:30	39	26.8	32.9	16.1	1.54	1		
2017/07/28 16:35	40.4	26.6	33.6	16	1.82	1		
2017/07/28 16:40	41.6	26.4	33.6	16	2.1	0		
2017/07/28 16:45	42.8	26.2	32.6	15.8	2.38	1		
2017/07/28 16:50	44.1	26	31.6	15.9	2.66	0		
2017/07/28 16:55	45.3	25.8	30.9	15.8	2.94	0		
2017/07/28 17:00	46.4	25.6	30.7	15.9	3.22	1	7.69	52
2017/07/28 17:05	47.4	25.3	29.8	15.9	0.23	1		
2017/07/28 17:10	48.4	25.1	28.2	16	0.51	1		
2017/07/28 17:15	49.4	24.8	27	16	0.79	1		
2017/07/28 17:20	50.8	24.6	25.8	16	1.07	0		
2017/07/28 17:25	51.4	24.5	25.4	15.9	1.16	0		

2017/07/28 17:30	51.4	24.3	25.1	0	1.16	0		
2017/07/28 17:35	51.2	24.1	24.8	0	1.16	0		
2017/07/28 17:40	51	23.9	24.5	0	1.16	1		
2017/07/28 17:45	50.6	23.7	25.9	0	1.16	0		
2017/07/28 17:50	50.2	23.5	25.9	0	1.16	0		
2017/07/28 17:55	49.9	23.4	25.4	0	1.16	0		
2017/07/28 18:00	49.4	23	25.1	0	1.16	0		
2017/07/28 18:05	49	22.9	24.6	16.1	0.04	0		
2017/07/28 18:10	49.2	22.6	24.4	16.1	0.33	0		
2017/07/28 18:15	50.3	22.3	24	16.1	0.61	0		
2017/07/28 18:20	51.4	22.1	23.8	16	0.89	1		
2017/07/28 18:25	52.4	21.9	24.5	16	1.17	1		
2017/07/28 18:30	53.5	21.7	23.1	16.1	1.46	0		
2017/07/28 18:35	54.5	21.4	22.8	16.1	1.74	1		
2017/07/28 18:40	55.1	21.2	22.4	16.1	1.8	0		
2017/07/28 18:45	55.2	21	22.1	0	1.8	0		
2017/07/28 18:50	54.9	20.8	21.8	0	1.8	0		
2017/07/28 18:55	54.4	20.5	21.5	0	1.8	0		
2017/07/28 19:00	54.1	20.4	21.3	0	1.8	0		
2017/07/28 19:05	53.6	20.2	20.9	0	0	0		
2017/07/28 19:10	53.3	19.9	20.7	0	0	0		
2017/07/28 19:15	52.8	19.7	20.6	0	0	1		
2017/07/28 19:20	52.6	19.4	20.5	16.2	0.03	1		
2017/07/28 19:25	52.8	19.2	20.4	16.3	0.32	0		
2017/07/28 19:30	54.1	18.9	19.9	16.3	0.61	0		
2017/07/28 19:35	55.3	18.7	20	16.3	0.76	1		
2017/07/28 19:40	55.5	18.6	20	0	0.76	1		
2017/07/28 19:45	55.5	18.5	19.7	0	0.76	0		
2017/07/28 19:50	55.2	18.4	19.5	0	0.76	0		
2017/07/28 19:55	54.9	18.2	19.2	0	0.76	1		
2017/07/28 20:00	54.4	18	19	0	0.76	1	2.96	64
2017/07/28 20:05	54.1	17.8	19	0	0	0		
2017/07/28 20:10	53.6	17.6	18.8	0	0	0		
2017/07/28 20:15	53.4	17.5	18.7	0	0	1		
2017/07/28 20:20	53	17.4	18.5	0	0	1		
2017/07/28 20:25	52.7	17.2	18.5	0	0	0		
2017/07/28 20:30	52.4	17.1	18.3	0	0	0		
2017/07/28 20:35	52.2	17	18.1	0	0	0		
2017/07/28 20:40	51.9	16.9	17.9	0	0	0		
2017/07/28 20:45	51.7	16.8	17.8	0	0	0		
2017/07/28 20:50	51.5	16.7	17.8	0	0	0		
2017/07/28 20:55	51.3	16.6	17.6	0	0	0		
2017/07/28 21:00	51.1	16.5	17.5	0	0	0		
2017/07/28 21:05	50.8	16.4	17.4	0	0	0		
2017/07/28 21:10	50.6	16.3	17.4	0	0	0		
2017/07/28 21:15	50.4	16.2	17.1	0	0	1		
2017/07/28 21:20	50.2	16.1	16.9	0	0	1		
2017/07/28 21:25	50.1	16	17	0	0	0		
2017/07/28 21:30	49.9	15.9	16.8	0	0	0		
2017/07/28 21:35	49.7	15.7	16.7	0	0	0		
2017/07/28 21:40	49.5	15.6	16.7	0	0	1		
2017/07/28 21:45	49.3	15.5	16.6	0	0	1		
2017/07/28 21:50	49.2	15.4	16.5	0	0	0		
2017/07/28 21:55	49	15.3	16.2	0	0	0		
2017/07/28 22:00	48.9	15.2	16.4	0	0	0		
2017/07/28 22:05	48.8	15.1	16.3	0	0	0		
2017/07/28 22:10	48.6	15.1	16.2	0	0	0		
2017/07/28 22:15	48.5	15	16.1	0	0	0		
2017/07/28 22:20	48.4	14.9	16	0	0	1		
2017/07/28 22:25	48.2	14.9	15.9	0	0	0		
2017/07/28 22:30	48.1	14.7	15.8	0	0	0		
2017/07/28 22:35	47.8	14.6	15.7	0	0	0		
2017/07/28 22:40	47.6	14.5	15.5	0	0	0		
2017/07/28 22:45	47.5	14.4	15.2	0	0	0		
2017/07/28 22:50	47.4	14.2	15.4	0	0	0		
2017/07/28 22:55	47.4	14.1	15.3	0	0	0		
2017/07/28 23:00	47.2	14.1	15.3	0	0	0	0.85	68
2017/07/28 23:05	47.2	14	15.1	0	0	0		
2017/07/28 23:10	47.2	13.9	15	0	0	0		
2017/07/28 23:15	47.1	13.8	25.9	0	0	1		
2017/07/28 23:20	46.9	13.7	47.4	0	0	1		
2017/07/28 23:25	46.5	13.7	48.1	0	0	1		
2017/07/28 23:30	46.5	13.7	46.7	0	0	1		

Appendix H: Raw sensor data for 2019/03/30

Table H.1 Raw sensor data for 2019/03/30

Date Time	Weather Temperature	Weather Humidity	Weather CloudCover	Weather WindSpeed	Pool Temp	Pool Outside Temp	Pool Heat Temp	PoolPump kWh	PoolPump Current
2019/03/30 00:00					23	21.9	23	0	0
2019/03/30 00:05					22.8	21.8	22.8	0	0
2019/03/30 00:10					22.9	21.7	22.8	0	0
2019/03/30 00:15					22.7	21.6	22.7	0	0
2019/03/30 00:20					22.7	21.5	22.7	0	0
2019/03/30 00:25					22.4	21.3	22.5	0	0
2019/03/30 00:30					22.4	21.3	22.5	0	0
2019/03/30 00:35					22.2	21.2	22.4	0	0
2019/03/30 00:40					22.2	21.1	22.3	0	0
2019/03/30 00:45					22.2	21	22.3	0	0
2019/03/30 00:50					22.1	21	22.3	0	0
2019/03/30 00:55					22	20.9	22.2	0	0
2019/03/30 01:00					22	20.9	22.1	0	0
2019/03/30 01:05					22	20.9	22.1	0	0
2019/03/30 01:10					21.9	20.8	22	0	0
2019/03/30 01:15					21.7	20.7	21.9	0	0
2019/03/30 01:20					21.7	20.6	21.9	0	0
2019/03/30 01:25					21.7	20.5	21.8	0	0
2019/03/30 01:30					21.6	20.5	21.7	0	0
2019/03/30 01:35					21.5	20.5	21.7	0	0
2019/03/30 01:40					21.5	20.4	21.6	0	0
2019/03/30 01:45					21.4	20.4	21.5	0	0
2019/03/30 01:50					21.2	20.3	21.5	0	0
2019/03/30 01:55					21.3	20.2	21.4	0	0
2019/03/30 02:00	13.54	75	0	1.5	21.2	20.2	21.3	0	0
2019/03/30 02:05					21.2	20.2	21.3	0	0
2019/03/30 02:10					21	19.9	21.1	0	0
2019/03/30 02:15					20.9	19.8	21.1	0	0
2019/03/30 02:20					20.9	19.8	21	0	0
2019/03/30 02:25					21	19.8	21	0	0
2019/03/30 02:30					20.8	19.7	20.9	0	0
2019/03/30 02:35					20.8	19.6	20.8	0	0
2019/03/30 02:40					20.7	19.6	20.8	0	0
2019/03/30 02:45					20.7	19.5	20.8	0	0
2019/03/30 02:50					20.7	19.5	20.7	0	0
2019/03/30 02:55					20.5	19.4	20.7	0	0
2019/03/30 03:00					20.7	19.4	20.6	0	0
2019/03/30 03:05					20.7	19.4	20.5	0	0
2019/03/30 03:10					20.5	19.3	20.4	0	0
2019/03/30 03:15					20.4	19.3	20.4	0	0
2019/03/30 03:20					20.6	19.3	20.3	0	0
2019/03/30 03:25					20.5	19.3	20.3	0	0
2019/03/30 03:30					20.4	19.2	20.2	0	0
2019/03/30 03:35					20.3	19.2	20.2	0	0
2019/03/30 03:40					20.3	19.1	20.2	0	0
2019/03/30 03:45					20.3	18.8	20	0	0
2019/03/30 03:50					20.2	18.8	20	0	0
2019/03/30 03:55					20.2	18.7	20	0	0
2019/03/30 04:00					20.1	18.7	20	0	0
2019/03/30 04:05					20	18.7	19.9	0	0
2019/03/30 04:10					19.9	18.7	19.8	0	0
2019/03/30 04:15					20	18.7	19.8	0	0
2019/03/30 04:20					20	18.6	19.8	0	0
2019/03/30 04:25					19.9	18.5	19.7	0	0
2019/03/30 04:30					19.9	18.5	19.6	0	0
2019/03/30 04:35					19.9	18.5	19.6	0	0
2019/03/30 04:40					19.8	18.5	19.5	0	0
2019/03/30 04:45					19.9	18.4	19.5	0	0
2019/03/30 04:50					19.8	18.4	19.5	0	0
2019/03/30 04:55					19.8	18.4	19.4	0	0
2019/03/30 05:00	22.47	52	0	1.87	19.8	18.3	19.4	0	0
2019/03/30 05:05					19.7	18.3	19.4	0	0
2019/03/30 05:10					19.7	18.2	19.3	0	0
2019/03/30 05:15					19.6	18.2	19.3	0	0

2019/03/30 05:20					19.6	18.3	19.4	0	0
2019/03/30 05:25					19.5	18.3	19.3	0	0
2019/03/30 05:30					19.7	18.2	19.3	0	0
2019/03/30 05:35					19.5	18.2	19.2	0	0
2019/03/30 05:40					19.7	18.2	19.2	0	0
2019/03/30 05:45					19.6	18.2	19.2	0	0
2019/03/30 05:50					19.5	18.1	19.2	0	0
2019/03/30 05:55					19.6	18.2	19.2	0	0
2019/03/30 06:00					19.5	18.2	19.2	0	0
2019/03/30 06:05					19.6	18.2	19.1	0	0
2019/03/30 06:10					19.6	18.2	19.1	0	0
2019/03/30 06:15					19.5	18.3	19.1	0	0
2019/03/30 06:20					19.7	18.2	19.1	0	0
2019/03/30 06:25					19.7	18.3	19.1	0	0
2019/03/30 06:30					19.7	18.3	19.1	0	0
2019/03/30 06:35					19.6	18.3	19.2	0	0
2019/03/30 06:40					19.5	18.2	19	0	0
2019/03/30 06:45					19.6	18.1	19.1	0	0
2019/03/30 06:50					19.4	18.2	19.1	0	0
2019/03/30 06:55					19.6	18.2	19.1	0	0
2019/03/30 07:00					19.5	18.2	19.1	0	0
2019/03/30 07:05					19.4	18.2	19	0	0
2019/03/30 07:10					19.5	18.2	19	0	0
2019/03/30 07:15					19.6	18.2	19.1	0	0
2019/03/30 07:20					19.6	18.3	19.1	0	0
2019/03/30 07:25					19.7	18.3	19.1	0	0
2019/03/30 07:30					19.5	18.3	19.1	0	0
2019/03/30 07:35					19.7	18.5	19.1	0	0
2019/03/30 07:40					19.6	18.4	19.1	0	0
2019/03/30 07:45					19.6	18.6	19.3	0	0
2019/03/30 07:50					19.8	18.7	19.3	0	0
2019/03/30 07:55					19.9	18.8	19.4	0	0
2019/03/30 08:00	33.99	28	0	2.98	20	18.8	19.4	0	0
2019/03/30 08:05					19.9	18.8	19.4	0	0
2019/03/30 08:10					19.8	18.7	19.3	0	0
2019/03/30 08:15					19.8	18.8	19.4	0	0
2019/03/30 08:20					19.9	18.9	19.5	0	0
2019/03/30 08:25					19.9	19	19.6	0	0
2019/03/30 08:30					19.9	19.1	19.6	0	0
2019/03/30 08:35					20	19.3	19.8	0	0
2019/03/30 08:40					20.1	19.6	19.8	0	0
2019/03/30 08:45					21.1	20.8	20.6	0	0
2019/03/30 08:50					20.9	21	20.7	0	0
2019/03/30 08:55					20.5	20.6	20.4	0	0
2019/03/30 09:00					20.5	20.4	20.3	0	0
2019/03/30 09:05					20.7	20.5	20.5	0	0
2019/03/30 09:10					20.9	20.7	20.7	0	0
2019/03/30 09:15					21.1	21.3	20.9	0	0
2019/03/30 09:20					21.4	21.9	21.4	0	0
2019/03/30 09:25					21.4	22	21.4	0	0
2019/03/30 09:30					21.5	22.3	21.7	0	0
2019/03/30 09:35					22.6	22.7	22.8	0.01417167	3.3
2019/03/30 09:40					24	23.8	32.6	0.06802889	3.2
2019/03/30 09:45					24.4	24.3	34.4	0.1188611	3.1
2019/03/30 09:50					24.5	24.2	35.3	0.1706222	3.1
2019/03/30 09:55					24.4	24.3	35.4	0.2216561	3.1
2019/03/30 10:00					24.2	23.9	33.9	0.009667777	3.1
2019/03/30 10:05					24.4	23.8	33.1	0.06086667	3.1
2019/03/30 10:10					24.8	23.9	33.3	0.1110206	3
2019/03/30 10:15					24.9	24.4	33.6	0.1621706	3.1
2019/03/30 10:20					25.2	24.6	34.6	0.2138033	3.1
2019/03/30 10:25					25.7	24.7	35.1	0.2653383	3.1
2019/03/30 10:30					25.3	25	35.8	0.3159383	3.1
2019/03/30 10:35					24.8	24.7	36	0.3669294	3
2019/03/30 10:40					25.4	25	36.3	0.4183361	3.1
2019/03/30 10:45					25	24.5	36.1	0.4707083	3.1
2019/03/30 10:50					25.2	24.7	35.1	0.5224817	3.1
2019/03/30 10:55					25.5	25.2	35.3	0.5741267	3.1
2019/03/30 11:00	33.8	24	0	1.37	25.2	25.1	35.2	0.0099	3.1
2019/03/30 11:05					25.4	24.8	34.2	0.06177111	3.1
2019/03/30 11:10					25.4	24.8	34.1	0.1129822	3.1
2019/03/30 11:15					25.8	25.3	34.4	0.1656356	3.1
2019/03/30 11:20					25.9	25.5	34.7	0.2168711	3.1

2019/03/30 11:25					25.1	24.8	34.4	0.2693106	3.1
2019/03/30 11:30					25.8	25.1	33.7	0.3221228	3.2
2019/03/30 11:35					26	25.7	34.5	0.3730283	3.2
2019/03/30 11:40					26.3	26.1	35.9	0.423885	3.2
2019/03/30 11:45					26.5	26.5	37.8	0.4756705	3.2
2019/03/30 11:50					26.4	26.3	37.9	0.5269672	3.1
2019/03/30 11:55					26.3	26	38	0.5784961	3.1
2019/03/30 12:00					25.8	25.8	37.5	0.009533334	3.1
2019/03/30 12:05					26.5	26	37.2	0.06127611	3.1
2019/03/30 12:10					26.5	26.1	37.8	0.1128722	3.1
2019/03/30 12:15					26.7	26.6	38.7	0.1644928	3.1
2019/03/30 12:20					27.1	27	39.7	0.2156245	3.1
2019/03/30 12:25					27.2	27	40.2	0.2674467	3.1
2019/03/30 12:30					27.1	27	40.6	0.3191589	3.2
2019/03/30 12:35					26.7	26.2	40.4	0.3713906	3.2
2019/03/30 12:40					26.7	26.6	39.7	0.4236161	3.2
2019/03/30 12:45					27	26.8	39.1	0.4752244	3.1
2019/03/30 12:50					27	26.5	38.6	0.526625	3.1
2019/03/30 12:55					27.4	26.7	37.5	0.5763267	3.1
2019/03/30 13:00					28.9	27	36.4	0.009374444	3.1
2019/03/30 13:05					28.2	27.3	37.1	0.060995	3.1
2019/03/30 13:10					27.8	27.5	38.5	0.1122	3.1
2019/03/30 13:15					27.9	27.6	39.7	0.1642728	3.1
2019/03/30 13:20					27.7	27.5	40.2	0.2176289	3.1
2019/03/30 13:25					28	27.6	40.6	0.2693411	3.2
2019/03/30 13:30					27.9	27.8	41.4	0.3211817	3.1
2019/03/30 13:35					28.2	28.1	42.4	0.3731078	3.1
2019/03/30 13:40					28.4	28.3	43.2	0.4250278	3.1
2019/03/30 13:45					28.6	28.4	43.5	0.4772778	3.1
2019/03/30 13:50					28.4	28.4	43.5	0.5291	3.1
2019/03/30 13:55					28.7	28.6	43.4	0.5793517	3.1
2019/03/30 14:00	31.21	19	0	1.81	28.1	28.3	43.3	0.009936667	3.1
2019/03/30 14:05					27.9	27.6	41.5	0.06320722	3.1
2019/03/30 14:10					28.6	28.4	39.6	0.1140272	3.1
2019/03/30 14:15					28.5	28.5	39.6	0.1664361	3.2
2019/03/30 14:20					28.6	28.5	39.7	0.2189489	3.1
2019/03/30 14:25					28.7	28.4	38.8	0.2706611	3.1
2019/03/30 14:30					28.6	27.9	38	0.3217561	3.1
2019/03/30 14:35					28.2	27.6	37.2	0.3741711	3.2
2019/03/30 14:40					28.5	27.7	36.9	0.4272645	3.1
2019/03/30 14:45					28.1	27.6	36	0.4784755	3.1
2019/03/30 14:50					28.3	27.5	35.1	0.5303833	3.2
2019/03/30 14:55					28.1	27.2	33.8	0.5829206	3.2
2019/03/30 15:00					28.2	27.1	32.4	0.00968	3.2
2019/03/30 15:05					28.4	27.4	31.9	0.06109889	3.1
2019/03/30 15:10					28.3	27.3	31.7	0.112035	3.1
2019/03/30 15:15					28	27	31.4	0.1435989	3.9
2019/03/30 15:20					28.1	26.8	30.4	0.2036711	3.9
2019/03/30 15:25					28	26.7	29.4	0.2626005	3.6
2019/03/30 15:30					28	26.6	27.8	0.3224528	3.6
2019/03/30 15:35					28.1	26.8	28.1	0.3817428	3.5
2019/03/30 15:40					28.4	26.8	28.1	0.4418822	3.5
2019/03/30 15:45					28.1	26.6	27.7	0.5015817	3.5
2019/03/30 15:50					28.1	26.5	27.5	0.5604683	3.5
2019/03/30 15:55					28.1	26.4	27.5	0.6204061	3.5
2019/03/30 16:00					27.5	26.4	27.4	0.01140944	3.5
2019/03/30 16:05					27.8	26.3	27.3	0.06973389	3.5
2019/03/30 16:10					27.8	26.2	27.3	0.1286572	3.5
2019/03/30 16:15					27.9	26.1	27.2	0.1878433	3.5
2019/03/30 16:20					27.8	26.1	27.2	0.2467972	3.5
2019/03/30 16:25					27.9	26.1	27.3	0.3055861	3.5
2019/03/30 16:30					27.8	26.1	27.2	0.3659761	3.6
2019/03/30 16:35					27.9	26	27.2	0.4252906	3.5
2019/03/30 16:40					27.8	26	27.2	0.4832422	3.5
2019/03/30 16:45					27.8	26	27.1	0.5425811	3.5
2019/03/30 16:50					27.6	26	27.1	0.6006184	3.5
2019/03/30 16:55					28.6	25.9	27	0.6594317	3.5
2019/03/30 17:00	18.51	47	0	2.11	28.7	26	27.1	0.01137889	3.5
2019/03/30 17:05					28.6	26	27.1	0.07017389	3.5
2019/03/30 17:10					27.8	25.9	26.9	0.1292561	3.6
2019/03/30 17:15					27.7	25.8	26.9	0.1888639	3.5
2019/03/30 17:20					27.4	25.8	26.8	0.2475367	3.5
2019/03/30 17:25					27.2	25.7	26.7	0.306075	3.5

2019/03/30 17:30					27.2	25.6	26.6	0.3650411	3.5
2019/03/30 17:35					27.5	25.6	26.6	0.4239767	3.6
2019/03/30 17:40					27.5	25.6	26.6	0.4837494	3.5
2019/03/30 17:45					27.2	25.5	26.6	0.54329	3.5
2019/03/30 17:50					27.2	25.4	26.5	0.6026717	3.5
2019/03/30 17:55					27	25.3	26.4	0.661815	3.5
2019/03/30 18:00					27	25.2	26.3	0.01131778	3.6
2019/03/30 18:05					26.6	25.2	26.3	0.06867667	3.5
2019/03/30 18:10					26	25.1	26.1	0.06867667	0
2019/03/30 18:15					26	25	26.1	0.06867667	0
2019/03/30 18:20					25.9	24.8	26	0.06867667	0
2019/03/30 18:25					25.7	24.7	25.8	0.06867667	0
2019/03/30 18:30					25.6	24.6	25.8	0.06867667	0
2019/03/30 18:35					25.2	24.5	25.7	0.06867667	0
2019/03/30 18:40					25.6	24.5	25.5	0.06867667	0
2019/03/30 18:45					25.2	24	25.4	0.06867667	0
2019/03/30 18:50					25	24	25.2	0.06867667	0
2019/03/30 18:55					24.9	24	25	0.06867667	0
2019/03/30 19:00					24.8	23.8	24.8	0	0
2019/03/30 19:05					24.7	23.9	24.6	0	0
2019/03/30 19:10					24.6	23.7	24.4	0	0
2019/03/30 19:15					24.7	23.7	24.2	0	0
2019/03/30 19:20					24.6	23.7	24	0	0
2019/03/30 19:25					24.3	23.5	23.7	0	0
2019/03/30 19:30					24.5	23.3	23.6	0	0
2019/03/30 19:35					24.3	23.3	23.4	0	0
2019/03/30 19:40					24.3	23.2	23.3	0	0
2019/03/30 19:45					24.3	23.1	23.2	0	0
2019/03/30 19:50					24.5	23.1	23.1	0	0
2019/03/30 19:55					24.5	23	22.8	0	0
2019/03/30 20:00	14.71	45	0	2.96	24.9	23	22.7	0	0
2019/03/30 20:05					24.5	22.9	22.6	0	0
2019/03/30 20:10					24.5	22.9	22.5	0	0
2019/03/30 20:15					24.5	22.9	22.5	0	0
2019/03/30 20:20					24.5	22.9	22.4	0	0
2019/03/30 20:25					24.5	22.8	22.3	0	0
2019/03/30 20:30					24.6	22.8	22.4	0	0
2019/03/30 20:35					24.7	22.6	22.3	0	0
2019/03/30 20:40					24.8	22.3	22.1	0	0
2019/03/30 20:45					24.5	22.2	22.1	0	0
2019/03/30 20:50					24.4	22.3	22	0	0
2019/03/30 20:55					24.5	22.1	22	0	0
2019/03/30 21:00					24.8	21.9	22	0	0
2019/03/30 21:05					24.8	21.8	22.1	0	0
2019/03/30 21:10					24.8	21.8	22.1	0	0
2019/03/30 21:15					24.6	21.9	22	0	0
2019/03/30 21:20					24.5	22	22	0	0
2019/03/30 21:25					24.5	22.2	21.9	0	0
2019/03/30 21:30					24.1	22.3	22	0	0
2019/03/30 21:35					23.9	22.2	22.2	0	0
2019/03/30 21:40					23.9	22.2	22.2	0	0
2019/03/30 21:45					24.3	22.1	22.3	0	0
2019/03/30 21:50					24.4	22	22.1	0	0
2019/03/30 21:55					24.3	21.7	22.1	0	0
2019/03/30 22:00					24.2	21.9	22.4	0	0
2019/03/30 22:05					24.2	21.9	22.4	0	0
2019/03/30 22:10					24.3	21.5	22.4	0	0
2019/03/30 22:15					24.3	21.6	22.3	0	0
2019/03/30 22:20					24.3	21.6	22.4	0	0
2019/03/30 22:25					24.3	21.2	22.5	0	0
2019/03/30 22:30					24.2	21.2	22.5	0	0
2019/03/30 22:35					24.3	21.2	22.3	0	0
2019/03/30 22:40					24	21.6	22.3	0	0
2019/03/30 22:45					23.6	21.5	22.3	0	0
2019/03/30 22:50					24	21	22.2	0	0
2019/03/30 22:55					23.3	21.3	22.2	0	0
2019/03/30 23:00	15.4	58	8	2.56	23.7	21.4	22.2	0	0
2019/03/30 23:05					23.8	21.1	22.3	0	0
2019/03/30 23:10					23.4	21.2	22.3	0	0
2019/03/30 23:15					23.6	21.3	22.2	0	0
2019/03/30 23:20					23.7	21.3	22.2	0	0
2019/03/30 23:25					23.2	21.4	22.2	0	0

Appendix I: Raw sensor data for 2019/04/02

Table I.1 Raw sensor data for 2019/04/02

Date Time	Weather Temperature	Weather Humidity	Weather CloudCover	Weather WindSpeed	Pool Temp	Pool Outside Temp	Pool Heat Temp	PoolPump kWh	PoolPump Current
2019/04/02 00:00					20.8	17.8	18.5	0	0
2019/04/02 00:05					20.8	17.8	18.3	0	0
2019/04/02 00:10					20.8	17.9	18.2	0	0
2019/04/02 00:15					20.8	17.9	18.2	0	0
2019/04/02 00:20					21	18.1	18.3	0	0
2019/04/02 00:25					20.9	18.2	18.2	0	0
2019/04/02 00:30					20.9	18.2	18.2	0	0
2019/04/02 00:35					20.8	18.1	18.2	0	0
2019/04/02 00:40					20.7	18	18.1	0	0
2019/04/02 00:45					20.7	18.1	18.1	0	0
2019/04/02 00:50					20.8	18.2	18.1	0	0
2019/04/02 00:55					20.9	18.1	18.2	0	0
2019/04/02 01:00					20.7	18.2	18.1	0	0
2019/04/02 01:05					20.8	18.1	18.2	0	0
2019/04/02 01:10					20.8	18.1	18.3	0	0
2019/04/02 01:15					20.7	18.1	18.4	0	0
2019/04/02 01:20					20.8	18	18.4	0	0
2019/04/02 01:25					20.8	18	18.4	0	0
2019/04/02 01:30					20.8	18	18.2	0	0
2019/04/02 01:35					20.6	18.1	18.3	0	0
2019/04/02 01:40					20.6	18.1	18.2	0	0
2019/04/02 01:45					20.5	18	18.2	0	0
2019/04/02 01:50					20.5	17.9	18.4	0	0
2019/04/02 01:55					20.5	17.8	18.3	0	0
2019/04/02 02:00	15.01	93	92	2.92	20.5	17.9	18.3	0	0
2019/04/02 02:05					20.5	17.8	18.4	0	0
2019/04/02 02:10					20.3	17.8	18.3	0	0
2019/04/02 02:15					20.4	17.8	18.3	0	0
2019/04/02 02:20					20.3	17.7	18.1	0	0
2019/04/02 02:25					20.3	17.7	18.2	0	0
2019/04/02 02:30					20	17.6	18.1	0	0
2019/04/02 02:35					20.1	17.6	18.1	0	0
2019/04/02 02:40					20.2	17.6	18.2	0	0
2019/04/02 02:45					20.1	17.6	18.2	0	0
2019/04/02 02:50					20.1	17.5	18.1	0	0
2019/04/02 02:55					19.9	17.4	18	0	0
2019/04/02 03:00					19.9	17.4	18.1	0	0
2019/04/02 03:05					19.8	17.3	18.1	0	0
2019/04/02 03:10					19.9	17.2	18.1	0	0
2019/04/02 03:15					19.9	17.2	18.1	0	0
2019/04/02 03:20					19.9	17.2	17.8	0	0
2019/04/02 03:25					19.6	17	17.9	0	0
2019/04/02 03:30					19.8	17.2	17.9	0	0
2019/04/02 03:35					19.9	17.2	17.9	0	0
2019/04/02 03:40					19.5	17	17.9	0	0
2019/04/02 03:45					19.7	17.1	18	0	0
2019/04/02 03:50					19.5	17	17.7	0	0
2019/04/02 03:55					19.5	16.9	17.6	0	0
2019/04/02 04:00					19.7	17	17.9	0	0
2019/04/02 04:05					19.5	17	17.8	0	0
2019/04/02 04:10					19.4	16.8	17.7	0	0
2019/04/02 04:15					19.3	16.8	17.8	0	0
2019/04/02 04:20					19.4	16.8	17.7	0	0
2019/04/02 04:25					19.3	16.8	17.7	0	0
2019/04/02 04:30					19.3	16.8	17.5	0	0
2019/04/02 04:35					19.3	16.7	17.5	0	0
2019/04/02 04:40					19.3	16.7	17.7	0	0
2019/04/02 04:45					19.1	16.6	17.5	0	0
2019/04/02 04:50					19.1	16.6	17.6	0	0
2019/04/02 04:55					19.1	16.6	17.5	0	0
2019/04/02 05:00	18.06	85	56	4.48	19.3	16.6	17.5	0	0
2019/04/02 05:05					19.1	16.6	17.5	0	0
2019/04/02 05:10					19.3	16.6	17.5	0	0
2019/04/02 05:15					19.2	16.6	17.5	0	0

2019/04/02 05:20					19.2	16.6	17.3	0	0
2019/04/02 05:25					19.2	16.6	17.4	0	0
2019/04/02 05:30					19.2	16.5	17.4	0	0
2019/04/02 05:35					19.2	16.6	17.4	0	0
2019/04/02 05:40					19.2	16.5	17.1	0	0
2019/04/02 05:45					19.2	16.5	17.2	0	0
2019/04/02 05:50					19.3	16.5	17.3	0	0
2019/04/02 05:55					19.1	16.5	17.4	0	0
2019/04/02 06:00					19.2	16.6	17.4	0	0
2019/04/02 06:05					18.9	16.6	17.2	0	0
2019/04/02 06:10					19	16.6	17.4	0	0
2019/04/02 06:15					19.2	16.6	17.3	0	0
2019/04/02 06:20					18.9	16.6	17.3	0	0
2019/04/02 06:25					18.9	16.6	17.3	0	0
2019/04/02 06:30					19.1	16.6	16.9	0	0
2019/04/02 06:35					19	16.7	17.2	0	0
2019/04/02 06:40					18.8	16.7	17.4	0	0
2019/04/02 06:45					18.9	16.7	17.4	0	0
2019/04/02 06:50					18.9	16.7	17.4	0	0
2019/04/02 06:55					19	16.7	17.2	0	0
2019/04/02 07:00					19	16.7	17.4	0	0
2019/04/02 07:05					18.8	18.5	17.3	0	0
2019/04/02 07:10					19.2	19.6	17.7	0	0
2019/04/02 07:15					19.3	19.3	17.7	0	0
2019/04/02 07:20					19.2	18.6	18	0	0
2019/04/02 07:25					19.4	18	18.1	0	0
2019/04/02 07:30					19.5	17.9	18.3	0	0
2019/04/02 07:35					19.2	17.9	18.3	0	0
2019/04/02 07:40					19	17.5	18.2	0	0
2019/04/02 07:45					19.2	17.4	18.4	0	0
2019/04/02 07:50					19.2	17.9	18.7	0	0
2019/04/02 07:55					19.4	17.9	18.6	0	0
2019/04/02 08:00	19.96	82	48	2.56	18.9	17.6	18.4	0	0
2019/04/02 08:05					19.1	17.1	18.4	0	0
2019/04/02 08:10					19.1	17.1	18.5	0	0
2019/04/02 08:15					19.1	17.2	18.5	0	0
2019/04/02 08:20					18.9	17.2	18.5	0	0
2019/04/02 08:25					19.1	17.2	18.5	0	0
2019/04/02 08:30					18.8	17.1	18.4	0	0
2019/04/02 08:35					18.9	17.1	18.4	0	0
2019/04/02 08:40					18.8	17.1	18.4	0	0
2019/04/02 08:45					18.7	17	18.4	0	0
2019/04/02 08:50					18.6	16.9	18.2	0	0
2019/04/02 08:55					18.5	17.1	18.1	0	0
2019/04/02 09:00					18.5	17.2	18.1	0	0
2019/04/02 09:05					18.5	17.1	18	0	0
2019/04/02 09:10					18.6	17.2	18	0	0
2019/04/02 09:15					18.7	17.3	18	0	0
2019/04/02 09:20					18.4	17.3	18	0	0
2019/04/02 09:25					18.5	17.2	17.9	0	0
2019/04/02 09:30					20	17.8	20.7	0.04981778	3.7
2019/04/02 09:35					20.6	18.1	21.5	0.1094072	3.6
2019/04/02 09:40					20.6	18.2	21.8	0.170115	3.6
2019/04/02 09:45					21.1	18.5	22	0.2297411	3.6
2019/04/02 09:50					21.2	18.7	22.3	0.2898561	3.6
2019/04/02 09:55					21.2	18.7	22.3	0.3495617	3.6
2019/04/02 10:00					21.3	18.9	22.3	0.01049889	3.6
2019/04/02 10:05					21.3	18.8	22.3	0.07046111	3.5
2019/04/02 10:10					21.3	19	22.3	0.129855	3.6
2019/04/02 10:15					21.5	19.2	22.5	0.1893406	3.5
2019/04/02 10:20					21.5	19.3	22.6	0.2492783	3.6
2019/04/02 10:25					21.6	19.2	22.5	0.307945	3.5
2019/04/02 10:30					21.6	19.3	22.5	0.3682739	3.5
2019/04/02 10:35					21.6	19.4	22.7	0.4275028	3.5
2019/04/02 10:40					21.8	19.4	22.7	0.48697	3.6
2019/04/02 10:45					21.7	19.8	23.1	0.5465167	3.5
2019/04/02 10:50					21.8	19.7	23	0.6060756	3.5
2019/04/02 10:55					22	20.1	23.3	0.665665	3.5
2019/04/02 11:00	26.55	64	20	3.57	22.1	20.5	23.8	0.01111611	3.6
2019/04/02 11:05					22.2	20.5	23.8	0.07099889	3.6
2019/04/02 11:10					22.4	20.9	24.2	0.1309794	3.6
2019/04/02 11:15					22.5	20.9	24.4	0.1903611	3.5
2019/04/02 11:20					22.3	20.6	23.8	0.2507328	3.5

2019/04/02 11:25					22.3	20.4	23.4	0.3102306	3.5
2019/04/02 11:30					22.7	21	24.7	0.3702417	3.5
2019/04/02 11:35					22.7	21.1	24.8	0.4298861	3.6
2019/04/02 11:40					22.8	21.2	24.5	0.4904839	3.6
2019/04/02 11:45					22.8	21	24.4	0.5500733	3.6
2019/04/02 11:50					22.6	20.9	24.2	0.609125	3.6
2019/04/02 11:55					23	21.1	24.1	0.6698756	3.6
2019/04/02 12:00					23.2	21.5	25	0.01091444	3.6
2019/04/02 12:05					23.2	21.7	25.1	0.07066889	3.5
2019/04/02 12:10					23.5	21.7	25	0.1300383	3.6
2019/04/02 12:15					23.5	21.8	25	0.1901228	3.6
2019/04/02 12:20					23.6	21.9	25.2	0.2507939	3.6
2019/04/02 12:25					23.4	22.1	25.5	0.3106278	3.6
2019/04/02 12:30					24.3	22.3	26	0.3698078	3.5
2019/04/02 12:35					25.8	22.2	26	0.42955	3.6
2019/04/02 12:40					23.8	22.1	25.4	0.489335	3.6
2019/04/02 12:45					23.6	21.7	24.5	0.5497494	3.5
2019/04/02 12:50					23.9	22.1	25.4	0.6095284	3.5
2019/04/02 12:55					23.9	22.1	25.4	0.6699611	3.5
2019/04/02 13:00					25	22.5	25.9	0.01073722	3.6
2019/04/02 13:05					24.5	22.6	26.2	0.07126778	3.6
2019/04/02 13:10					24.1	22.2	25.4	0.1314806	3.5
2019/04/02 13:15					24	22.2	25.3	0.1913878	3.6
2019/04/02 13:20					24.1	22.2	25.2	0.25135	3.6
2019/04/02 13:25					24.2	22.4	25.4	0.3107744	3.5
2019/04/02 13:30					23.9	22	25	0.3706205	3.5
2019/04/02 13:35					23.9	21.9	24.7	0.4301856	3.5
2019/04/02 13:40					23.9	21.8	24.7	0.4907222	3.5
2019/04/02 13:45					24	21.9	24.6	0.5506967	3.5
2019/04/02 13:50					24.5	22.6	25.4	0.6105428	3.6
2019/04/02 13:55					24.6	22.7	25.9	0.670725	3.5
2019/04/02 14:00	18.78	83	48	1.31	24.5	22.6	25.7	0.01100611	3.5
2019/04/02 14:05					24.4	22.6	25.6	0.07131667	3.6
2019/04/02 14:10					24.6	22.6	25.6	0.1312361	3.5
2019/04/02 14:15					24.7	22.6	25.6	0.1901044	3.5
2019/04/02 14:20					24.5	22.5	25.2	0.2495228	3.5
2019/04/02 14:25					24.4	22.4	25.2	0.3092222	3.5
2019/04/02 14:30					24.6	22.9	26	0.3688605	3.5
2019/04/02 14:35					24.4	22.8	25.9	0.4276678	3.5
2019/04/02 14:40					24.5	22.7	25.4	0.4876606	3.6
2019/04/02 14:45					24.9	23	26	0.5476167	3.6
2019/04/02 14:50					24.6	22.9	25.8	0.6065767	3.5
2019/04/02 14:55					25	23	25.9	0.666435	3.5
2019/04/02 15:00					24.9	23.3	26.1	0.009936667	3.5
2019/04/02 15:05					25.3	23.5	26.2	0.070345	3.6
2019/04/02 15:10					25	23.3	26	0.1297083	3.5
2019/04/02 15:15					25.4	23.4	26.1	0.1891328	3.5
2019/04/02 15:20					25.2	23.2	25.8	0.2485633	3.5
2019/04/02 15:25					25.2	23.2	26	0.3079144	3.6
2019/04/02 15:30					24.7	22.9	25.3	0.3681944	3.5
2019/04/02 15:35					24.5	22.6	25	0.4280283	3.5
2019/04/02 15:40					24.8	22.6	25.1	0.4879661	3.5
2019/04/02 15:45					24.8	22.6	25	0.5475556	3.5
2019/04/02 15:50					24.8	22.7	25.3	0.6066194	3.5
2019/04/02 15:55					25.4	23	25.8	0.6660805	3.5
2019/04/02 16:00					24.8	23	25.6	0.01085333	3.5
2019/04/02 16:05					24.8	22.7	25	0.07043666	3.5
2019/04/02 16:10					25.1	22.9	25.2	0.1292622	3.5
2019/04/02 16:15					24.8	22.8	25.2	0.1885828	3.5
2019/04/02 16:20					24.7	22.7	25	0.2474328	3.5
2019/04/02 16:25					24.5	22.3	24.5	0.3071689	3.5
2019/04/02 16:30					24.9	22.3	24.6	0.3668439	3.5
2019/04/02 16:35					25.4	22.5	24.9	0.4262194	3.5
2019/04/02 16:40					24.5	22.3	24.6	0.4856439	3.5
2019/04/02 16:45					24.5	22.2	24.5	0.5449278	3.5
2019/04/02 16:50					24.4	22.1	24.4	0.6039245	3.5
2019/04/02 16:55					24.4	22.1	24.3	0.6627744	3.5
2019/04/02 17:00	15.48	84	80	1.16	24.5	22	24.3	0.01104278	3.5
2019/04/02 17:05					24.4	22	24.2	0.06978278	3.5
2019/04/02 17:10					24.5	22	24.3	0.1289811	3.5
2019/04/02 17:15					24.5	22	24.3	0.18777	3.5
2019/04/02 17:20					24.4	22	24.2	0.2467728	3.5
2019/04/02 17:25					24.3	21.8	24.1	0.305195	3.4

2019/04/02 17:30					24.1	21.8	24.1	0.3641122	3.5
2019/04/02 17:35					24.4	21.8	24.1	0.4230783	3.5
2019/04/02 17:40					24.3	21.8	24.2	0.4818733	3.4
2019/04/02 17:45					24.2	21.8	24.1	0.5405033	3.5
2019/04/02 17:50					24.1	21.8	24.1	0.5991945	3.4
2019/04/02 17:55					24.1	21.8	24	0.6582156	3.5
2019/04/02 18:00					24.1	21.7	23.9	0	3.4
2019/04/02 18:05					24	21.7	23.8	0	0
2019/04/02 18:10					23.8	21.6	23.6	0	0
2019/04/02 18:15					23.2	21.5	23.5	0	0
2019/04/02 18:20					22.4	21.3	23.2	0	0
2019/04/02 18:25					21.9	20.7	22.7	0	0
2019/04/02 18:30					21.4	20.3	22.3	0	0
2019/04/02 18:35					21.1	19.9	22	0	0
2019/04/02 18:40					21	19.5	21.8	0	0
2019/04/02 18:45					21	19.3	21.6	0	0
2019/04/02 18:50					21	19.4	21.5	0	0
2019/04/02 18:55					21.1	19.4	21.3	0	0
2019/04/02 19:00					21.1	19.4	21.2	0	0
2019/04/02 19:05					21.2	19.4	21	0	0
2019/04/02 19:10					21.2	19.4	20.9	0	0
2019/04/02 19:15					21.1	19.3	20.9	0	0
2019/04/02 19:20					21	19.2	20.8	0	0
2019/04/02 19:25					20.9	19.2	20.7	0	0
2019/04/02 19:30					20.9	19.2	20.6	0	0
2019/04/02 19:35					21.1	19.2	20.6	0	0
2019/04/02 19:40					21	19	20.4	0	0
2019/04/02 19:45					20.8	18.9	20.3	0	0
2019/04/02 19:50					20.8	18.9	20.3	0	0
2019/04/02 19:55					20.7	18.9	20.2	0	0
2019/04/02 20:00	14.61	93	56	1.52	20.6	18.7	20.1	0	0
2019/04/02 20:05					20.8	18.7	20.1	0	0
2019/04/02 20:10					20.8	18.6	20.1	0	0
2019/04/02 20:15					20.7	18.7	20.1	0	0
2019/04/02 20:20					20.8	18.7	20.1	0	0
2019/04/02 20:25					20.8	18.6	20.1	0	0
2019/04/02 20:30					20.9	18.6	20.1	0	0
2019/04/02 20:35					20.8	18.6	20.1	0	0
2019/04/02 20:40					20.7	18.6	20.1	0	0
2019/04/02 20:45					20.7	18.7	20.2	0	0
2019/04/02 20:50					20.7	18.7	20.2	0	0
2019/04/02 20:55					20.8	18.7	20.1	0	0
2019/04/02 21:00					20.9	18.6	20	0	0
2019/04/02 21:05					20.8	18.5	20	0	0
2019/04/02 21:10					20.4	18.4	19.9	0	0
2019/04/02 21:15					20.4	18.4	20	0	0
2019/04/02 21:20					20.3	18.4	19.9	0	0
2019/04/02 21:25					20.4	18.2	19.6	0	0
2019/04/02 21:30					20.3	18.2	19.6	0	0
2019/04/02 21:35					20.3	18.2	19.6	0	0
2019/04/02 21:40					20	18	19.4	0	0
2019/04/02 21:45					20.2	18	19.4	0	0
2019/04/02 21:50					20.3	18.1	19.6	0	0
2019/04/02 21:55					20.3	18.2	19.6	0	0
2019/04/02 22:00					20.3	18.1	19.5	0	0
2019/04/02 22:05					20.2	18.1	19.5	0	0
2019/04/02 22:10					20.1	18.2	19.5	0	0
2019/04/02 22:15					20.3	18	19.3	0	0
2019/04/02 22:20					20.3	17.8	19.3	0	0
2019/04/02 22:25					20.3	17.9	19.3	0	0
2019/04/02 22:30					20.2	17.9	19.3	0	0
2019/04/02 22:35					20.1	17.9	19.3	0	0
2019/04/02 22:40					20.2	17.9	19.3	0	0
2019/04/02 22:45					19.8	17.9	19.2	0	0
2019/04/02 22:50					19.6	17.9	19.2	0	0
2019/04/02 22:55					19.7	17.9	19.3	0	0
2019/04/02 23:00	14.57	94	56	2.11	19.9	17.8	19.4	0	0
2019/04/02 23:05					20	17.8	19.4	0	0
2019/04/02 23:10					20.1	17.8	19.3	0	0
2019/04/02 23:25					19.8	17.9	19.2	0	0
2019/04/02 23:30					19.5	17.9	19.3	0	0
2019/04/02 23:35					20	17.9	19.2	0	0

Appendix J: Program computing HMM Transition and Emission states

```
Private Sub Hmm_AnalyseProp(SDate As DateTime, EDate As DateTime, RoofPool As String, RainSeason As
  String)
   ' Weather Observed and Roof Hidden
   Dim dsHmm1 As New DataSet
   Dim cmdHmm1 As SqlCeCommand = Nothing
   Dim adpHmm1 As New SqlCeDataAdapter

   'Current
   Dim TransProp_RThRTh, TransProp_RThRTm, TransProp_RThRTl, TransProp_RThTot As Double
   Dim TransProp_RTmRTh, TransProp_RTmRTm, TransProp_RTmRTl, TransProp_RTmTot As Double
   Dim TransProp_RTlRTh, TransProp_RTlRTm, TransProp_RTlRTl, TransProp_RTlTot As Double
   Dim Prop_RTh, Prop_RTm, Prop_RTl, Prop_RTtot As Double

   Dim EmissionProp_ClRTh, EmissionProp_ThRTh, EmissionProp_TmRTh, EmissionProp_TlRTh,
     EmissionProp_RThTot As Double
   Dim EmissionProp_ClRTm, EmissionProp_ThRTm, EmissionProp_TmRTm, EmissionProp_TlRTm,
     EmissionProp_RTmTot As Double
   Dim EmissionProp_ClRTl, EmissionProp_ThRTl, EmissionProp_TmRTl, EmissionProp_TlRTl,
     EmissionProp_RTlTot As Double

   SDate = SDate.Year.ToString & "/" & SDate.Month.ToString & "/" & SDate.Day.ToString & " 00:00"
   EDate = EDate.Year.ToString & "/" & EDate.Month.ToString & "/" & EDate.Day.ToString & " 23:59"

   Dim i As Integer

   Const CloudHumHLSplit As Double = 0.7

   Dim Hidden As String = ""
   Dim PreHidden As String = ""

   TransProp_RThRTh = 0
   TransProp_RThRTm = 0
   TransProp_RThRTl = 0

   TransProp_RTmRTh = 0
   TransProp_RTmRTm = 0
   TransProp_RTmRTl = 0

   TransProp_RTlRTh = 0
   TransProp_RTlRTm = 0
   TransProp_RTlRTl = 0

   Prop_RTh = 0
   Prop_RTm = 0
   Prop_RTl = 0

   EmissionProp_ClRTh = 0
   EmissionProp_ClRTm = 0
   EmissionProp_ClRTl = 0

   EmissionProp_ThRTh = 0
   EmissionProp_ThRTm = 0
   EmissionProp_ThRTl = 0

   EmissionProp_TmRTh = 0
   EmissionProp_TmRTm = 0
   EmissionProp_TmRTl = 0

   EmissionProp_TlRTh = 0
   EmissionProp_TlRTm = 0
   EmissionProp_TlRTl = 0
   Txt_HmmTransitionSequence.Text = ""
   Txt_HmmEmissionSequence.Text = ""


   Using connHmm1 As New SqlCeConnection(SQLConn)
       cmdHmm1 = connHmm1.CreateCommand()
       cmdHmm1.CommandText = "SELECT * FROM SolarHeat  WHERE SolarHeat_Day >= '" & SDate & "' AND
        SolarHeat_Day <= '" & EDate & "'  ORDER By SolarHeat_Day"

       adpHmm1.SelectCommand = cmdHmm1
```

```
adpHmm1.MissingSchemaAction = MissingSchemaAction.AddWithKey ' import Primary Key
adpHmm1.Fill(dsHmm1)
Txt_HmmEmissionProp.Text = ""
Txt_HmmTransitionProp.Text = ""

If dsHmm1.Tables(0).Rows.Count > 1 Then

    For i = 0 To dsHmm1.Tables(0).Rows.Count - 1
        If RoofPool = "Roof" Then
            If dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_RoofTempMax") >= ConstRoofH Then
                ' Roof Temp High
                Hidden = "RTh"
                Prop_RTh += 1
                If (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_HumidityMax") >
                  CloudHumHLSplit) And (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_CloudMax")
                  > CloudHumHLSplit) Then
                    'Ch Ta
                    EmissionProp_ClRTh += 1
                    Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
                     ")ClRTh "
                ElseIf (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_TempMax") >=
                  ConstPredWeatherH) Then
                    'Cl Th
                    EmissionProp_ThRTh += 1
                    Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
                     ")ThRTh "
                ElseIf (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_TempMax") >=
                  ConstPredWeatherM) Then
                    'Cl Tm
                    EmissionProp_TmRTh += 1
                    Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
                     ")TmRTh "
                Else
                    'Cl Tl
                    EmissionProp_TlRTh += 1
                    Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
                     ")TlRTh "
                End If


            ElseIf dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_RoofTempMax") >= ConstRoofM
              Then
                ' Roof Temp Mild
                Hidden = "RTm"
                Prop_RTm += 1
                If (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_HumidityMax") >
                  CloudHumHLSplit) And (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_CloudMax")
                  > CloudHumHLSplit) Then
                    'Ch Ta
                    EmissionProp_ClRTm += 1
                    Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
                     ")ClRTm "
                ElseIf (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_TempMax") >=
                  ConstPredWeatherH) Then
                    'Cl Th
                    EmissionProp_ThRTm += 1
                    Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
                     ")ThRTm "
                ElseIf (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_TempMax") >=
                  ConstPredWeatherM) Then
                    'Cl Tm
                    EmissionProp_TmRTm += 1
                    Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
                     ")TmRTm "
                Else
                    'Cl Tl
                    EmissionProp_TlRTm += 1
                    Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
                     ")TlRTm "
                End If


            Else
                ' Roof Temp Low
                Hidden = "RTl"
                Prop_RTl += 1
                If (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_HumidityMax") >
                  CloudHumHLSplit) And (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_CloudMax")
                  > CloudHumHLSplit) Then
                    'Ch Ta
```

```
            EmissionProp_ClRTl += 1
            Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
             ")ClRTl "
        ElseIf (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_TempMax") >=
          ConstPredWeatherH) Then
            'Cl Th
            EmissionProp_ThRTl += 1
            Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
             ")ThRTl "
        ElseIf (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_TempMax") >=
          ConstPredWeatherM) Then
            'Cl Tm
            EmissionProp_TmRTl += 1
            Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
             ")TmRTl "
        Else
            'Cl Tl
            EmissionProp_TlRTl += 1
            Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
             ")TlRTl "
        End If

    End If

    Txt_HmmTransitionSequence.Text = Txt_HmmTransitionSequence.Text & "(" & i & ")" &
      Hidden & " "

    If PreHidden = "RTh" And Hidden = "RTh" Then TransProp_RThRTh += 1
    If PreHidden = "RTh" And Hidden = "RTm" Then TransProp_RThRTm += 1
    If PreHidden = "RTh" And Hidden = "RTl" Then TransProp_RThRTl += 1


    If PreHidden = "RTm" And Hidden = "RTh" Then TransProp_RTmRTh += 1
    If PreHidden = "RTm" And Hidden = "RTm" Then TransProp_RTmRTm += 1
    If PreHidden = "RTm" And Hidden = "RTl" Then TransProp_RTmRTl += 1

    If PreHidden = "RTl" And Hidden = "RTh" Then TransProp_RTlRTh += 1
    If PreHidden = "RTl" And Hidden = "RTm" Then TransProp_RTlRTm += 1
    If PreHidden = "RTl" And Hidden = "RTl" Then TransProp_RTlRTl += 1
    PreHidden = Hidden


Else ' Pool

    If dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_PoolHeatTemp") >= ConstPoolH Then
        'Pool Temp High
        Hidden = "RTh"
        Prop_RTh += 1
        If (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_HumidityMax") >
          CloudHumHLSplit) And (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_CloudMax")
          > CloudHumHLSplit) Then
            'Ch Ta
            EmissionProp_ClRTh += 1
            Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
             ")ClRTh "
        ElseIf (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_TempMax") >=
          ConstPredWeatherH) Then
            'Cl Th
            EmissionProp_ThRTh += 1
            Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
             ")ThRTh "
        ElseIf (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_TempMax") >=
          ConstPredWeatherM) Then
            'Cl Tm
            EmissionProp_TmRTh += 1
            Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
             ")TmRTh "
        Else
            'Cl Tl
            EmissionProp_TlRTh += 1
            Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
             ")TlRTh "
        End If


    ElseIf dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_PoolHeatTemp") >= ConstPoolM
      Then
        ' Pool Temp Mild
        Hidden = "RTm"
        Prop_RTm += 1
```

```
                If (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_HumidityMax") >
                  CloudHumHLSplit) And (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_CloudMax")
                  > CloudHumHLSplit) Then
                    'Ch Ta
                    EmissionProp_ClRTm += 1
                    Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
                     ")ClRTm "
                ElseIf (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_TempMax") >=
                  ConstPredWeatherH) Then
                    'Cl Th
                    EmissionProp_ThRTm += 1
                    Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
                     ")ThRTm "
                ElseIf (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_TempMax") >=
                  ConstPredWeatherM) Then
                    'Cl Tm
                    EmissionProp_TmRTm += 1
                    Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
                     ")TmRTm "
                Else
                    'Cl Tl
                    EmissionProp_TlRTm += 1
                    Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
                     ")TlRTm "
                End If


            Else ' Pool Heat Low
                Hidden = "RTl"
                Prop_RTl += 1
                If (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_HumidityMax") >
                  CloudHumHLSplit) And (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_CloudMax")
                  > CloudHumHLSplit) Then
                    'Ch Ta
                    EmissionProp_ClRTl += 1
                    Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
                     ")ClRTl "
                ElseIf (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_TempMax") >=
                  ConstPredWeatherH) Then
                    'Cl Th
                    EmissionProp_ThRTl += 1
                    Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
                     ")ThRTl "
                ElseIf (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_TempMax") >=
                  ConstPredWeatherM) Then
                    'Cl Tm
                    EmissionProp_TmRTl += 1
                    Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
                     ")TmRTl "
                Else
                    'Cl Tl
                    EmissionProp_TlRTl += 1
                    Txt_HmmEmissionSequence.Text = Txt_HmmEmissionSequence.Text & "(" & i &
                     ")TlRTl "
                End If

            End If

            Txt_HmmTransitionSequence.Text = Txt_HmmTransitionSequence.Text & "(" & i & ")" &
              Hidden & " "

            If PreHidden = "RTh" And Hidden = "RTh" Then TransProp_RThRTh += 1
            If PreHidden = "RTh" And Hidden = "RTm" Then TransProp_RThRTm += 1
            If PreHidden = "RTh" And Hidden = "RTl" Then TransProp_RThRTl += 1


            If PreHidden = "RTm" And Hidden = "RTh" Then TransProp_RTmRTh += 1
            If PreHidden = "RTm" And Hidden = "RTm" Then TransProp_RTmRTm += 1
            If PreHidden = "RTm" And Hidden = "RTl" Then TransProp_RTmRTl += 1

            If PreHidden = "RTl" And Hidden = "RTh" Then TransProp_RTlRTh += 1
            If PreHidden = "RTl" And Hidden = "RTm" Then TransProp_RTlRTm += 1
            If PreHidden = "RTl" And Hidden = "RTl" Then TransProp_RTlRTl += 1
            PreHidden = Hidden

        End If
    Next
End If
```

```
End Using
adpHmm1.SelectCommand.Dispose()
adpHmm1.Dispose()
dsHmm1.Dispose()


TransProp_RThTot = TransProp_RThRTh + TransProp_RThRTm + TransProp_RThRTl
TransProp_RTmTot = TransProp_RTmRTh + TransProp_RTmRTm + TransProp_RTmRTl
TransProp_RTlTot = TransProp_RTlRTh + TransProp_RTlRTm + TransProp_RTlRTl


TransProp_RThRTh = Math.Round(TransProp_RThRTh / TransProp_RThTot, 2)
TransProp_RThRTm = Math.Round(TransProp_RThRTm / TransProp_RThTot, 2)
TransProp_RThRTl = Math.Round(TransProp_RThRTl / TransProp_RThTot, 2)

TransProp_RTmRTh = Math.Round(TransProp_RTmRTh / TransProp_RTmTot, 2)
TransProp_RTmRTm = Math.Round(TransProp_RTmRTm / TransProp_RTmTot, 2)
TransProp_RTmRTl = Math.Round(TransProp_RTmRTl / TransProp_RTmTot, 2)

TransProp_RTlRTh = Math.Round(TransProp_RTlRTh / TransProp_RTlTot, 2)
TransProp_RTlRTm = Math.Round(TransProp_RTlRTm / TransProp_RTlTot, 2)
TransProp_RTlRTl = Math.Round(TransProp_RTlRTl / TransProp_RTlTot, 2)


Txt_HmmTransitionProp.Text = "RThRTh = " & TransProp_RThRTh & ", "
Txt_HmmTransitionProp.Text = Txt_HmmTransitionProp.Text & "RThRTm = " & TransProp_RThRTm & ", "
Txt_HmmTransitionProp.Text = Txt_HmmTransitionProp.Text & "RThRTl = " & TransProp_RThRTl

Txt_HmmTransitionProp.Text = Txt_HmmTransitionProp.Text & vbCrLf

Txt_HmmTransitionProp.Text = Txt_HmmTransitionProp.Text & "RTmRTh = " & TransProp_RTmRTh & ", "
Txt_HmmTransitionProp.Text = Txt_HmmTransitionProp.Text & "RTmRTm = " & TransProp_RTmRTm & ", "
Txt_HmmTransitionProp.Text = Txt_HmmTransitionProp.Text & "RTmRTl = " & TransProp_RTmRTl
Txt_HmmTransitionProp.Text = Txt_HmmTransitionProp.Text & vbCrLf


Txt_HmmTransitionProp.Text = Txt_HmmTransitionProp.Text & "RTlRTh = " & TransProp_RTlRTh & ", "
Txt_HmmTransitionProp.Text = Txt_HmmTransitionProp.Text & "RTlRTm = " & TransProp_RTlRTm & ", "
Txt_HmmTransitionProp.Text = Txt_HmmTransitionProp.Text & "RTlRTl = " & TransProp_RTlRTl
Txt_HmmTransitionProp.Text = Txt_HmmTransitionProp.Text & vbCrLf

Prop_RTtot = Prop_RTh + Prop_RTm + Prop_RTl
Prop_RTh = Math.Round(Prop_RTh / Prop_RTtot, 2)
Prop_RTm = Math.Round(Prop_RTm / Prop_RTtot, 2)
Prop_RTl = Math.Round(Prop_RTl / Prop_RTtot, 2)


Txt_HmmTransitionProp.Text = Txt_HmmTransitionProp.Text & "P(RTh) = " & Prop_RTh & vbCrLf
Txt_HmmTransitionProp.Text = Txt_HmmTransitionProp.Text & "P(RTm) = " & Prop_RTm & vbCrLf
Txt_HmmTransitionProp.Text = Txt_HmmTransitionProp.Text & "P(RTl) = " & Prop_RTl & vbCrLf


EmissionProp_RThTot = EmissionProp_ClRTh + EmissionProp_ThRTh + EmissionProp_TmRTh +
 EmissionProp_TlRTh
EmissionProp_RTmTot = EmissionProp_ClRTm + EmissionProp_ThRTm + EmissionProp_TmRTm +
 EmissionProp_TlRTm
EmissionProp_RTlTot = EmissionProp_ClRTl + EmissionProp_ThRTl + EmissionProp_TmRTl +
 EmissionProp_TlRTl



EmissionProp_ClRTh = Math.Round(EmissionProp_ClRTh / EmissionProp_RThTot, 2)
EmissionProp_ThRTh = Math.Round(EmissionProp_ThRTh / EmissionProp_RThTot, 2)
EmissionProp_TmRTh = Math.Round(EmissionProp_TmRTh / EmissionProp_RThTot, 2)
EmissionProp_TlRTh = Math.Round(EmissionProp_TlRTh / EmissionProp_RThTot, 2)

EmissionProp_ClRTm = Math.Round(EmissionProp_ClRTm / EmissionProp_RTmTot, 2)
EmissionProp_ThRTm = Math.Round(EmissionProp_ThRTm / EmissionProp_RTmTot, 2)
EmissionProp_TmRTm = Math.Round(EmissionProp_TmRTm / EmissionProp_RTmTot, 2)
EmissionProp_TlRTm = Math.Round(EmissionProp_TlRTm / EmissionProp_RTmTot, 2)


EmissionProp_ClRTl = Math.Round(EmissionProp_ClRTl / EmissionProp_RTlTot, 2)
EmissionProp_ThRTl = Math.Round(EmissionProp_ThRTl / EmissionProp_RTlTot, 2)
EmissionProp_TmRTl = Math.Round(EmissionProp_TmRTl / EmissionProp_RTlTot, 2)
EmissionProp_TlRTl = Math.Round(EmissionProp_TlRTl / EmissionProp_RTlTot, 2)

Txt_HmmEmissionProp.Text = "EmClRTh = " & EmissionProp_ClRTh & ", "
Txt_HmmEmissionProp.Text = Txt_HmmEmissionProp.Text & "EmThRTh = " & EmissionProp_ThRTh & ", "
```

```
Txt_HmmEmissionProp.Text = Txt_HmmEmissionProp.Text & "EmTmRTh = " & EmissionProp_TmRTh & ", "
Txt_HmmEmissionProp.Text = Txt_HmmEmissionProp.Text & "EmTlRTh = " & EmissionProp_TlRTh
Txt_HmmEmissionProp.Text = Txt_HmmEmissionProp.Text & vbCrLf


Txt_HmmEmissionProp.Text = Txt_HmmEmissionProp.Text & "EmClRTm = " & EmissionProp_ClRTm & ", "
Txt_HmmEmissionProp.Text = Txt_HmmEmissionProp.Text & "EmThRTm = " & EmissionProp_ThRTm & ", "
Txt_HmmEmissionProp.Text = Txt_HmmEmissionProp.Text & "EmTmRTm = " & EmissionProp_TmRTm & ", "
Txt_HmmEmissionProp.Text = Txt_HmmEmissionProp.Text & "EmTlRTm = " & EmissionProp_TlRTm
Txt_HmmEmissionProp.Text = Txt_HmmEmissionProp.Text & vbCrLf


Txt_HmmEmissionProp.Text = Txt_HmmEmissionProp.Text & "EmClRTl = " & EmissionProp_ClRTl & ", "
Txt_HmmEmissionProp.Text = Txt_HmmEmissionProp.Text & "EmThRTl = " & EmissionProp_ThRTl & ", "
Txt_HmmEmissionProp.Text = Txt_HmmEmissionProp.Text & "EmTmRTl = " & EmissionProp_TmRTl & ", "
Txt_HmmEmissionProp.Text = Txt_HmmEmissionProp.Text & "EmTlRTl = " & EmissionProp_TlRTl
Txt_HmmEmissionProp.Text = Txt_HmmEmissionProp.Text & vbCrLf

If RoofPool = "Roof" Then ' Record new values
    If RainSeason = "H" Then
        TransPropWeatherRainH_RThRTh = TransProp_RThRTh
        TransPropWeatherRainH_RThRTm = TransProp_RThRTm
        TransPropWeatherRainH_RThRTl = TransProp_RThRTl

        TransPropWeatherRainH_RTmRTh = TransProp_RTmRTh
        TransPropWeatherRainH_RTmRTm = TransProp_RTmRTm
        TransPropWeatherRainH_RTmRTl = TransProp_RTmRTl

        TransPropWeatherRainH_RTlRTh = TransProp_RTlRTh
        TransPropWeatherRainH_RTlRTm = TransProp_RTlRTm
        TransPropWeatherRainH_RTlRTl = TransProp_RTlRTl

        PropWeatherRainH_RTh = Prop_RTh
        PropWeatherRainH_RTm = Prop_RTm
        PropWeatherRainH_RTl = Prop_RTl

        EmissionPropWeatherRainH_ClRTh = EmissionProp_ClRTh
        EmissionPropWeatherRainH_ClRTm = EmissionProp_ClRTm
        EmissionPropWeatherRainH_ClRTl = EmissionProp_ClRTl

        EmissionPropWeatherRainH_ThRTh = EmissionProp_ThRTh
        EmissionPropWeatherRainH_ThRTm = EmissionProp_ThRTm
        EmissionPropWeatherRainH_ThRTl = EmissionProp_ThRTl

        EmissionPropWeatherRainH_TmRTh = EmissionProp_TmRTh
        EmissionPropWeatherRainH_TmRTm = EmissionProp_TmRTm
        EmissionPropWeatherRainH_TmRTl = EmissionProp_TmRTl

        EmissionPropWeatherRainH_TlRTh = EmissionProp_TlRTh
        EmissionPropWeatherRainH_TlRTm = EmissionProp_TlRTm
        EmissionPropWeatherRainH_TlRTl = EmissionProp_TlRTl

    ElseIf RainSeason = "L" Then
        TransPropWeatherRainL_RThRTh = TransProp_RThRTh
        TransPropWeatherRainL_RThRTm = TransProp_RThRTm
        TransPropWeatherRainL_RThRTl = TransProp_RThRTl

        TransPropWeatherRainL_RTmRTh = TransProp_RTmRTh
        TransPropWeatherRainL_RTmRTm = TransProp_RTmRTm
        TransPropWeatherRainL_RTmRTl = TransProp_RTmRTl

        TransPropWeatherRainL_RTlRTh = TransProp_RTlRTh
        TransPropWeatherRainL_RTlRTm = TransProp_RTlRTm
        TransPropWeatherRainL_RTlRTl = TransProp_RTlRTl

        PropWeatherRainL_RTh = Prop_RTh
        PropWeatherRainL_RTm = Prop_RTm
        PropWeatherRainL_RTl = Prop_RTl

        EmissionPropWeatherRainL_ClRTh = EmissionProp_ClRTh
        EmissionPropWeatherRainL_ClRTm = EmissionProp_ClRTm
        EmissionPropWeatherRainL_ClRTl = EmissionProp_ClRTl

        EmissionPropWeatherRainL_ThRTh = EmissionProp_ThRTh
        EmissionPropWeatherRainL_ThRTm = EmissionProp_ThRTm
        EmissionPropWeatherRainL_ThRTl = EmissionProp_ThRTl

        EmissionPropWeatherRainL_TmRTh = EmissionProp_TmRTh
        EmissionPropWeatherRainL_TmRTm = EmissionProp_TmRTm
        EmissionPropWeatherRainL_TmRTl = EmissionProp_TmRTl
```

```
        EmissionPropWeatherRainL_TlRTh = EmissionProp_TlRTh
        EmissionPropWeatherRainL_TlRTm = EmissionProp_TlRTm
        EmissionPropWeatherRainL_TlRTl = EmissionProp_TlRTl

    ElseIf RainSeason = "N" Then
        TransPropWeatherRainN_RThRTh = TransProp_RThRTh
        TransPropWeatherRainN_RThRTm = TransProp_RThRTm
        TransPropWeatherRainN_RThRTl = TransProp_RThRTl

        TransPropWeatherRainN_RTmRTh = TransProp_RTmRTh
        TransPropWeatherRainN_RTmRTm = TransProp_RTmRTm
        TransPropWeatherRainN_RTmRTl = TransProp_RTmRTl

        TransPropWeatherRainN_RTlRTh = TransProp_RTlRTh
        TransPropWeatherRainN_RTlRTm = TransProp_RTlRTm
        TransPropWeatherRainN_RTlRTl = TransProp_RTlRTl

        PropWeatherRainN_RTh = Prop_RTh
        PropWeatherRainN_RTm = Prop_RTm
        PropWeatherRainN_RTl = Prop_RTl

        EmissionPropWeatherRainN_ClRTh = EmissionProp_ClRTh
        EmissionPropWeatherRainN_ClRTm = EmissionProp_ClRTm
        EmissionPropWeatherRainN_ClRTl = EmissionProp_ClRTl

        EmissionPropWeatherRainN_ThRTh = EmissionProp_ThRTh
        EmissionPropWeatherRainN_ThRTm = EmissionProp_ThRTm
        EmissionPropWeatherRainN_ThRTl = EmissionProp_ThRTl

        EmissionPropWeatherRainN_TmRTh = EmissionProp_TmRTh
        EmissionPropWeatherRainN_TmRTm = EmissionProp_TmRTm
        EmissionPropWeatherRainN_TmRTl = EmissionProp_TmRTl

        EmissionPropWeatherRainN_TlRTh = EmissionProp_TlRTh
        EmissionPropWeatherRainN_TlRTm = EmissionProp_TlRTm
        EmissionPropWeatherRainN_TlRTl = EmissionProp_TlRTl
    End If

Else ' Pool
    If RainSeason = "H" Then
        TransPropPoolRainH_RThRTh = TransProp_RThRTh
        TransPropPoolRainH_RThRTm = TransProp_RThRTm
        TransPropPoolRainH_RThRTl = TransProp_RThRTl

        TransPropPoolRainH_RTmRTh = TransProp_RTmRTh
        TransPropPoolRainH_RTmRTm = TransProp_RTmRTm
        TransPropPoolRainH_RTmRTl = TransProp_RTmRTl

        TransPropPoolRainH_RTlRTh = TransProp_RTlRTh
        TransPropPoolRainH_RTlRTm = TransProp_RTlRTm
        TransPropPoolRainH_RTlRTl = TransProp_RTlRTl

        PropPoolRainH_RTh = Prop_RTh
        PropPoolRainH_RTm = Prop_RTm
        PropPoolRainH_RTl = Prop_RTl

        EmissionPropPoolRainH_ClRTh = EmissionProp_ClRTh
        EmissionPropPoolRainH_ClRTm = EmissionProp_ClRTm
        EmissionPropPoolRainH_ClRTl = EmissionProp_ClRTl

        EmissionPropPoolRainH_ThRTh = EmissionProp_ThRTh
        EmissionPropPoolRainH_ThRTm = EmissionProp_ThRTm
        EmissionPropPoolRainH_ThRTl = EmissionProp_ThRTl

        EmissionPropPoolRainH_TmRTh = EmissionProp_TmRTh
        EmissionPropPoolRainH_TmRTm = EmissionProp_TmRTm
        EmissionPropPoolRainH_TmRTl = EmissionProp_TmRTl

        EmissionPropPoolRainH_TlRTh = EmissionProp_TlRTh
        EmissionPropPoolRainH_TlRTm = EmissionProp_TlRTm
        EmissionPropPoolRainH_TlRTl = EmissionProp_TlRTl

    ElseIf RainSeason = "L" Then
        TransPropPoolRainL_RThRTh = TransProp_RThRTh
        TransPropPoolRainL_RThRTm = TransProp_RThRTm
        TransPropPoolRainL_RThRTl = TransProp_RThRTl

        TransPropPoolRainL_RTmRTh = TransProp_RTmRTh
        TransPropPoolRainL_RTmRTm = TransProp_RTmRTm
```

```
            TransPropPoolRainL_RTmRTl = TransProp_RTmRTl

            TransPropPoolRainL_RTlRTh = TransProp_RTlRTh
            TransPropPoolRainL_RTlRTm = TransProp_RTlRTm
            TransPropPoolRainL_RTlRTl = TransProp_RTlRTl

            PropPoolRainL_RTh = Prop_RTh
            PropPoolRainL_RTm = Prop_RTm
            PropPoolRainL_RTl = Prop_RTl

            EmissionPropPoolRainL_ClRTh = EmissionProp_ClRTh
            EmissionPropPoolRainL_ClRTm = EmissionProp_ClRTm
            EmissionPropPoolRainL_ClRTl = EmissionProp_ClRTl

            EmissionPropPoolRainL_ThRTh = EmissionProp_ThRTh
            EmissionPropPoolRainL_ThRTm = EmissionProp_ThRTm
            EmissionPropPoolRainL_ThRTl = EmissionProp_ThRTl

            EmissionPropPoolRainL_TmRTh = EmissionProp_TmRTh
            EmissionPropPoolRainL_TmRTm = EmissionProp_TmRTm
            EmissionPropPoolRainL_TmRTl = EmissionProp_TmRTl

            EmissionPropPoolRainL_TlRTh = EmissionProp_TlRTh
            EmissionPropPoolRainL_TlRTm = EmissionProp_TlRTm
            EmissionPropPoolRainL_TlRTl = EmissionProp_TlRTl

        ElseIf RainSeason = "N" Then
            TransPropPoolRainN_RThRTh = TransProp_RThRTh
            TransPropPoolRainN_RThRTm = TransProp_RThRTm
            TransPropPoolRainN_RThRTl = TransProp_RThRTl

            TransPropPoolRainN_RTmRTh = TransProp_RTmRTh
            TransPropPoolRainN_RTmRTm = TransProp_RTmRTm
            TransPropPoolRainN_RTmRTl = TransProp_RTmRTl

            TransPropPoolRainN_RTlRTh = TransProp_RTlRTh
            TransPropPoolRainN_RTlRTm = TransProp_RTlRTm
            TransPropPoolRainN_RTlRTl = TransProp_RTlRTl

            PropPoolRainN_RTh = Prop_RTh
            PropPoolRainN_RTm = Prop_RTm
            PropPoolRainN_RTl = Prop_RTl

            EmissionPropPoolRainN_ClRTh = EmissionProp_ClRTh
            EmissionPropPoolRainN_ClRTm = EmissionProp_ClRTm
            EmissionPropPoolRainN_ClRTl = EmissionProp_ClRTl

            EmissionPropPoolRainN_ThRTh = EmissionProp_ThRTh
            EmissionPropPoolRainN_ThRTm = EmissionProp_ThRTm
            EmissionPropPoolRainN_ThRTl = EmissionProp_ThRTl

            EmissionPropPoolRainN_TmRTh = EmissionProp_TmRTh
            EmissionPropPoolRainN_TmRTm = EmissionProp_TmRTm
            EmissionPropPoolRainN_TmRTl = EmissionProp_TmRTl

            EmissionPropPoolRainN_TlRTh = EmissionProp_TlRTh
            EmissionPropPoolRainN_TlRTm = EmissionProp_TlRTm
            EmissionPropPoolRainN_TlRTl = EmissionProp_TlRTl
        End If
    End If
End Sub
```

Appendix K: Program computing Viterbi HMM Five Day predictions and compare against historical recordings

```
Private Sub Hmm_Viterbi(SDate As DateTime, PropHigh As Double, PropMild As Double, PropLow As Double,
                            EmHCl As Double, EmHTh As Double, EmHTm As Double, EmHTl As Double,
                            EmMCl As Double, EmMTh As Double, EmMTm As Double, EmMTl As Double,
                            EmLCl As Double, EmLTh As Double, EmLTm As Double, EmLTl As Double,
                            TrHH As Double, TrHM As Double, TrHL As Double,
                            TrMH As Double, TrMM As Double, TrML As Double,
                            TrLH As Double, TrLM As Double, TrLL As Double,
                            RoofPool As String)
        'Prediction 5day Roof Temperature and Pool Temp

        Const ConstRoofH As Double = 0.6
        Const ConstRoofM As Double = 0.4

        Const ConstPredWeatherH As Double = 0.6
        Const ConstPredWeatherM As Double = 0.4

        Const ConstPoolH As Double = 0.75
        Const ConstPoolM As Double = 0.55

        Const ConstPredWeatherPoolH As Double = 0.5
        Const ConstPredWeatherPoolM As Double = 0.35

        Const CloudHumHLSplit As Double = 0.7


        Dim Edate, DispDate As DateTime
        Edate = SDate.AddDays(5)
        SDate = SDate.Year.ToString & "/" & SDate.Month.ToString & "/" & SDate.Day.ToString & " 00:00"
        Edate = Edate.Year.ToString & "/" & Edate.Month.ToString & "/" & Edate.Day.ToString & " 23:59"


        Dim WeatherCond As String
        Dim ViterbiBestHigh(5) As Double
        Dim ViterbiBestMild(5) As Double
        Dim ViterbiBestLow(5) As Double
        Dim ViterbiRoofTemp(5) As String
        Dim ViterbiPoolTemp(5) As String

        Dim TempHigh As Double = 0
        Dim TempMild As Double = 0
        Dim TempLow As Double = 0


        Dim dsHmm1 As New DataSet
        Dim cmdHmm1 As SqlCeCommand = Nothing
        Dim adpHmm1 As New SqlCeDataAdapter


        Using connHmm1 As New SqlCeConnection(SQLConn)
            cmdHmm1 = connHmm1.CreateCommand()
            cmdHmm1.CommandText = "SELECT * FROM SolarHeat  WHERE SolarHeat_Day >= '" & SDate & "' AND
             SolarHeat_Day <= '" & Edate & "'  ORDER By SolarHeat_Day"

            adpHmm1.SelectCommand = cmdHmm1
            adpHmm1.MissingSchemaAction = MissingSchemaAction.AddWithKey ' import Primary Key
            adpHmm1.Fill(dsHmm1)

            If dsHmm1.Tables(0).Rows.Count > 1 Then
                For i = 0 To dsHmm1.Tables(0).Rows.Count - 1

                    DispDate = dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Day")


                    Select Case RoofPool
                        Case "Roof"

                               If (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_HumidityMax") >
                                     CloudHumHLSplit) And
                                     (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_CloudMax") >
                                     CloudHumHLSplit) Then
                                   'Ch Ta
```

```
                WeatherCond = "Cl"
            ElseIf (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_TempMax") >=
              ConstPredWeatherH) Then
                'Cl Th
                WeatherCond = "H"
            ElseIf (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_TempMax") >=
              ConstPredWeatherM) Then
                'Cl Tm
                WeatherCond = "M"
            Else
                'Cl Tl
                WeatherCond = "L"
            End If

         If dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_RoofTempMax") >= ConstRoofH
          Then
             ViterbiRoofTemp(i) = "H"
         ElseIf dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_RoofTempMax") >= ConstRoofM
          Then
             ViterbiRoofTemp(i) = "M"
         Else
             ViterbiRoofTemp(i) = "L"
         End If

     Case ("Pool")
            If (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_HumidityMax") >
                  CloudHumHLSplit) And
                  (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_CloudMax") >
                  CloudHumHLSplit) Then
                'Ch Ta
                WeatherCond = "Cl"
            ElseIf (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_TempMax") >=
              ConstPredWeatherPoolH) Then
                'Cl Th
                WeatherCond = "H"
            ElseIf (dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_TempMax") >=
              ConstPredWeatherPoolM) Then
                'Cl Tm
                WeatherCond = "M"
            Else
                'Cl Tl
                WeatherCond = "L"
            End If


         If dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_PoolHeatTemp") >= ConstPoolH
          Then
             ViterbiRoofTemp(i) = "H"
         ElseIf dsHmm1.Tables(0).Rows(i).Item("SolarHeat_Num_PoolHeatTemp ") >=
          ConstPoolM Then
             ViterbiRoofTemp(i) = "M"
         Else
             ViterbiRoofTemp(i) = "L"
         End If
End Select



Select Case WeatherCond
    Case "Cl"
       If i = 0 Then
            ViterbiBestHigh(i) = PropHigh * EmHCl
            ViterbiBestMild(i) = PropMild * EmMCl
            ViterbiBestLow(i) = PropLow * EmLCl
        Else
            ViterbiBestHigh(i) = ViterbiBestHigh(i - 1) * TrHH * EmHCl
            TempHigh = ViterbiBestMild(i - 1) * TrMH * EmHCl
            If TempHigh > ViterbiBestHigh(i) Then ViterbiBestHigh(i) = TempHigh
            TempHigh = ViterbiBestLow(i - 1) * TrLH * EmHCl
            If TempHigh > ViterbiBestHigh(i) Then ViterbiBestHigh(i) = TempHigh

            ViterbiBestMild(i) = ViterbiBestHigh(i - 1) * TrHM * EmMCl
            TempMild = ViterbiBestMild(i - 1) * TrMM * EmMCl
            If TempMild > ViterbiBestMild(i) Then ViterbiBestMild(i) = TempMild
            TempMild = ViterbiBestLow(i - 1) * TrLM * EmMCl
            If TempMild > ViterbiBestMild(i) Then ViterbiBestMild(i) = TempMild

            ViterbiBestLow(i) = ViterbiBestHigh(i - 1) * TrHL * EmLCl
            TempLow = ViterbiBestMild(i - 1) * TrML * EmLCl
            If TempLow > ViterbiBestLow(i) Then ViterbiBestLow(i) = TempLow
```

```
            TempLow = ViterbiBestLow(i - 1) * TrLL * EmLCl
            If TempLow > ViterbiBestLow(i) Then ViterbiBestLow(i) = TempLow
        End If

    Case "H"
        If i = 0 Then
            ViterbiBestHigh(i) = PropHigh * EmHTh
            ViterbiBestMild(i) = PropMild * EmMTh
            ViterbiBestLow(i) = PropLow * EmLTh
        Else
            ViterbiBestHigh(i) = ViterbiBestHigh(i - 1) * TrHH * EmHTh
            TempHigh = ViterbiBestMild(i - 1) * TrMH * EmHTh
            If TempHigh > ViterbiBestHigh(i) Then ViterbiBestHigh(i) = TempHigh
            TempHigh = ViterbiBestLow(i - 1) * TrLH * EmHTh
            If TempHigh > ViterbiBestHigh(i) Then ViterbiBestHigh(i) = TempHigh

            ViterbiBestMild(i) = ViterbiBestHigh(i - 1) * TrHM * EmMTh
            TempMild = ViterbiBestMild(i - 1) * TrMM * EmMTh
            If TempMild > ViterbiBestMild(i) Then ViterbiBestMild(i) = TempMild
            TempMild = ViterbiBestLow(i - 1) * TrLM * EmMTh
            If TempMild > ViterbiBestLow(i) Then ViterbiBestMild(i) = TempMild

            ViterbiBestLow(i) = ViterbiBestHigh(i - 1) * TrHL * EmLTh
            TempLow = ViterbiBestMild(i - 1) * TrML * EmLTh
            If TempLow > ViterbiBestLow(i) Then ViterbiBestLow(i) = TempLow
            TempLow = ViterbiBestLow(i - 1) * TrLL * EmLTh
            If TempLow > ViterbiBestLow(i) Then ViterbiBestLow(i) = TempLow
        End If

    Case "M"
        If i = 0 Then
            ViterbiBestHigh(i) = PropHigh * EmHTm
            ViterbiBestMild(i) = PropMild * EmMTm
            ViterbiBestLow(i) = PropLow * EmLTm
        Else
            ViterbiBestHigh(i) = ViterbiBestHigh(i - 1) * TrHH * EmHTm
            TempHigh = ViterbiBestMild(i - 1) * TrMH * EmHTm
            If TempHigh > ViterbiBestHigh(i) Then ViterbiBestHigh(i) = TempHigh
            TempHigh = ViterbiBestLow(i - 1) * TrLH * EmHTm
            If TempHigh > ViterbiBestHigh(i) Then ViterbiBestHigh(i) = TempHigh

            ViterbiBestMild(i) = ViterbiBestHigh(i - 1) * TrHM * EmMTm
            TempMild = ViterbiBestMild(i - 1) * TrMM * EmMTm
            If TempMild > ViterbiBestMild(i) Then ViterbiBestMild(i) = TempMild
            TempMild = ViterbiBestLow(i - 1) * TrLM * EmMTm
            If TempMild > ViterbiBestMild(i) Then ViterbiBestMild(i) = TempMild

            ViterbiBestLow(i) = ViterbiBestHigh(i - 1) * TrHL * EmLTm
            TempLow = ViterbiBestMild(i - 1) * TrML * EmLTm
            If TempLow > ViterbiBestLow(i) Then ViterbiBestLow(i) = TempLow
            TempLow = ViterbiBestLow(i - 1) * TrLL * EmLTm
            If TempLow > ViterbiBestLow(i) Then ViterbiBestLow(i) = TempLow
        End If

    Case "L"
        If i = 0 Then
            ViterbiBestHigh(i) = PropHigh * EmHTl
            ViterbiBestMild(i) = PropMild * EmMTl
            ViterbiBestLow(i) = PropLow * EmLTl
        Else
            ViterbiBestHigh(i) = ViterbiBestHigh(i - 1) * TrHH * EmHTl
            TempHigh = ViterbiBestMild(i - 1) * TrMH * EmHTl
            If TempHigh > ViterbiBestHigh(i) Then ViterbiBestHigh(i) = TempHigh
            TempHigh = ViterbiBestLow(i - 1) * TrLH * EmHTl
            If TempHigh > ViterbiBestHigh(i) Then ViterbiBestHigh(i) = TempHigh

            ViterbiBestMild(i) = ViterbiBestHigh(i - 1) * TrHM * EmMTl
            TempMild = ViterbiBestMild(i - 1) * TrMM * EmMTl
            If TempMild > ViterbiBestMild(i) Then ViterbiBestMild(i) = TempMild
            TempMild = ViterbiBestLow(i - 1) * TrLM * EmMTl
            If TempMild > ViterbiBestMild(i) Then ViterbiBestMild(i) = TempMild

            ViterbiBestLow(i) = ViterbiBestHigh(i - 1) * TrHL * EmLTl
            TempLow = ViterbiBestMild(i - 1) * TrML * EmLTl
            If TempLow > ViterbiBestLow(i) Then ViterbiBestLow(i) = TempLow
            TempLow = ViterbiBestLow(i - 1) * TrLL * EmLTl
            If TempLow > ViterbiBestLow(i) Then ViterbiBestLow(i) = TempLow
        End If

End Select
```

```
ViterbiBestHigh(i) = Math.Round(ViterbiBestHigh(i), 6)
ViterbiBestMild(i) = Math.Round(ViterbiBestMild(i), 6)
ViterbiBestLow(i) = Math.Round(ViterbiBestLow(i), 6)

Select Case i
    Case 0
        Lb_Viterbi_Day1_H.ForeColor = Color.Red
        Lb_Viterbi_Day1_M.ForeColor = Color.Red
        Lb_Viterbi_Day1_L.ForeColor = Color.Red
        If (ViterbiBestHigh(i) > ViterbiBestMild(i)) And (ViterbiBestHigh(i) >
         ViterbiBestLow(i)) Then Lb_Viterbi_Day1_H.ForeColor = Color.Green
        If (ViterbiBestMild(i) > ViterbiBestHigh(i)) And (ViterbiBestMild(i) >
         ViterbiBestLow(i)) Then Lb_Viterbi_Day1_M.ForeColor = Color.Green
        If (ViterbiBestLow(i) > ViterbiBestHigh(i)) And (ViterbiBestLow(i) >
         ViterbiBestMild(i)) Then Lb_Viterbi_Day1_L.ForeColor = Color.Green

        Lb_Viterbi_Day1_Date.Text = DispDate.Year.ToString & "/" &
         DispDate.Month.ToString & "/" & DispDate.Day.ToString
        Lb_Viterbi_Day1_Cond.Text = WeatherCond
        Lb_Viterbi_Day1_H.Text = ViterbiBestHigh(i)
        Lb_Viterbi_Day1_M.Text = ViterbiBestMild(i)
        Lb_Viterbi_Day1_L.Text = ViterbiBestLow(i)
        Lb_Viterbi_Day1_Record.Text = ViterbiRoofTemp(i)


    Case 1
        Lb_Viterbi_Day2_H.ForeColor = Color.Red
        Lb_Viterbi_Day2_M.ForeColor = Color.Red
        Lb_Viterbi_Day2_L.ForeColor = Color.Red
        If (ViterbiBestHigh(i) > ViterbiBestMild(i)) And (ViterbiBestHigh(i) >
         ViterbiBestLow(i)) Then Lb_Viterbi_Day2_H.ForeColor = Color.Green
        If (ViterbiBestMild(i) > ViterbiBestHigh(i)) And (ViterbiBestMild(i) >
         ViterbiBestLow(i)) Then Lb_Viterbi_Day2_M.ForeColor = Color.Green
        If (ViterbiBestLow(i) > ViterbiBestHigh(i)) And (ViterbiBestLow(i) >
         ViterbiBestMild(i)) Then Lb_Viterbi_Day2_L.ForeColor = Color.Green

        Lb_Viterbi_Day2_Date.Text = DispDate.Year.ToString & "/" &
         DispDate.Month.ToString & "/" & DispDate.Day.ToString
        Lb_Viterbi_Day2_Cond.Text = WeatherCond
        Lb_Viterbi_Day2_H.Text = ViterbiBestHigh(i)
        Lb_Viterbi_Day2_M.Text = ViterbiBestMild(i)
        Lb_Viterbi_Day2_L.Text = ViterbiBestLow(i)
        Lb_Viterbi_Day2_Record.Text = ViterbiRoofTemp(i)

    Case 2
        Lb_Viterbi_Day3_H.ForeColor = Color.Red
        Lb_Viterbi_Day3_M.ForeColor = Color.Red
        Lb_Viterbi_Day3_L.ForeColor = Color.Red
        If (ViterbiBestHigh(i) > ViterbiBestMild(i)) And (ViterbiBestHigh(i) >
         ViterbiBestLow(i)) Then Lb_Viterbi_Day3_H.ForeColor = Color.Green
        If (ViterbiBestMild(i) > ViterbiBestHigh(i)) And (ViterbiBestMild(i) >
         ViterbiBestLow(i)) Then Lb_Viterbi_Day3_M.ForeColor = Color.Green
        If (ViterbiBestLow(i) > ViterbiBestHigh(i)) And (ViterbiBestLow(i) >
         ViterbiBestMild(i)) Then Lb_Viterbi_Day3_L.ForeColor = Color.Green

        Lb_Viterbi_Day3_Date.Text = DispDate.Year.ToString & "/" &
         DispDate.Month.ToString & "/" & DispDate.Day.ToString
        Lb_Viterbi_Day3_Cond.Text = WeatherCond
        Lb_Viterbi_Day3_H.Text = ViterbiBestHigh(i)
        Lb_Viterbi_Day3_M.Text = ViterbiBestMild(i)
        Lb_Viterbi_Day3_L.Text = ViterbiBestLow(i)
        Lb_Viterbi_Day3_Record.Text = ViterbiRoofTemp(i)

    Case 3
        Lb_Viterbi_Day4_H.ForeColor = Color.Red
        Lb_Viterbi_Day4_M.ForeColor = Color.Red
        Lb_Viterbi_Day4_L.ForeColor = Color.Red
        If (ViterbiBestHigh(i) > ViterbiBestMild(i)) And (ViterbiBestHigh(i) >
         ViterbiBestLow(i)) Then Lb_Viterbi_Day4_H.ForeColor = Color.Green
        If (ViterbiBestMild(i) > ViterbiBestHigh(i)) And (ViterbiBestMild(i) >
         ViterbiBestLow(i)) Then Lb_Viterbi_Day4_M.ForeColor = Color.Green
        If (ViterbiBestLow(i) > ViterbiBestHigh(i)) And (ViterbiBestLow(i) >
         ViterbiBestMild(i)) Then Lb_Viterbi_Day4_L.ForeColor = Color.Green

        Lb_Viterbi_Day4_Date.Text = DispDate.Year.ToString & "/" &
         DispDate.Month.ToString & "/" & DispDate.Day.ToString
        Lb_Viterbi_Day4_Cond.Text = WeatherCond
        Lb_Viterbi_Day4_H.Text = ViterbiBestHigh(i)
        Lb_Viterbi_Day4_M.Text = ViterbiBestMild(i)
```

```
                    Lb_Viterbi_Day4_L.Text = ViterbiBestLow(i)
                    Lb_Viterbi_Day4_Record.Text = ViterbiRoofTemp(i)

                Case 4
                    Lb_Viterbi_Day5_H.ForeColor = Color.Red
                    Lb_Viterbi_Day5_M.ForeColor = Color.Red
                    Lb_Viterbi_Day5_L.ForeColor = Color.Red
                    If (ViterbiBestHigh(i) > ViterbiBestMild(i)) And (ViterbiBestHigh(i) >
                     ViterbiBestLow(i)) Then Lb_Viterbi_Day5_H.ForeColor = Color.Green
                    If (ViterbiBestMild(i) > ViterbiBestHigh(i)) And (ViterbiBestMild(i) >
                     ViterbiBestLow(i)) Then Lb_Viterbi_Day5_M.ForeColor = Color.Green
                    If (ViterbiBestLow(i) > ViterbiBestHigh(i)) And (ViterbiBestLow(i) >
                     ViterbiBestMild(i)) Then Lb_Viterbi_Day5_L.ForeColor = Color.Green

                    Lb_Viterbi_Day5_Date.Text = DispDate.Year.ToString & "/" &
                     DispDate.Month.ToString & "/" & DispDate.Day.ToString
                    Lb_Viterbi_Day5_Cond.Text = WeatherCond
                    Lb_Viterbi_Day5_H.Text = ViterbiBestHigh(i)
                    Lb_Viterbi_Day5_M.Text = ViterbiBestMild(i)
                    Lb_Viterbi_Day5_L.Text = ViterbiBestLow(i)
                    Lb_Viterbi_Day5_Record.Text = ViterbiRoofTemp(i)
            End Select
        Next
    End If

End Using
adpHmm1.SelectCommand.Dispose()
adpHmm1.Dispose()
dsHmm1.Dispose()

End Sub
```

Appendix L: Program training Pool HMM model with Forward, Backwards and Baum-Welch algorithm

Note: Reset values to Zero and display commands was remove

```
Private Sub HmmBw_LearnAndPredictPool(SDate As DateTime, EDate As DateTime)

  Dim dsHmm1 As New DataSet
  Dim cmdHmm1 As SqlCeCommand = Nothing
  Dim adpHmm1 As New SqlCeDataAdapter
  Dim ObsPathWeather, ObsPathPool, PoolTempPred, PoolTempPredRec As String
  Dim ObsPathWeatherTemp, ObsPathPoolTemp As String
  Dim i, j As Integer
  Dim EmCH, EmCM, EmCL, EmHH, EmHM, EmHL, EmMH, EmMM, EmML, EmLH, EmLM, EmLL As Integer
  Dim Norm As Double
  Dim HmmDays As Integer = 14
  Dim PredCorrect As String
  Dim PredCorrectTot, PredBorderCorrectTot, PredIncorrectTot As Double

  SDate = SDate.Year.ToString & "/" & SDate.Month.ToString & "/" & SDate.Day.ToString & " 00:00"
  EDate = EDate.Year.ToString & "/" & EDate.Month.ToString & "/" & EDate.Day.ToString & " 23:59"

  Using connHmm1 As New SqlCeConnection(SQLConn)
      cmdHmm1 = connHmm1.CreateCommand()
      cmdHmm1.CommandText = "SELECT * FROM SolarHeat  WHERE SolarHeat_Day >= '" & SDate & "' AND
       SolarHeat_Day <= '" & EDate & "'  ORDER By SolarHeat_Day"

      adpHmm1.SelectCommand = cmdHmm1
      adpHmm1.MissingSchemaAction = MissingSchemaAction.AddWithKey ' import Primary Key
      adpHmm1.Fill(dsHmm1)
      Txt_HmmEmissionProp.Text = ""
      Txt_HmmTransitionProp.Text = ""


      If dsHmm1.Tables(0).Rows.Count > HmmDays Then

          For i = HmmDays To dsHmm1.Tables(0).Rows.Count - 2

              'Retrieve previous hmmdays day
              For j = i - HmmDays To i - 1
                  If (dsHmm1.Tables(0).Rows(j).Item("SolarHeat_Num_HumidityMax") > CloudHumHLSplit)
                    And (dsHmm1.Tables(0).Rows(j).Item("SolarHeat_Num_CloudMax") > CloudHumHLSplit)
                    Then
                      ObsPathWeather = ObsPathWeather & "Cl,"
                      ObsPathWeatherTemp = "Cl"
                  ElseIf (dsHmm1.Tables(0).Rows(j).Item("SolarHeat_Num_TempMax") >=
                    ConstPredWeatherH) Then
                      ObsPathWeather = ObsPathWeather & "H,"
                      ObsPathWeatherTemp = "H"
                  ElseIf (dsHmm1.Tables(0).Rows(j).Item("SolarHeat_Num_TempMax") >=
                    ConstPredWeatherM) Then
                      ObsPathWeather = ObsPathWeather & "M,"
                      ObsPathWeatherTemp = "M"
                  Else
                      ObsPathWeather = ObsPathWeather & "L,"
                      ObsPathWeatherTemp = "L"
                  End If


                  ' Recorded only past  Pool conditions - Mod1-1
                  If dsHmm1.Tables(0).Rows(j).Item("SolarHeat_Num_PoolHeatTemp") >= ConstPoolH Then
                      ObsPathPool = ObsPathPool & "H,"
                      ObsPathPoolTemp = "H"
                      If ObsPathWeatherTemp = "C" Then EmCH += 1
                      If ObsPathWeatherTemp = "H" Then EmHH += 1
                      If ObsPathWeatherTemp = "M" Then EmMH += 1
                      If ObsPathWeatherTemp = "L" Then EmLH += 1

                  ElseIf dsHmm1.Tables(0).Rows(j).Item("SolarHeat_Num_PoolHeatTemp") >= ConstPoolM
                    Then
                      ObsPathPool = ObsPathPool & "M,"
                      ObsPathPoolTemp = "M"
                      If ObsPathWeatherTemp = "C" Then EmCM += 1
                      If ObsPathWeatherTemp = "H" Then EmHM += 1
```

```
            If ObsPathWeatherTemp = "M" Then EmMM += 1
            If ObsPathWeatherTemp = "L" Then EmLM += 1
        Else
            ObsPathPool = ObsPathPool & "L,"
            ObsPathPoolTemp = "L"
            If ObsPathWeatherTemp = "C" Then EmCL += 1
            If ObsPathWeatherTemp = "H" Then EmHL += 1
            If ObsPathWeatherTemp = "M" Then EmML += 1
            If ObsPathWeatherTemp = "L" Then EmLL += 1
        End If
    Next j

    1 predicted days weather status no pool condition must still occure
    If (dsHmm1.Tables(0).Rows(j).Item("SolarHeat_Num_HumidityMax") > CloudHumHLSplit) And
         (dsHmm1.Tables(0).Rows(j).Item("SolarHeat_Num_CloudMax") > CloudHumHLSplit) Then
        ObsPathWeather = ObsPathWeather & "Cl,"
        ObsPathWeatherTemp = "Cl"
    ElseIf (dsHmm1.Tables(0).Rows(j).Item("SolarHeat_Num_TempMax") >=
         ConstPredWeatherPoolH) Then
        ObsPathWeather = ObsPathWeather & "H,"
        ObsPathWeatherTemp = "H"
    ElseIf (dsHmm1.Tables(0).Rows(j).Item("SolarHeat_Num_TempMax") >=
         ConstPredWeatherPoolM) Then
        ObsPathWeather = ObsPathWeather & "M,"
        ObsPathWeatherTemp = "M"
    Else
        ObsPathWeather = ObsPathWeather & "L,"
        ObsPathWeatherTemp = "L"
    End If


   ' Recorded past Pool condition only valid if validate historical data
    If dsHmm1.Tables(0).Rows(j).Item("SolarHeat_Num_PoolHeatTemp") >= ConstPoolH Then
        PoolTempPredRec = PoolTempPredRec & "H,"
    ElseIf dsHmm1.Tables(0).Rows(j).Item("SolarHeat_Num_PoolHeatTemp") >= ConstPoolM Then
        PoolTempPredRec = PoolTempPredRec & "M,"
    Else
        PoolTempPredRec = PoolTempPredRec & "L,"
    End If

    'Remove last ","
    ObsPathWeather = Microsoft.VisualBasic.Left(ObsPathWeather,
      Microsoft.VisualBasic.Len(ObsPathWeather) - 1)
    ObsPathPool = Microsoft.VisualBasic.Left(ObsPathPool,
      Microsoft.VisualBasic.Len(ObsPathPool) - 1)
    PoolTempPredRec = Microsoft.VisualBasic.Left(PoolTempPredRec,
      Microsoft.VisualBasic.Len(PoolTempPredRec) - 1)


    'Learn
    BaumWelchPool(ObsPathWeather)

   'Mod1-2
    Dim hmmdlastdays As Boolean = True

    If hmmdlastdays = True Then
        BwEmissionPropPool_ClPHh = Math.Round((BwEmissionPropPool_ClPHh * ShareOld) + (EmCH
          / 15 * ShareNew), 2)
        BwEmissionPropPool_ThPHh = Math.Round((BwEmissionPropPool_ThPHh * ShareOld) + (EmHH
          / 15 * ShareNew), 2)
        BwEmissionPropPool_TmPHh = Math.Round((BwEmissionPropPool_TmPHh * ShareOld) + (EmMH
          / 15 * ShareNew), 2)
        BwEmissionPropPool_TlPHh = Math.Round((BwEmissionPropPool_TlPHh * ShareOld) + (EmLH
          / 15 * ShareNew), 2)
        Norm = BwEmissionPropPool_ClPHh + BwEmissionPropPool_ThPHh +
          BwEmissionPropPool_TmPHh + BwEmissionPropPool_TlPHh
        BwEmissionPropPool_ClPHh = Math.Round(BwEmissionPropPool_ClPHh / Norm, 2)
        BwEmissionPropPool_ThPHh = Math.Round(BwEmissionPropPool_ThPHh / Norm, 2)
        BwEmissionPropPool_TmPHh = Math.Round(BwEmissionPropPool_TmPHh / Norm, 2)
        BwEmissionPropPool_TlPHh = Math.Round(BwEmissionPropPool_TlPHh / Norm, 2)

        BwEmissionPropPool_ClPHm = Math.Round((BwEmissionPropPool_ClPHm * ShareOld) + (EmCM
          / 15 * ShareNew), 2)
        BwEmissionPropPool_ThPHm = Math.Round((BwEmissionPropPool_ThPHm * ShareOld) + (EmHM
          / 15 * ShareNew), 2)
        BwEmissionPropPool_TmPHm = Math.Round((BwEmissionPropPool_TmPHm * ShareOld) + (EmMM
          / 15 * ShareNew), 2)
        BwEmissionPropPool_TlPHm = Math.Round((BwEmissionPropPool_TlPHm * ShareOld) + (EmLM
          / 15 * ShareNew), 2)
```

```
                    Norm = BwEmissionPropPool_ClPHm + BwEmissionPropPool_ThPHm +
                      BwEmissionPropPool_TmPHm + BwEmissionPropPool_TlPHm
                    BwEmissionPropPool_ClPHm = Math.Round(BwEmissionPropPool_ClPHm / Norm, 2)
                    BwEmissionPropPool_ThPHm = Math.Round(BwEmissionPropPool_ThPHm / Norm, 2)
                    BwEmissionPropPool_TmPHm = Math.Round(BwEmissionPropPool_TmPHm / Norm, 2)
                    BwEmissionPropPool_TlPHm = Math.Round(BwEmissionPropPool_TlPHm / Norm, 2)


                    BwEmissionPropPool_ClPHl = Math.Round((BwEmissionPropPool_ClPHl * ShareOld) + (EmCL
                      / 15 * ShareNew), 2)
                    BwEmissionPropPool_ThPHl = Math.Round((BwEmissionPropPool_ThPHl * ShareOld) + (EmHL
                      / 15 * ShareNew), 2)
                    BwEmissionPropPool_TmPHl = Math.Round((BwEmissionPropPool_TmPHl * ShareOld) + (EmML
                      / 15 * ShareNew), 2)
                    BwEmissionPropPool_TlPHl = Math.Round((BwEmissionPropPool_TlPHl * ShareOld) + (EmLL
                      / 15 * ShareNew), 2)
                    Norm = BwEmissionPropPool_ClPHl + BwEmissionPropPool_ThPHl +
                      BwEmissionPropPool_TmPHl + BwEmissionPropPool_TlPHl
                    BwEmissionPropPool_ClPHl = Math.Round(BwEmissionPropPool_ClPHl / Norm, 2)
                    BwEmissionPropPool_ThPHl = Math.Round(BwEmissionPropPool_ThPHl / Norm, 2)
                    BwEmissionPropPool_TmPHl = Math.Round(BwEmissionPropPool_TmPHl / Norm, 2)
                    BwEmissionPropPool_TlPHl = Math.Round(BwEmissionPropPool_TlPHl / Norm, 2)
                End If


                PoolTempPred = HmmBw_ViterbiPoolPath(ObsPathWeather, ObsPathPool)

                HmmBw_UpdatePoolDiagramValues()

                If PoolTempPredRec = Microsoft.VisualBasic.Right(PoolTempPred, 1) Then
                    PredCorrect = "C"
                    PredCorrectTot += 1
                ElseIf (PoolTempPredRec = "M") And Microsoft.VisualBasic.Right(PoolTempPred, 1) <> "M"
                  Then
                    PredCorrect = "B"
                    PredBorderCorrectTot += 1
                ElseIf (PoolTempPredRec <> "M") And (Microsoft.VisualBasic.Right(PoolTempPred, 1) =
                  "M") Then
                    PredCorrect = "B"
                    PredBorderCorrectTot += 1
                Else
                    PredCorrect = "I"
                    PredIncorrectTot += 1
                End If


            Next i
        End If

    End Using
    adpHmm1.SelectCommand.Dispose()
    adpHmm1.Dispose()
    dsHmm1.Dispose()

End Sub


Private Sub BaumWelchPool(ObservationPath As String)
    Dim i, j As Integer
    Dim PathHmm(60) As String

    Dim BwTempPropPool_PHh, BwTempPropPool_PHm, BwTempPropPool_PHl As Double

    Dim TempHigh, TempMild, TempLow As Double

    Dim BwTempTransPropPool_PHhPHh, BwTempTransPropPool_PHhPHm, BwTempTransPropPool_PHhPHl As Double
    Dim BwTempTransPropPool_PHhPHh_Top, BwTempTransPropPool_PHhPHm_Top, BwTempTransPropPool_PHhPHl_Top
     As Double
    Dim BwTempTransPropPool_PHh_Bottom As Double

    Dim BwTempTransPropPool_PHmPHh, BwTempTransPropPool_PHmPHm, BwTempTransPropPool_PHmPHl As Double
    Dim BwTempTransPropPool_PHmPHh_Top, BwTempTransPropPool_PHmPHm_Top, BwTempTransPropPool_PHmPHl_Top
     As Double
    Dim BwTempTransPropPool_PHm_Bottom As Double

    Dim BwTempTransPropPool_PHlPHh, BwTempTransPropPool_PHlPHm, BwTempTransPropPool_PHlPHl As Double
    Dim BwTempTransPropPool_PHlPHh_Top, BwTempTransPropPool_PHlPHm_Top, BwTempTransPropPool_PHlPHl_Top
     As Double
    Dim BwTempTransPropPool_PHl_Bottom As Double
```

```
Dim BwTempEmissionPropPool_ClPHh, BwTempEmissionPropPool_ClPHm, BwTempEmissionPropPool_ClPHl As
 Double
Dim BwTempEmissionPropPool_ClPHh_Top, BwTempEmissionPropPool_ClPHm_Top,
 BwTempEmissionPropPool_ClPHl_Top As Double

Dim BwTempEmissionPropPool_ThPHh, BwTempEmissionPropPool_ThPHm, BwTempEmissionPropPool_ThPHl As
 Double
Dim BwTempEmissionPropPool_ThPHh_Top, BwTempEmissionPropPool_ThPHm_Top,
 BwTempEmissionPropPool_ThPHl_Top As Double

Dim BwTempEmissionPropPool_TmPHh, BwTempEmissionPropPool_TmPHm, BwTempEmissionPropPool_TmPHl As
 Double
Dim BwTempEmissionPropPool_TmPHh_Top, BwTempEmissionPropPool_TmPHm_Top,
 BwTempEmissionPropPool_TmPHl_Top As Double

Dim BwTempEmissionPropPool_TlPHh, BwTempEmissionPropPool_TlPHm, BwTempEmissionPropPool_TlPHl As
 Double
Dim BwTempEmissionPropPool_TlPHh_Top, BwTempEmissionPropPool_TlPHm_Top,
 BwTempEmissionPropPool_TlPHl_Top As Double

Dim BwTempEmissionPropPool_PHh_Bottom, BwTempEmissionPropPool_PHm_Bottom,
 BwTempEmissionPropPool_PHl_Bottom As Double



PathHmm = Split(ObservationPath, ",")

Hmm_Fwd("n", BwPropPool_PHh, BwPropPool_PHm, BwPropPool_PHl, BwEmissionPropPool_ClPHh,
 BwEmissionPropPool_ThPHh, BwEmissionPropPool_TmPHh, BwEmissionPropPool_TlPHh,
            BwEmissionPropPool_ClPHm, BwEmissionPropPool_ThPHm, BwEmissionPropPool_TmPHm,
 BwEmissionPropPool_TlPHm, BwEmissionPropPool_ClPHl, BwEmissionPropPool_ThPHl,
 BwEmissionPropPool_TmPHl, BwEmissionPropPool_TlPHl,
            BwTransPropPool_PHhPHh, BwTransPropPool_PHhPHm, BwTransPropPool_PHhPHl,
 BwTransPropPool_PHmPHh, BwTransPropPool_PHmPHm, BwTransPropPool_PHmPHl, BwTransPropPool_PHlPHh,
 BwTransPropPool_PHlPHm, BwTransPropPool_PHlPHl, "Pool", ObservationPath)

Hmm_Bwd("n", BwPropPool_PHh, BwPropPool_PHm, BwPropPool_PHl, BwEmissionPropPool_ClPHh,
 BwEmissionPropPool_ThPHh, BwEmissionPropPool_TmPHh, BwEmissionPropPool_TlPHh,
            BwEmissionPropPool_ClPHm, BwEmissionPropPool_ThPHm, BwEmissionPropPool_TmPHm,
 BwEmissionPropPool_TlPHm, BwEmissionPropPool_ClPHl, BwEmissionPropPool_ThPHl,
 BwEmissionPropPool_TmPHl, BwEmissionPropPool_TlPHl,
            BwTransPropPool_PHhPHh, BwTransPropPool_PHhPHm, BwTransPropPool_PHhPHl,
 BwTransPropPool_PHmPHh, BwTransPropPool_PHmPHm, BwTransPropPool_PHmPHl, BwTransPropPool_PHlPHh,
 BwTransPropPool_PHlPHm, BwTransPropPool_PHlPHl, "Pool", ObservationPath)
'Bck 1st value at 1 but Fwd at 0

'Baum-Welch - Start
BwTempPropPool_PHh = Math.Round((HmmFwdHigh(0) * HmmBwdHigh(1)) / HmmBwdLikelihood, 2)
BwTempPropPool_PHm = Math.Round((HmmFwdMild(0) * HmmBwdMild(1)) / HmmBwdLikelihood, 2)
BwTempPropPool_PHl = Math.Round((HmmFwdLow(0) * HmmBwdLow(1)) / HmmBwdLikelihood, 2)

'Mod 2-1
'Can never be 0, Test High
If BwTempPropPool_PHh = 0 Then
    If BwTempPropPool_PHm >= BwTempPropPool_PHl Then
        BwTempPropPool_PHm -= 0.01
        BwTempPropPool_PHh += 0.01
    Else
        BwTempPropPool_PHl -= 0.01
        BwTempPropPool_PHh += 0.01
    End If
End If

'Test Medium
If BwTempPropPool_PHm = 0 Then
    If BwTempPropPool_PHh >= BwTempPropPool_PHl Then
        BwTempPropPool_PHh -= 0.01
        BwTempPropPool_PHm += 0.01
    Else
        BwTempPropPool_PHl -= 0.01
        BwTempPropPool_PHm += 0.01
    End If
End If

'Test Low
If BwTempPropPool_PHl = 0 Then
    If BwTempPropPool_PHh >= BwTempPropPool_PHm Then
        BwTempPropPool_PHh -= 0.01
        BwTempPropPool_PHl += 0.01
    Else
```

```
        BwTempPropPool_PHm -= 0.01
        BwTempPropPool_PHl += 0.01
    End If
End If

'Baum-Welch - Transision

For i = 0 To PathHmm.Length - 2 ' T - 1
    j = i + 1
    Select Case PathHmm(i + 1)
        Case "Cl"
            TempHigh = BwEmissionPropPool_ClPHh
            TempMild = BwEmissionPropPool_ClPHm
            TempLow = BwEmissionPropPool_ClPHl
        Case "H"
            TempHigh = BwEmissionPropPool_ThPHh
            TempMild = BwEmissionPropPool_ThPHm
            TempLow = BwEmissionPropPool_ThPHl
        Case "M"
            TempHigh = BwEmissionPropPool_TmPHh
            TempMild = BwEmissionPropPool_TmPHm
            TempLow = BwEmissionPropPool_TmPHl
        Case "L"
            TempHigh = BwEmissionPropPool_TlPHh
            TempMild = BwEmissionPropPool_TlPHm
            TempLow = BwEmissionPropPool_TlPHl
    End Select

    BwTempTransPropPool_PHhPHh_Top = BwTempTransPropPool_PHhPHh_Top + (HmmFwdHigh(i) *
     BwTransPropPool_PHhPHh * TempHigh * HmmBwdHigh(j + 1))
    BwTempTransPropPool_PHhPHm_Top = BwTempTransPropPool_PHhPHm_Top + (HmmFwdHigh(i) *
     BwTransPropPool_PHhPHm * TempMild * HmmBwdMild(j + 1))
    BwTempTransPropPool_PHhPHl_Top = BwTempTransPropPool_PHhPHl_Top + (HmmFwdHigh(i) *
     BwTransPropPool_PHhPHl * TempLow * HmmBwdLow(j + 1))
    BwTempTransPropPool_PHh_Bottom = BwTempTransPropPool_PHh_Bottom + (HmmFwdHigh(i) *
     HmmBwdHigh(j))


    BwTempTransPropPool_PHmPHh_Top = BwTempTransPropPool_PHmPHh_Top + (HmmFwdMild(i) *
     BwTransPropPool_PHmPHh * TempHigh * HmmBwdHigh(j + 1))
    BwTempTransPropPool_PHmPHm_Top = BwTempTransPropPool_PHmPHm_Top + (HmmFwdMild(i) *
     BwTransPropPool_PHmPHm * TempMild * HmmBwdMild(j + 1))
    BwTempTransPropPool_PHmPHl_Top = BwTempTransPropPool_PHmPHl_Top + (HmmFwdMild(i) *
     BwTransPropPool_PHmPHl * TempLow * HmmBwdLow(j + 1))
    BwTempTransPropPool_PHm_Bottom = BwTempTransPropPool_PHm_Bottom + (HmmFwdMild(i) *
     HmmBwdMild(j))

    BwTempTransPropPool_PHlPHh_Top = BwTempTransPropPool_PHlPHh_Top + (HmmFwdLow(i) *
     BwTransPropPool_PHlPHh * TempHigh * HmmBwdHigh(j + 1))
    BwTempTransPropPool_PHlPHm_Top = BwTempTransPropPool_PHlPHm_Top + (HmmFwdLow(i) *
     BwTransPropPool_PHlPHm * TempMild * HmmBwdMild(j + 1))
    BwTempTransPropPool_PHlPHl_Top = BwTempTransPropPool_PHlPHl_Top + (HmmFwdLow(i) *
     BwTransPropPool_PHlPHl * TempLow * HmmBwdLow(j + 1))
    BwTempTransPropPool_PHl_Bottom = BwTempTransPropPool_PHl_Bottom + (HmmFwdLow(i) * HmmBwdLow(j))

Next i

BwTempTransPropPool_PHhPHh = Math.Round(BwTempTransPropPool_PHhPHh_Top /
 BwTempTransPropPool_PHh_Bottom, 2)
BwTempTransPropPool_PHhPHm = Math.Round(BwTempTransPropPool_PHhPHm_Top /
 BwTempTransPropPool_PHh_Bottom, 2)
BwTempTransPropPool_PHhPHl = Math.Round(BwTempTransPropPool_PHhPHl_Top /
 BwTempTransPropPool_PHh_Bottom, 2)

BwTempTransPropPool_PHmPHh = Math.Round(BwTempTransPropPool_PHmPHh_Top /
 BwTempTransPropPool_PHm_Bottom, 2)
BwTempTransPropPool_PHmPHm = Math.Round(BwTempTransPropPool_PHmPHm_Top /
 BwTempTransPropPool_PHm_Bottom, 2)
BwTempTransPropPool_PHmPHl = Math.Round(BwTempTransPropPool_PHmPHl_Top /
 BwTempTransPropPool_PHm_Bottom, 2)

BwTempTransPropPool_PHlPHh = Math.Round(BwTempTransPropPool_PHlPHh_Top /
 BwTempTransPropPool_PHl_Bottom, 2)
BwTempTransPropPool_PHlPHm = Math.Round(BwTempTransPropPool_PHlPHm_Top /
 BwTempTransPropPool_PHl_Bottom, 2)
BwTempTransPropPool_PHlPHl = Math.Round(BwTempTransPropPool_PHlPHl_Top /
 BwTempTransPropPool_PHl_Bottom, 2)

'Mod 2-2
'Can never be 0 (High), Test High
```

```
If BwTempTransPropPool_PHhPHh = 0 Then
    If BwTempTransPropPool_PHhPHm >= BwTempTransPropPool_PHhPHl Then
        BwTempTransPropPool_PHhPHm -= 0.01
        BwTempTransPropPool_PHhPHh += 0.01
    Else
        BwTempTransPropPool_PHhPHl -= 0.01
        BwTempTransPropPool_PHhPHh += 0.01
    End If
End If

'Test Mild
If BwTempTransPropPool_PHhPHm = 0 Then
    If BwTempTransPropPool_PHhPHh >= BwTempTransPropPool_PHhPHl Then
        BwTempTransPropPool_PHhPHh -= 0.01
        BwTempTransPropPool_PHhPHm += 0.01
    Else
        BwTempTransPropPool_PHhPHl -= 0.01
        BwTempTransPropPool_PHhPHm += 0.01
    End If
End If

'Test Low
If BwTempTransPropPool_PHhPHl = 0 Then
    If BwTempTransPropPool_PHhPHh >= BwTempTransPropPool_PHhPHm Then
        BwTempTransPropPool_PHhPHh -= 0.01
        BwTempTransPropPool_PHhPHl += 0.01
    Else
        BwTempTransPropPool_PHhPHm -= 0.01
        BwTempTransPropPool_PHhPHl += 0.01
    End If
End If

'Can never be 0 (Mild), Test High
If BwTempTransPropPool_PHmPHh = 0 Then
    If BwTempTransPropPool_PHmPHm >= BwTempTransPropPool_PHmPHl Then
        BwTempTransPropPool_PHmPHm -= 0.01
        BwTempTransPropPool_PHmPHh += 0.01
    Else
        BwTempTransPropPool_PHmPHl -= 0.01
        BwTempTransPropPool_PHmPHh += 0.01
    End If
End If

'Test Mild
If BwTempTransPropPool_PHmPHm = 0 Then
    If BwTempTransPropPool_PHmPHh >= BwTempTransPropPool_PHmPHl Then
        BwTempTransPropPool_PHmPHh -= 0.01
        BwTempTransPropPool_PHmPHm += 0.01
    Else
        BwTempTransPropPool_PHmPHl -= 0.01
        BwTempTransPropPool_PHmPHm += 0.01
    End If
End If


'Test Low
If BwTempTransPropPool_PHmPHl = 0 Then
    If BwTempTransPropPool_PHmPHh >= BwTempTransPropPool_PHmPHm Then
        BwTempTransPropPool_PHmPHh -= 0.01
        BwTempTransPropPool_PHmPHl += 0.01
    Else
        BwTempTransPropPool_PHmPHm -= 0.01
        BwTempTransPropPool_PHmPHl += 0.01
    End If
End If

'Can never be 0 (Low) , Test High
If BwTempTransPropPool_PHlPHh = 0 Then
    If BwTempTransPropPool_PHlPHm >= BwTempTransPropPool_PHlPHl Then
        BwTempTransPropPool_PHlPHm -= 0.01
        BwTempTransPropPool_PHlPHh += 0.01
    Else
        BwTempTransPropPool_PHlPHl -= 0.01
        BwTempTransPropPool_PHlPHh += 0.01
    End If
End If

'Test Mild
If BwTempTransPropPool_PHlPHm = 0 Then
    If BwTempTransPropPool_PHlPHh >= BwTempTransPropPool_PHlPHl Then
```

```
        BwTempTransPropPool_PHlPHh -= 0.01
        BwTempTransPropPool_PHlPHm += 0.01
    Else
        BwTempTransPropPool_PHlPHl -= 0.01
        BwTempTransPropPool_PHlPHm += 0.01
    End If
End If


'Test Low
If BwTempTransPropPool_PHlPHl = 0 Then
    If BwTempTransPropPool_PHlPHh >= BwTempTransPropPool_PHlPHm Then
        BwTempTransPropPool_PHlPHh -= 0.01
        BwTempTransPropPool_PHlPHl += 0.01
    Else
        BwTempTransPropPool_PHlPHm -= 0.01
        BwTempTransPropPool_PHlPHl += 0.01
    End If
End If

BwPropPool_PHh = BwTempPropPool_PHh
BwPropPool_PHm = BwTempPropPool_PHm
BwPropPool_PHl = BwTempPropPool_PHl

BwTransPropPool_PHhPHh = BwTempTransPropPool_PHhPHh
BwTransPropPool_PHhPHm = BwTempTransPropPool_PHhPHm
BwTransPropPool_PHhPHl = BwTempTransPropPool_PHhPHl

BwTransPropPool_PHmPHh = BwTempTransPropPool_PHmPHh
BwTransPropPool_PHmPHm = BwTempTransPropPool_PHmPHm
BwTransPropPool_PHmPHl = BwTempTransPropPool_PHmPHl

BwTransPropPool_PHlPHh = BwTempTransPropPool_PHlPHh
BwTransPropPool_PHlPHm = BwTempTransPropPool_PHlPHm
BwTransPropPool_PHlPHl = BwTempTransPropPool_PHlPHl


'Baum-Welch - Emmision


For i = 0 To PathHmm.Length - 1 ' T - 1
    j = i + 1

    Select Case PathHmm(i)
        Case "Cl"
            BwTempEmissionPropPool_ClPHh_Top = BwTempEmissionPropPool_ClPHh_Top + (HmmFwdHigh(i) *
              HmmBwdHigh(j))
            BwTempEmissionPropPool_ClPHm_Top = BwTempEmissionPropPool_ClPHm_Top + (HmmFwdMild(i) *
              HmmBwdMild(j))
            BwTempEmissionPropPool_ClPHl_Top = BwTempEmissionPropPool_ClPHl_Top + (HmmFwdLow(i) *
              HmmBwdLow(j))
        Case "H"
            BwTempEmissionPropPool_ThPHh_Top = BwTempEmissionPropPool_ThPHh_Top + (HmmFwdHigh(i) *
              HmmBwdHigh(j))
            BwTempEmissionPropPool_ThPHm_Top = BwTempEmissionPropPool_ThPHm_Top + (HmmFwdMild(i) *
              HmmBwdMild(j))
            BwTempEmissionPropPool_ThPHl_Top = BwTempEmissionPropPool_ThPHl_Top + (HmmFwdLow(i) *
              HmmBwdLow(j))
        Case "M"
            BwTempEmissionPropPool_TmPHh_Top = BwTempEmissionPropPool_TmPHh_Top + (HmmFwdHigh(i) *
              HmmBwdHigh(j))
            BwTempEmissionPropPool_TmPHm_Top = BwTempEmissionPropPool_TmPHm_Top + (HmmFwdMild(i) *
              HmmBwdMild(j))
            BwTempEmissionPropPool_TmPHl_Top = BwTempEmissionPropPool_TmPHl_Top + (HmmFwdLow(i) *
              HmmBwdLow(j))
        Case "L"
            BwTempEmissionPropPool_TlPHh_Top = BwTempEmissionPropPool_TlPHh_Top + (HmmFwdHigh(i) *
              HmmBwdHigh(j))
            BwTempEmissionPropPool_TlPHm_Top = BwTempEmissionPropPool_TlPHm_Top + (HmmFwdMild(i) *
              HmmBwdMild(j))
            BwTempEmissionPropPool_TlPHl_Top = BwTempEmissionPropPool_TlPHl_Top + (HmmFwdLow(i) *
              HmmBwdLow(j))
    End Select

    BwTempEmissionPropPool_PHh_Bottom = BwTempEmissionPropPool_PHh_Bottom + (HmmFwdHigh(i) *
     HmmBwdHigh(j))
    BwTempEmissionPropPool_PHm_Bottom = BwTempEmissionPropPool_PHm_Bottom + (HmmFwdMild(i) *
     HmmBwdMild(j))
    BwTempEmissionPropPool_PHl_Bottom = BwTempEmissionPropPool_PHl_Bottom + (HmmFwdLow(i) *
     HmmBwdLow(j))
```

```
Next i

BwTempEmissionPropPool_ClPHh = Math.Round(BwTempEmissionPropPool_ClPHh_Top /
 BwTempEmissionPropPool_PHh_Bottom, 2)
BwTempEmissionPropPool_ThPHh = Math.Round(BwTempEmissionPropPool_ThPHh_Top /
 BwTempEmissionPropPool_PHh_Bottom, 2)
BwTempEmissionPropPool_TmPHh = Math.Round(BwTempEmissionPropPool_TmPHh_Top /
 BwTempEmissionPropPool_PHh_Bottom, 2)
BwTempEmissionPropPool_TlPHh = Math.Round(BwTempEmissionPropPool_TlPHh_Top /
 BwTempEmissionPropPool_PHh_Bottom, 2)

BwTempEmissionPropPool_ClPHm = Math.Round(BwTempEmissionPropPool_ClPHm_Top /
 BwTempEmissionPropPool_PHm_Bottom, 2)
BwTempEmissionPropPool_ThPHm = Math.Round(BwTempEmissionPropPool_ThPHm_Top /
 BwTempEmissionPropPool_PHm_Bottom, 2)
BwTempEmissionPropPool_TmPHm = Math.Round(BwTempEmissionPropPool_TmPHm_Top /
 BwTempEmissionPropPool_PHm_Bottom, 2)
BwTempEmissionPropPool_TlPHm = Math.Round(BwTempEmissionPropPool_TlPHm_Top /
 BwTempEmissionPropPool_PHm_Bottom, 2)

BwTempEmissionPropPool_ClPHl = Math.Round(BwTempEmissionPropPool_ClPHl_Top /
 BwTempEmissionPropPool_PHl_Bottom, 2)
BwTempEmissionPropPool_ThPHl = Math.Round(BwTempEmissionPropPool_ThPHl_Top /
 BwTempEmissionPropPool_PHl_Bottom, 2)
BwTempEmissionPropPool_TmPHl = Math.Round(BwTempEmissionPropPool_TmPHl_Top /
 BwTempEmissionPropPool_PHl_Bottom, 2)
BwTempEmissionPropPool_TlPHl = Math.Round(BwTempEmissionPropPool_TlPHl_Top /
 BwTempEmissionPropPool_PHl_Bottom, 2)

Dim OnlyPart As Boolean = True

' Mod 3
If OnlyPart = False Then
    ' Original
    BwEmissionPropPool_ClPHh = BwTempEmissionPropPool_ClPHh
    BwEmissionPropPool_ThPHh = BwTempEmissionPropPool_ThPHh
    BwEmissionPropPool_TmPHh = BwTempEmissionPropPool_TmPHh
    BwEmissionPropPool_TlPHh = BwTempEmissionPropPool_TlPHh

    BwEmissionPropPool_ClPHm = BwTempEmissionPropPool_ClPHm
    BwEmissionPropPool_ThPHm = BwTempEmissionPropPool_ThPHm
    BwEmissionPropPool_TmPHm = BwTempEmissionPropPool_TmPHm
    BwEmissionPropPool_TlPHm = BwTempEmissionPropPool_TlPHm

    BwEmissionPropPool_ClPHl = BwTempEmissionPropPool_ClPHl
    BwEmissionPropPool_ThPHl = BwTempEmissionPropPool_ThPHl
    BwEmissionPropPool_TmPHl = BwTempEmissionPropPool_TmPHl
    BwEmissionPropPool_TlPHl = BwTempEmissionPropPool_TlPHl

Else

    BwEmissionPropPool_ClPHh = Math.Round((BwTempEmissionPropPool_ClPHh * ShareNew) +
     (BwEmissionPropPool_ClPHh * ShareOld), 2)
    BwEmissionPropPool_ThPHh = Math.Round((BwTempEmissionPropPool_ThPHh * ShareNew) +
     (BwEmissionPropPool_ThPHh * ShareOld), 2)
    BwEmissionPropPool_TmPHh = Math.Round((BwTempEmissionPropPool_TmPHh * ShareNew) +
     (BwEmissionPropPool_TmPHh * ShareOld), 2)
    BwEmissionPropPool_TlPHh = Math.Round((BwTempEmissionPropPool_TlPHh * ShareNew) +
     (BwEmissionPropPool_TlPHh * ShareOld), 2)

    BwEmissionPropPool_ClPHm = Math.Round((BwTempEmissionPropPool_ClPHm * ShareNew) +
     (BwEmissionPropPool_ClPHm * ShareOld), 2)
    BwEmissionPropPool_ThPHm = Math.Round((BwTempEmissionPropPool_ThPHm * ShareNew) +
     (BwEmissionPropPool_ThPHm * ShareOld), 2)
    BwEmissionPropPool_TmPHm = Math.Round((BwTempEmissionPropPool_TmPHm * ShareNew) +
     (BwEmissionPropPool_TmPHm * ShareOld), 2)
    BwEmissionPropPool_TlPHm = Math.Round((BwTempEmissionPropPool_TlPHm * ShareNew) +
     (BwEmissionPropPool_TlPHm * ShareOld), 2)

    BwEmissionPropPool_ClPHl = Math.Round((BwTempEmissionPropPool_ClPHl * ShareNew) +
     (BwEmissionPropPool_ClPHl * ShareOld), 2)
    BwEmissionPropPool_ThPHl = Math.Round((BwTempEmissionPropPool_ThPHl * ShareNew) +
     (BwEmissionPropPool_ThPHl * ShareOld), 2)
    BwEmissionPropPool_TmPHl = Math.Round((BwTempEmissionPropPool_TmPHl * ShareNew) +
     (BwEmissionPropPool_TmPHl * ShareOld), 2)
    BwEmissionPropPool_TlPHl = Math.Round((BwTempEmissionPropPool_TlPHl * ShareNew) +
     (BwEmissionPropPool_TlPHl * ShareOld), 2)
End If
```

```vb
'Mod 2-3
'Can never be 0 (High), Test Cloud
If BwEmissionPropPool_ClPHh = 0 Then
    If (BwEmissionPropPool_ThPHh >= BwEmissionPropPool_TmPHh) And (BwEmissionPropPool_ThPHh >=
     BwEmissionPropPool_TlPHh) Then
        BwEmissionPropPool_ThPHh -= 0.01
        BwEmissionPropPool_ClPHh += 0.01
    ElseIf (BwEmissionPropPool_TmPHh >= BwEmissionPropPool_ThPHh) And (BwEmissionPropPool_TmPHh >=
     BwEmissionPropPool_TlPHh) Then
        BwEmissionPropPool_TmPHh -= 0.01
        BwEmissionPropPool_ClPHh += 0.01
    Else
        BwEmissionPropPool_TlPHh -= 0.01
        BwEmissionPropPool_ClPHh += 0.01
    End If
End If

'Test High
If BwEmissionPropPool_ThPHh = 0 Then
    If (BwEmissionPropPool_ClPHh >= BwEmissionPropPool_TmPHh) And (BwEmissionPropPool_ClPHh >=
     BwEmissionPropPool_TlPHh) Then
        BwEmissionPropPool_ClPHh -= 0.01
        BwEmissionPropPool_ThPHh += 0.01
    ElseIf (BwEmissionPropPool_TmPHh >= BwEmissionPropPool_ClPHh) And (BwEmissionPropPool_TmPHh >=
     BwEmissionPropPool_TlPHh) Then
        BwEmissionPropPool_TmPHh -= 0.01
        BwEmissionPropPool_ThPHh += 0.01
    Else
        BwEmissionPropPool_TlPHh -= 0.01
        BwEmissionPropPool_ThPHh += 0.01
    End If
End If

'Test Mild
If BwEmissionPropPool_TmPHh = 0 Then
    If (BwEmissionPropPool_ClPHh >= BwEmissionPropPool_ThPHh) And (BwEmissionPropPool_ClPHh >=
     BwEmissionPropPool_TlPHh) Then
        BwEmissionPropPool_ClPHh -= 0.01
        BwEmissionPropPool_TmPHh += 0.01
    ElseIf (BwEmissionPropPool_ThPHh >= BwEmissionPropPool_ClPHh) And (BwEmissionPropPool_ThPHh >=
     BwEmissionPropPool_TlPHh) Then
        BwEmissionPropPool_ThPHh -= 0.01
        BwEmissionPropPool_TmPHh += 0.01
    Else
        BwEmissionPropPool_TlPHh -= 0.01
        BwEmissionPropPool_TmPHh += 0.01
    End If
End If

'Test Low
If BwEmissionPropPool_TlPHh = 0 Then
    If (BwEmissionPropPool_ClPHh >= BwEmissionPropPool_ThPHh) And (BwEmissionPropPool_ClPHh >=
     BwEmissionPropPool_TmPHh) Then
        BwEmissionPropPool_ClPHh -= 0.01
        BwEmissionPropPool_TlPHh += 0.01
    ElseIf (BwEmissionPropPool_ThPHh >= BwEmissionPropPool_ClPHh) And (BwEmissionPropPool_ThPHh >=
     BwEmissionPropPool_TmPHh) Then
        BwEmissionPropPool_ThPHh -= 0.01
        BwEmissionPropPool_TlPHh += 0.01
    Else
        BwEmissionPropPool_TmPHh -= 0.01
        BwEmissionPropPool_TlPHh += 0.01
    End If
End If

'Can never be 0 (Mild), Test Cloud
If BwEmissionPropPool_ClPHm = 0 Then
    If (BwEmissionPropPool_ThPHm >= BwEmissionPropPool_TmPHm) And (BwEmissionPropPool_ThPHm >=
     BwEmissionPropPool_TlPHm) Then
        BwEmissionPropPool_ThPHm -= 0.01
        BwEmissionPropPool_ClPHm += 0.01
    ElseIf (BwEmissionPropPool_TmPHm >= BwEmissionPropPool_ThPHm) And (BwEmissionPropPool_TmPHm >=
     BwEmissionPropPool_TlPHm) Then
        BwEmissionPropPool_TmPHm -= 0.01
        BwEmissionPropPool_ClPHm += 0.01
    Else
        BwEmissionPropPool_TlPHm -= 0.01
        BwEmissionPropPool_ClPHm += 0.01
    End If
End If
```

```
'Test High
If BwEmissionPropPool_ThPHm = 0 Then
    If (BwEmissionPropPool_ClPHm >= BwEmissionPropPool_TmPHm) And (BwEmissionPropPool_ClPHm >=
     BwEmissionPropPool_TlPHm) Then
        BwEmissionPropPool_ClPHm -= 0.01
        BwEmissionPropPool_ThPHm += 0.01
    ElseIf (BwEmissionPropPool_TmPHm >= BwEmissionPropPool_ClPHm) And (BwEmissionPropPool_TmPHm >=
     BwEmissionPropPool_TlPHm) Then
        BwEmissionPropPool_TmPHm -= 0.01
        BwEmissionPropPool_ThPHm += 0.01
    Else
        BwEmissionPropPool_TlPHm -= 0.01
        BwEmissionPropPool_ThPHm += 0.01
    End If
End If

'Test Mild
If BwEmissionPropPool_TmPHm = 0 Then
    If (BwEmissionPropPool_ClPHm >= BwEmissionPropPool_ThPHm) And (BwEmissionPropPool_ClPHm >=
     BwEmissionPropPool_TlPHm) Then
        BwEmissionPropPool_ClPHm -= 0.01
        BwEmissionPropPool_TmPHm += 0.01
    ElseIf (BwEmissionPropPool_ThPHm >= BwEmissionPropPool_ClPHm) And (BwEmissionPropPool_ThPHm >=
     BwEmissionPropPool_TlPHm) Then
        BwEmissionPropPool_ThPHm -= 0.01
        BwEmissionPropPool_TmPHm += 0.01
    Else
        BwEmissionPropPool_TlPHm -= 0.01
        BwEmissionPropPool_TmPHm += 0.01
    End If
End If

'Test Low
If BwEmissionPropPool_TlPHm = 0 Then
    If (BwEmissionPropPool_ClPHm >= BwEmissionPropPool_ThPHm) And (BwEmissionPropPool_ClPHm >=
     BwEmissionPropPool_TmPHm) Then
        BwEmissionPropPool_ClPHm -= 0.01
        BwEmissionPropPool_TlPHm += 0.01
    ElseIf (BwEmissionPropPool_ThPHm >= BwEmissionPropPool_ClPHm) And (BwEmissionPropPool_ThPHm >=
     BwEmissionPropPool_TmPHm) Then
        BwEmissionPropPool_ThPHm -= 0.01
        BwEmissionPropPool_TlPHm += 0.01
    Else
        BwEmissionPropPool_TmPHm -= 0.01
        BwEmissionPropPool_TlPHm += 0.01
    End If
End If

'Can never be 0 (Low), Test Cloud
If BwEmissionPropPool_ClPHl = 0 Then
    If (BwEmissionPropPool_ThPHl >= BwEmissionPropPool_TmPHl) And (BwEmissionPropPool_ThPHl >=
     BwEmissionPropPool_TlPHl) Then
        BwEmissionPropPool_ThPHl -= 0.01
        BwEmissionPropPool_ClPHl += 0.01
    ElseIf (BwEmissionPropPool_TmPHl >= BwEmissionPropPool_ThPHl) And (BwEmissionPropPool_TmPHl >=
     BwEmissionPropPool_TlPHl) Then
        BwEmissionPropPool_TmPHl -= 0.01
        BwEmissionPropPool_ClPHl += 0.01
    Else
        BwEmissionPropPool_TlPHl -= 0.01
        BwEmissionPropPool_ClPHl += 0.01
    End If
End If

'Test High
If BwEmissionPropPool_ThPHl = 0 Then
    If (BwEmissionPropPool_ClPHl >= BwEmissionPropPool_TmPHl) And (BwEmissionPropPool_ClPHl >=
     BwEmissionPropPool_TlPHl) Then
        BwEmissionPropPool_ClPHl -= 0.01
        BwEmissionPropPool_ThPHl += 0.01
    ElseIf (BwEmissionPropPool_TmPHl >= BwEmissionPropPool_ClPHl) And (BwEmissionPropPool_TmPHl >=
     BwEmissionPropPool_TlPHl) Then
        BwEmissionPropPool_TmPHl -= 0.01
        BwEmissionPropPool_ThPHl += 0.01
    Else
        BwEmissionPropPool_TlPHl -= 0.01
        BwEmissionPropPool_ThPHl += 0.01
    End If
End If
```

```vb
    'Test Mild
    If BwEmissionPropPool_TmPHl = 0 Then
        If (BwEmissionPropPool_ClPHl >= BwEmissionPropPool_ThPHl) And (BwEmissionPropPool_ClPHl >=
         BwEmissionPropPool_TlPHl) Then
            BwEmissionPropPool_ClPHl -= 0.01
            BwEmissionPropPool_TmPHl += 0.01
        ElseIf (BwEmissionPropPool_ThPHl >= BwEmissionPropPool_ClPHl) And (BwEmissionPropPool_ThPHl >=
         BwEmissionPropPool_TlPHl) Then
            BwEmissionPropPool_ThPHl -= 0.01
            BwEmissionPropPool_TmPHl += 0.01
        Else
            BwEmissionPropPool_TlPHl -= 0.01
            BwEmissionPropPool_TmPHl += 0.01
        End If
    End If

    'Test Low
    If BwEmissionPropPool_TlPHl = 0 Then
        If (BwEmissionPropPool_ClPHl >= BwEmissionPropPool_ThPHl) And (BwEmissionPropPool_ClPHl >=
         BwEmissionPropPool_TmPHl) Then
            BwEmissionPropPool_ClPHl -= 0.01
            BwEmissionPropPool_TlPHl += 0.01
        ElseIf (BwEmissionPropPool_ThPHl >= BwEmissionPropPool_ClPHl) And (BwEmissionPropPool_ThPHl >=
         BwEmissionPropPool_TmPHl) Then
            BwEmissionPropPool_ThPHl -= 0.01
            BwEmissionPropPool_TlPHl += 0.01
        Else
            BwEmissionPropPool_TmPHl -= 0.01
            BwEmissionPropPool_TlPHl += 0.01
        End If
    End If
End Sub


Private Sub Hmm_Fwd(Disp As Char, PropHigh As Double, PropMild As Double, PropLow As Double,
                       EmHCl As Double, EmHTh As Double, EmHTm As Double, EmHTl As Double,
                       EmMCl As Double, EmMTh As Double, EmMTm As Double, EmMTl As Double,
                       EmLCl As Double, EmLTh As Double, EmLTm As Double, EmLTl As Double,
                       TrHH As Double, TrHM As Double, TrHL As Double,
                       TrMH As Double, TrMM As Double, TrML As Double,
                       TrLH As Double, TrLM As Double, TrLL As Double,
                       ObsPath As String)

    Dim HmmFwdPath(60) As String
    Dim i As Integer

    For i = 0 To HmmFwdPath.Length - 1

        Select Case HmmFwdPath(i)
            Case "Cl"
                If i = 0 Then
                    HmmFwdHigh(i) = PropHigh * EmHCl
                    HmmFwdMild(i) = PropMild * EmMCl
                    HmmFwdLow(i) = PropLow * EmLCl
                Else
                    HmmFwdHigh(i) = (HmmFwdHigh(i - 1) * TrHH + HmmFwdMild(i - 1) * TrMH + HmmFwdLow(i
                      - 1) * TrLH) * EmHCl
                    HmmFwdMild(i) = (HmmFwdHigh(i - 1) * TrHM + HmmFwdMild(i - 1) * TrMM + HmmFwdLow(i
                      - 1) * TrLM) * EmMCl
                    HmmFwdLow(i) = (HmmFwdHigh(i - 1) * TrHL + HmmFwdMild(i - 1) * TrML + HmmFwdLow(i -
                      1) * TrLL) * EmLCl
                End If
            Case "H"
                If i = 0 Then
                    HmmFwdHigh(i) = PropHigh * EmHTh
                    HmmFwdMild(i) = PropMild * EmMTh
                    HmmFwdLow(i) = PropLow * EmLTh
                Else
                    HmmFwdHigh(i) = (HmmFwdHigh(i - 1) * TrHH + HmmFwdMild(i - 1) * TrMH + HmmFwdLow(i
                      - 1) * TrLH) * EmHTh
                    HmmFwdMild(i) = (HmmFwdHigh(i - 1) * TrHM + HmmFwdMild(i - 1) * TrMM + HmmFwdLow(i
                      - 1) * TrLM) * EmMTh
                    HmmFwdLow(i) = (HmmFwdHigh(i - 1) * TrHL + HmmFwdMild(i - 1) * TrML + HmmFwdLow(i -
                      1) * TrLL) * EmLTh
                End If
            Case "M"
                If i = 0 Then
                    HmmFwdHigh(i) = PropHigh * EmHTm
                    HmmFwdMild(i) = PropMild * EmMTm
```

```
                HmmFwdLow(i) = PropLow * EmLTm
            Else
                HmmFwdHigh(i) = (HmmFwdHigh(i - 1) * TrHH + HmmFwdMild(i - 1) * TrMH + HmmFwdLow(i
                  - 1) * TrLH) * EmHTm
                HmmFwdMild(i) = (HmmFwdHigh(i - 1) * TrHM + HmmFwdMild(i - 1) * TrMM + HmmFwdLow(i
                  - 1) * TrLM) * EmMTm
                HmmFwdLow(i) = (HmmFwdHigh(i - 1) * TrHL + HmmFwdMild(i - 1) * TrML + HmmFwdLow(i -
                  1) * TrLL) * EmLTm
            End If
        Case "L"
            If i = 0 Then
                HmmFwdHigh(i) = PropHigh * EmHTl
                HmmFwdMild(i) = PropMild * EmMTl
                HmmFwdLow(i) = PropLow * EmLTl
            Else
                HmmFwdHigh(i) = (HmmFwdHigh(i - 1) * TrHH + HmmFwdMild(i - 1) * TrMH + HmmFwdLow(i
                  - 1) * TrLH) * EmHTl
                HmmFwdMild(i) = (HmmFwdHigh(i - 1) * TrHM + HmmFwdMild(i - 1) * TrMM + HmmFwdLow(i
                  - 1) * TrLM) * EmMTl
                HmmFwdLow(i) = (HmmFwdHigh(i - 1) * TrHL + HmmFwdMild(i - 1) * TrML + HmmFwdLow(i -
                  1) * TrLL) * EmLTl
            End If

    End Select
    HmmFwdHigh(i) = Math.Round(HmmFwdHigh(i), 10)
    HmmFwdMild(i) = Math.Round(HmmFwdMild(i), 10)
    HmmFwdLow(i) = Math.Round(HmmFwdLow(i), 10)

Next
HmmFwdLikelihood = Math.Round(HmmFwdHigh(i - 1) + HmmFwdMild(i - 1) + HmmFwdLow(i - 1), 10)

End Sub

Private Sub Hmm_Bwd(Disp As Char, PropHigh As Double, PropMild As Double, PropLow As Double,
                     EmHCl As Double, EmHTh As Double, EmHTm As Double, EmHTl As Double,
                     EmMCl As Double, EmMTh As Double, EmMTm As Double, EmMTl As Double,
                     EmLCl As Double, EmLTh As Double, EmLTm As Double, EmLTl As Double,
                     TrHH As Double, TrHM As Double, TrHL As Double,
                     TrMH As Double, TrMM As Double, TrML As Double,
                     TrLH As Double, TrLM As Double, TrLL As Double,
                     ObsPath As String)

    Dim HmmBwdPath(60) As String
    HmmBwdPath = Split(ObsPath, ",")
    Dim i As Integer

    For i = HmmBwdPath.Length - 1 To 0 Step -1
        HmmBwdHigh(HmmBwdPath.Length) = 1
        HmmBwdMild(HmmBwdPath.Length) = 1
        HmmBwdLow(HmmBwdPath.Length) = 1

        Select Case HmmBwdPath(i)
            Case "Cl"
                If i = 0 Then
                    HmmBwdLikelihood = Math.Round(PropHigh * HmmBwdHigh(1) * EmHCl + PropMild *
                      HmmBwdMild(1) * EmMCl + PropLow * HmmBwdLow(1) * EmLCl, 10)
                Else
                    HmmBwdHigh(i) = TrHH * EmHCl * HmmBwdHigh(i + 1) + TrHM * EmMCl * HmmBwdMild(i + 1)
                      + TrHL * EmLCl * HmmBwdLow(i + 1)
                    HmmBwdMild(i) = TrMH * EmHCl * HmmBwdHigh(i + 1) + TrMM * EmMCl * HmmBwdMild(i + 1)
                      + TrML * EmLCl * HmmBwdLow(i + 1)
                    HmmBwdLow(i) = TrLH * EmHCl * HmmBwdHigh(i + 1) + TrLM * EmMCl * HmmBwdMild(i + 1)
                      + TrLL * EmLCl * HmmBwdLow(i + 1)
                End If
            Case "H"
                If i = 0 Then
                    HmmBwdLikelihood = Math.Round(PropHigh * HmmBwdHigh(1) * EmHTh + PropMild *
                      HmmBwdMild(1) * EmMTh + PropLow * HmmBwdLow(1) * EmLTh, 10)
                Else
                    HmmBwdHigh(i) = TrHH * EmHTh * HmmBwdHigh(i + 1) + TrHM * EmMTh * HmmBwdMild(i + 1)
                      + TrHL * EmLTh * HmmBwdLow(i + 1)
                    HmmBwdMild(i) = TrMH * EmHTh * HmmBwdHigh(i + 1) + TrMM * EmMTh * HmmBwdMild(i + 1)
                      + TrML * EmLTh * HmmBwdLow(i + 1)
                    HmmBwdLow(i) = TrLH * EmHTh * HmmBwdHigh(i + 1) + TrLM * EmMTh * HmmBwdMild(i + 1)
                      + TrLL * EmLTh * HmmBwdLow(i + 1)
                End If
            Case "M"
                If i = 0 Then
                    HmmBwdLikelihood = Math.Round(PropHigh * HmmBwdHigh(1) * EmHTm + PropMild *
                      HmmBwdMild(1) * EmMTm + PropLow * HmmBwdLow(1) * EmLTm, 10)
```

```
                Else
                    HmmBwdHigh(i) = TrHH * EmHTm * HmmBwdHigh(i + 1) + TrHM * EmMTm * HmmBwdMild(i + 1)
                      + TrHL * EmLTm * HmmBwdLow(i + 1)
                    HmmBwdMild(i) = TrMH * EmHTm * HmmBwdHigh(i + 1) + TrMM * EmMTm * HmmBwdMild(i + 1)
                      + TrML * EmLTm * HmmBwdLow(i + 1)
                    HmmBwdLow(i) = TrLH * EmHTm * HmmBwdHigh(i + 1) + TrLM * EmMTm * HmmBwdMild(i + 1)
                      + TrLL * EmLTm * HmmBwdLow(i + 1)
                End If
            Case "L"
                If i = 0 Then
                    HmmBwdLikelihood = Math.Round(PropHigh * HmmBwdHigh(1) * EmHTl + PropMild *
                      HmmBwdMild(1) * EmMTl + PropLow * HmmBwdLow(1) * EmLTl, 10)
                Else
                    HmmBwdHigh(i) = TrHH * EmHTl * HmmBwdHigh(i + 1) + TrHM * EmMTl * HmmBwdMild(i + 1)
                      + TrHL * EmLTl * HmmBwdLow(i + 1)
                    HmmBwdMild(i) = TrMH * EmHTl * HmmBwdHigh(i + 1) + TrMM * EmMTl * HmmBwdMild(i + 1)
                      + TrML * EmLTl * HmmBwdLow(i + 1)
                    HmmBwdLow(i) = TrLH * EmHTl * HmmBwdHigh(i + 1) + TrLM * EmMTl * HmmBwdMild(i + 1)
                      + TrLL * EmLTl * HmmBwdLow(i + 1)
                End If
        End Select

        HmmBwdHigh(i) = Math.Round(HmmBwdHigh(i), 10)
        HmmBwdMild(i) = Math.Round(HmmBwdMild(i), 10)
        HmmBwdLow(i) = Math.Round(HmmBwdLow(i), 10)
    Next

End Sub
```

Appendix M: Sample of Baum-Welch related Pool calculations

Date	ObsWeather	PoolTempRec	Pool TempPredRec	Pool Temp Pred Past	Pool Temp Pred	Predict Correct	PrH	PrM	PrL	TrHH	TrHM	TrHl	TrMH	TrMM	TrML	TrLH	TrLM	TrLL	EmCH	EMCM	EmCL	EmHH	EmHM	EmHL	EmMH	EmMM	EmML	EmLH	EmLM	EmLL
2019/03/15							0.48	0.34	0.18	0.75	0.17	0.08	0.19	0.62	0.19	0.22	0.33	0.44	0	0.06	0.11	0.96	0.71	0.11	0.04	0.24	0.56	0	0	0.22
2019/03/22	H,H,H,H,H,H,H,H	H,H,H,H,H,H,H	H	H,H,H,H,H	H	C	0.76	0.22	0.02	0.89	0.1	0.01	0.38	0.6	0.02	0.54	0.39	0.07	0.01	0.05	0.1	0.94	0.73	0.2	0.04	0.22	0.5	0.01	0.01	0.2
2019/03/23	H,H,H,H,H,H,H,H	H,H,H,H,H,H,H	H	H,H,H,H,H	H	C	0.84	0.15	0.01	0.92	0.07	0.01	0.49	0.49	0.01	0.67	0.31	0.02	0.01	0.05	0.09	0.95	0.76	0.28	0.04	0.2	0.45	0.01	0.01	0.18
2019/03/24	H,H,H,H,H,H,M,H	H,H,H,H,H,H,H	H	H,H,H,H,H	H	C	0.88	0.11	0.01	0.9	0.09	0.01	0.47	0.52	0.01	0.7	0.29	0.01	0.01	0.05	0.08	0.95	0.76	0.28	0.05	0.21	0.48	0.01	0.01	0.16
2019/03/25	H,H,H,H,H,M,M,H	H,H,H,H,H,H,H	L	H,H,H,H,H	H	I	0.91	0.08	0.01	0.82	0.16	0.02	0.34	0.65	0.02	0.54	0.44	0.03	0.01	0.05	0.07	0.94	0.73	0.27	0.06	0.24	0.52	0.01	0.01	0.14
2019/03/26	H,H,H,H,M,M,H,H	H,H,H,H,H,H,L	H	H,M,M,H,H	H	C	0.93	0.06	0.01	0.74	0.23	0.03	0.27	0.71	0.03	0.41	0.53	0.06	0.01	0.05	0.06	0.94	0.71	0.26	0.06	0.26	0.55	0.01	0.01	0.13
2019/03/27	H,H,H,M,M,H,H,H	H,H,H,H,H,L,H	H	H,M,H,H,H	H	C	0.94	0.05	0.01	0.67	0.3	0.04	0.24	0.73	0.03	0.33	0.57	0.09	0.01	0.05	0.05	0.94	0.7	0.25	0.06	0.28	0.58	0.01	0.01	0.12
2019/03/28	H,H,M,M,H,H,M,H	H,H,H,H,L,H,H	M	H,H,H,M,H	H	B	0.95	0.04	0.01	0.48	0.45	0.08	0.18	0.78	0.04	0.27	0.61	0.11	0.01	0.05	0.05	0.93	0.68	0.24	0.07	0.3	0.61	0.01	0.01	0.11
2019/03/29	H,M,M,H,H,M,H,H	H,H,H,L,H,H,M	H	L,M,M,H,H	H	C	0.95	0.04	0.01	0.29	0.56	0.16	0.18	0.79	0.04	0.24	0.63	0.13	0.01	0.05	0.05	0.92	0.67	0.23	0.08	0.32	0.64	0.01	0.01	0.1
2019/03/30	M,M,H,H,M,H,H,H	H,H,L,H,H,M,H	H	H,M,H,H,H	H	C	0.8	0.13	0.07	0.22	0.58	0.2	0.2	0.78	0.03	0.28	0.64	0.08	0.01	0.05	0.05	0.89	0.67	0.23	0.11	0.32	0.66	0.01	0.01	0.09
2019/03/31	M,H,H,M,H,H,H,H	H,L,H,H,M,H,H	H	H,H,M,H,H	H	C	0.47	0.25	0.27	0.26	0.61	0.12	0.22	0.76	0.02	0.33	0.64	0.03	0.01	0.05	0.05	0.88	0.68	0.24	0.12	0.31	0.67	0.01	0.01	0.08
2019/04/01	H,H,M,H,H,H,H,H	L,H,H,M,H,H,H	H	M,H,H,H,H	H	C	0.62	0.27	0.11	0.3	0.62	0.08	0.24	0.75	0.01	0.37	0.61	0.02	0.01	0.05	0.05	0.89	0.7	0.28	0.11	0.29	0.64	0.01	0.01	0.07
2019/04/02	H,M,H,H,H,H,H,H	H,H,M,H,H,H,H	H	H,H,H,H,H	H	C	0.73	0.24	0.04	0.29	0.63	0.08	0.27	0.72	0.01	0.39	0.6	0.01	0.01	0.05	0.05	0.9	0.72	0.29	0.1	0.28	0.64	0.01	0.01	0.06
2019/04/03	M,H,H,H,H,H,M,H	H,M,H,H,H,H,H	H	H,H,H,M,H	H	C	0.43	0.41	0.16	0.31	0.63	0.06	0.29	0.71	0.01	0.43	0.55	0.01	0.01	0.05	0.05	0.89	0.72	0.28	0.11	0.28	0.65	0.01	0.01	0.05
2019/04/04	H,H,H,H,H,M,M,H	M,H,H,H,H,H,H	H	H,H,M,M,H	H	C	0.52	0.42	0.06	0.31	0.64	0.05	0.27	0.72	0.01	0.4	0.59	0.01	0.01	0.05	0.05	0.89	0.72	0.3	0.11	0.28	0.64	0.01	0.01	0.05
2019/04/05	H,H,H,H,M,M,M,Cl	H,H,H,H,H,H,H	L	H,M,M,M,M	M	B	0.59	0.39	0.02	0.28	0.67	0.05	0.18	0.8	0.02	0.25	0.73	0.02	0.01	0.06	0.06	0.88	0.69	0.3	0.12	0.3	0.64	0.01	0.01	0.05
2019/04/06	H,H,H,M,M,M,Cl,Cl	H,H,H,H,H,H,L	M	H,M,M,M,M	M	C	0.66	0.34	0.01	0.26	0.69	0.05	0.09	0.88	0.03	0.11	0.86	0.03	0.01	0.08	0.07	0.87	0.65	0.28	0.12	0.31	0.64	0.01	0.01	0.05
2019/04/07	H,H,M,M,M,Cl,Cl,M	H,H,H,H,H,L,M	L	H,M,M,M,M	M	B	0.71	0.27	0.01	0.23	0.72	0.05	0.03	0.92	0.05	0.03	0.92	0.05	0.01	0.1	0.08	0.87	0.6	0.26	0.12	0.33	0.65	0.01	0.01	0.05
2019/04/08	H,M,M,M,Cl,Cl,L,M	H,H,H,H,L,M,L	L	H,M,M,M,M	M	B	0.75	0.23	0.01	0.08	0.81	0.11	0.01	0.89	0.1	0.01	0.9	0.09	0.01	0.12	0.08	0.87	0.54	0.24	0.12	0.35	0.64	0.01	0.02	0.08
2019/04/09	M,M,M,Cl,Cl,L,L,H	H,H,H,L,M,L,L	H	L,M,M,L,M	M	B	0.5	0.46	0.04	0.03	0.78	0.19	0.01	0.82	0.16	0.01	0.85	0.15	0.01	0.14	0.08	0.79	0.5	0.22	0.2	0.35	0.61	0.01	0.04	0.12
2019/04/10	M,M,Cl,Cl,L,L,M,H	H,H,L,M,L,L,H	M	M,M,L,M,M	M	C	0.36	0.56	0.09	0.01	0.69	0.29	0.01	0.77	0.22	0.01	0.79	0.21	0.01	0.16	0.08	0.72	0.46	0.2	0.27	0.35	0.59	0.01	0.06	0.15
2019/04/11	M,Cl,Cl,L,L,M,H,M	H,L,M,L,L,H,M	M	L,M,L,M,M	M	C	0.27	0.57	0.16	0.01	0.8	0.19	0.01	0.72	0.27	0.01	0.73	0.26	0.01	0.17	0.08	0.65	0.43	0.19	0.34	0.35	0.57	0.01	0.08	0.18
2019/04/12	Cl,Cl,L,L,M,H,M,H	L,M,L,L,H,M,M	M	L,M,M,M,M	M	C	0.03	0.86	0.11	0.01	0.84	0.16	0.01	0.69	0.3	0.01	0.7	0.29	0.04	0.18	0.08	0.63	0.42	0.18	0.33	0.34	0.55	0.01	0.09	0.2
2019/04/13	Cl,L,L,M,H,M,M,M	M,L,L,H,M,M,M	L	H,L,M,M,M	M	B	0.01	0.94	0.05	0.01	0.78	0.21	0.01	0.59	0.4	0.01	0.62	0.37	0.05	0.18	0.07	0.6	0.39	0.17	0.35	0.35	0.55	0.01	0.1	0.22
2019/04/14	L,L,M,H,M,M,L,M	L,L,H,M,M,M,L	M	M,L,M,M,L	L	B	0.01	0.89	0.1	0.01	0.73	0.27	0.01	0.49	0.5	0.01	0.54	0.45	0.05	0.16	0.06	0.57	0.37	0.16	0.38	0.36	0.55	0.01	0.13	0.24
2019/04/15	L,M,H,M,M,L,M,H	L,H,M,M,M,L,M	M	M,L,M,L,M	M	C	0.01	0.82	0.17	0.01	0.71	0.28	0.01	0.45	0.54	0.01	0.53	0.46	0.05	0.14	0.05	0.57	0.37	0.16	0.39	0.36	0.56	0.01	0.15	0.24
2019/04/16	M,H,M,M,L,M,H,H	H,M,M,M,L,M,M	M	M,L,L,M,L	L	B	0.01	0.74	0.25	0.01	0.74	0.24	0.01	0.49	0.49	0.01	0.52	0.46	0.05	0.13	0.05	0.58	0.38	0.17	0.39	0.37	0.56	0.01	0.14	0.23
2019/04/17	H,M,M,L,M,H,M,H	M,M,M,L,M,M,M	M	L,L,M,M,M	M	C	0.02	0.86	0.13	0.01	0.72	0.27	0.01	0.46	0.53	0.01	0.52	0.47	0.05	0.12	0.05	0.58	0.4	0.17	0.39	0.37	0.57	0.01	0.14	0.22
2019/06/27	M,L,L,L,L,L,L,L	L,L,L,L,L,L,L	L	L,M,L,L,L	L	C	0.02	0.22	0.76	0.01	0.82	0.17	0.01	0.53	0.46	0.01	0.62	0.36	0.05	0.05	0.05	0.05	0.05	0.05	0.81	0.33	0.44	0.16	0.66	0.57
2019/06/28	L,L,L,L,L,L,L,M	L,L,L,L,L,L,L	L	L,M,L,M,L	L	C	0.01	0.25	0.75	0.01	0.83	0.16	0.01	0.55	0.44	0.01	0.65	0.34	0.05	0.05	0.05	0.05	0.05	0.05	0.78	0.31	0.41	0.19	0.68	0.6
2019/06/29	L,L,L,L,L,L,L,H	L,L,L,L,L,L,L	L	L,M,L,M,L	L	C	0.01	0.27	0.72	0.01	0.84	0.15	0.01	0.57	0.42	0.01	0.67	0.32	0.05	0.05	0.05	0.08	0.06	0.06	0.7	0.28	0.37	0.24	0.7	0.63
2019/06/30	L,L,L,L,L,L,M,H	L,L,L,L,L,L,L	L	L,M,L,L,M	M	B	0.01	0.29	0.7	0.01	0.84	0.15	0.01	0.58	0.41	0.01	0.68	0.31	0.05	0.05	0.05	0.1	0.07	0.06	0.67	0.26	0.35	0.25	0.7	0.64
2019/07/01	L,L,L,L,L,M,M,L	L,L,L,L,L,L,L	L	L,M,L,L,M	M	B	0.01	0.31	0.68	0.01	0.83	0.16	0.01	0.58	0.42	0.01	0.68	0.31	0.05	0.05	0.05	0.09	0.06	0.05	0.67	0.26	0.34	0.26	0.71	0.65
2019/07/02	L,L,L,L,M,M,L,L	L,L,L,L,L,L,L	L	L,M,L,M,L	L	C	0.01	0.33	0.65	0.01	0.82	0.17	0.01	0.57	0.42	0.01	0.68	0.31	0.05	0.05	0.05	0.08	0.05	0.05	0.67	0.26	0.33	0.27	0.72	0.66
2019/07/03	L,L,L,M,M,L,L,M	L,L,L,L,L,L,L	L	L,M,L,M,L	L	C	0.01	0.35	0.63	0.01	0.81	0.18	0.01	0.56	0.43	0.01	0.67	0.32	0.05	0.05	0.05	0.07	0.05	0.05	0.68	0.27	0.34	0.26	0.71	0.66
2019/07/04	L,L,M,M,L,L,M,M	L,L,L,L,L,L,L	L	L,M,L,L,L	L	C	0.01	0.37	0.62	0.02	0.79	0.2	0.01	0.53	0.45	0.01	0.65	0.34	0.05	0.05	0.05	0.06	0.05	0.05	0.7	0.29	0.36	0.25	0.69	0.64
2019/07/05	L,M,M,L,L,M,M,M	L,L,L,L,L,L,L	L	L,M,L,L,L	L	C	0.01	0.39	0.59	0.03	0.75	0.22	0.02	0.5	0.48	0.02	0.61	0.37	0.05	0.05	0.05	0.05	0.05	0.05	0.72	0.32	0.39	0.23	0.66	0.61
2019/07/06	M,M,L,L,M,M,M,L	L,L,L,L,L,L,L	L	L,M,L,L,L	L	C	0.02	0.35	0.63	0.04	0.73	0.23	0.02	0.48	0.49	0.03	0.59	0.39	0.05	0.05	0.05	0.05	0.05	0.05	0.74	0.35	0.42	0.22	0.64	0.58
2019/07/07	M,L,L,M,M,M,L,M	L,L,L,L,L,L,L	L	L,M,L,L,L	L	C	0.04	0.3	0.66	0.05	0.71	0.24	0.03	0.46	0.51	0.04	0.57	0.4	0.05	0.05	0.05	0.05	0.05	0.05	0.76	0.37	0.44	0.21	0.62	0.56
2019/07/08	L,L,M,M,M,L,M,L	L,L,L,L,L,L,L	L	L,M,L,L,L	L	C	0.01	0.33	0.66	0.06	0.7	0.25	0.04	0.44	0.52	0.04	0.55	0.4	0.05	0.05	0.05	0.05	0.05	0.05	0.77	0.38	0.45	0.21	0.61	0.55
2019/07/09	L,M,M,M,L,M,L,M	L,L,L,L,L,L,L	L	L,M,L,L,L	L	C	0.01	0.36	0.62	0.07	0.68	0.25	0.06	0.41	0.53	0.06	0.53	0.41	0.05	0.05	0.05	0.05	0.05	0.05	0.78	0.4	0.47	0.2	0.59	0.53
2019/07/10	M,M,M,L,M,L,M,H	L,L,L,L,L,L,L	L	L,M,L,L,L	L	C	0.02	0.33	0.65	0.07	0.68	0.25	0.08	0.39	0.53	0.07	0.51	0.42	0.05	0.05	0.05	0.06	0.06	0.06	0.78	0.42	0.49	0.19	0.56	0.5
2019/07/11	M,M,L,M,L,M,M,H	L,L,L,L,L,L,L	L	L,M,L,L,L	L	C	0.03	0.3	0.67	0.07	0.68	0.25	0.1	0.37	0.53	0.08	0.49	0.43	0.05	0.05	0.05	0.06	0.07	0.07	0.78	0.43	0.51	0.18	0.54	0.47
2019/07/12	M,L,M,L,M,M,H,M	L,L,L,L,L,L,L	L	L,L,L,L,L	L	C	0.05	0.25	0.7	0.07	0.68	0.25	0.12	0.35	0.53	0.08	0.48	0.43	0.05	0.05	0.05	0.06	0.08	0.07	0.78	0.44	0.53	0.17	0.52	0.44
2019/07/13	L,M,L,M,M,H,M,M	L,L,L,L,L,L,L	L	L,L,L,L,L	L	C	0.02	0.3	0.68	0.08	0.67	0.25	0.15	0.33	0.53	0.1	0.46	0.44	0.05	0.05	0.05	0.06	0.09	0.07	0.79	0.46	0.54	0.16	0.49	0.43
2019/07/14	M,L,M,M,H,M,M,M	L,L,L,L,L,L,L	L	L,M,L,L,L	L	C	0.03	0.26	0.71	0.11	0.64	0.25	0.18	0.31	0.51	0.11	0.45	0.43	0.05	0.05	0.05	0.06	0.1	0.07	0.8	0.48	0.56	0.15	0.46	0.4
2019/07/15	L,M,M,H,M,M,M,M	L,L,L,L,L,L,L	L	L,L,L,L,L	L	C	0.01	0.3	0.69	0.15	0.61	0.25	0.23	0.28	0.48	0.15	0.43	0.43	0.05	0.05	0.05	0.06	0.11	0.07	0.81	0.5	0.58	0.14	0.42	0.38
2019/07/16	M,M,H,M,M,M,M,M	L,L,L,L,L,L,L	L	L,L,L,L,L	L	C	0.01	0.28	0.71	0.19	0.58	0.24	0.29	0.25	0.46	0.19	0.39	0.41	0.05	0.05	0.05	0.06	0.12	0.07	0.82	0.53	0.61	0.13	0.38	0.34

Date	ObsWeather	PoolTempRec	Pool TempPred Rec	Pool Temp Pred Past	Pool Temp Pred	Predict Correct	PrH	PrM	PrL	TrHH	TrHM	TrHl	TrMH	TrMM	TrML	TrLH	TrLM	TrLL	EmCH	EMCM	EmCL	EmHH	EmHM	EmHL	EmMH	EmMM	EmML	EmLH	EmLM	EmLL
2019/09/20	H,H,M,H,H,H,H,H	H,M,M,H,H,H,H	H	H,H,H,H,H	H	C	0.07	0.76	0.17	0.18	0.81	0.01	0.53	0.46	0.01	0.56	0.44	0.01	0.05	0.05	0.05	0.5	0.55	0.36	0.44	0.38	0.55	0.09	0.09	0.1
2019/09/21	H,M,H,H,H,H,H,H	M,M,H,H,H,H,H	H	H,H,H,H,H	H	C	0.06	0.81	0.12	0.17	0.82	0.01	0.52	0.47	0.01	0.58	0.41	0.01	0.05	0.05	0.05	0.53	0.59	0.42	0.42	0.35	0.5	0.08	0.08	0.09
2019/09/22	M,H,H,H,H,H,H,H	M,H,H,H,H,H,H	H	H,H,H,H,H	H	C	0.07	0.77	0.16	0.16	0.83	0.01	0.5	0.49	0.01	0.56	0.43	0.01	0.05	0.05	0.05	0.57	0.62	0.4	0.38	0.33	0.53	0.07	0.07	0.08
2019/09/23	H,H,H,H,H,H,H,H	H,H,H,H,H,H,H	H	H,H,H,H,H	H	C	0.07	0.82	0.11	0.15	0.84	0.01	0.49	0.51	0.01	0.55	0.45	0.01	0.05	0.05	0.05	0.61	0.66	0.46	0.34	0.3	0.48	0.06	0.06	0.07
2019/09/24	H,H,H,H,H,H,H,L	H,H,H,H,H,H,H	M	H,H,H,H,M	M	C	0.07	0.85	0.08	0.14	0.85	0.01	0.47	0.52	0.01	0.53	0.46	0.01	0.05	0.05	0.05	0.64	0.68	0.5	0.31	0.27	0.43	0.07	0.07	0.07
2019/09/25	H,H,H,H,H,H,L,L	H,H,H,H,H,H,M	L	H,H,H,M,M	M	B	0.07	0.87	0.06	0.14	0.86	0.01	0.46	0.53	0.01	0.52	0.47	0.01	0.05	0.05	0.05	0.65	0.69	0.53	0.28	0.24	0.39	0.09	0.09	0.08
2019/09/26	H,H,H,H,H,L,L,M	H,H,H,H,H,M,L	L	H,H,M,M,M	M	B	0.07	0.88	0.05	0.14	0.85	0.01	0.46	0.53	0.01	0.52	0.47	0.01	0.05	0.05	0.05	0.64	0.68	0.55	0.27	0.23	0.37	0.11	0.11	0.09
2019/09/27	H,H,H,H,L,L,L,H	H,H,H,H,M,L,L	L	H,M,M,M,H	H	I	0.07	0.89	0.04	0.14	0.85	0.01	0.45	0.54	0.01	0.51	0.48	0.01	0.05	0.05	0.05	0.64	0.68	0.57	0.24	0.21	0.33	0.14	0.14	0.11
2019/09/28	H,H,H,L,L,L,M,H	H,H,H,M,L,L,L	M	M,M,M,M,H	H	B	0.07	0.9	0.03	0.14	0.85	0.01	0.45	0.54	0.01	0.51	0.49	0.01	0.05	0.05	0.05	0.62	0.66	0.57	0.23	0.2	0.31	0.17	0.16	0.12
2019/09/29	H,H,L,L,L,M,H,H	H,H,M,L,L,L,M	L	L,H,M,H,M	M	B	0.07	0.91	0.03	0.14	0.85	0.01	0.45	0.54	0.01	0.5	0.49	0.01	0.05	0.05	0.05	0.6	0.65	0.57	0.22	0.19	0.3	0.19	0.18	0.13
2019/09/30	H,L,L,L,M,H,H,H	H,M,L,L,L,M,L	L	L,H,H,M,H	H	I	0.06	0.91	0.03	0.14	0.85	0.01	0.45	0.54	0.01	0.5	0.49	0.01	0.05	0.05	0.05	0.58	0.64	0.57	0.21	0.18	0.29	0.22	0.2	0.14
2019/10/01	L,L,L,M,H,H,H,L	M,L,L,L,M,L,L	L	L,M,H,H,M	M	B	0.06	0.91	0.02	0.14	0.85	0.01	0.46	0.53	0.01	0.5	0.49	0.01	0.05	0.05	0.05	0.56	0.61	0.55	0.2	0.17	0.28	0.24	0.23	0.18
2019/10/02	L,L,M,H,H,H,L,M	L,L,L,M,L,L,L	L	M,H,H,M,M	M	B	0.06	0.92	0.02	0.14	0.85	0.01	0.46	0.53	0.01	0.5	0.49	0.01	0.05	0.05	0.05	0.54	0.59	0.53	0.21	0.18	0.29	0.25	0.25	0.2
2019/10/03	L,M,H,H,H,L,M,M	L,L,M,L,L,L,L	L	L,M,M,M,M	M	B	0.06	0.93	0.02	0.14	0.85	0.01	0.47	0.52	0.01	0.51	0.48	0.01	0.05	0.05	0.05	0.52	0.57	0.5	0.24	0.2	0.31	0.24	0.26	0.2
2019/10/04	M,H,H,H,L,M,M,H	L,M,L,L,L,L,L	M	L,M,M,M,H	H	B	0.07	0.9	0.03	0.14	0.85	0.01	0.47	0.52	0.01	0.51	0.48	0.01	0.05	0.05	0.05	0.52	0.56	0.48	0.25	0.22	0.34	0.23	0.25	0.19
2019/10/05	H,H,H,L,M,M,M,H	M,L,L,L,L,L,M	M	L,H,M,M,H	H	B	0.07	0.91	0.03	0.14	0.85	0.01	0.47	0.52	0.01	0.51	0.48	0.01	0.05	0.05	0.05	0.51	0.56	0.48	0.27	0.23	0.35	0.22	0.24	0.18
2019/11/14	H,H,H,M,M,M,M,H	H,H,H,M,H,H,H	H	M,M,H,M,H	H	C	0.1	0.87	0.03	0.16	0.83	0.01	0.45	0.54	0.01	0.52	0.48	0.01	0.05	0.05	0.05	0.81	0.85	0.86	0.19	0.16	0.12	0.05	0.05	0.05
2019/11/15	H,H,M,M,M,M,M,H	H,H,M,H,H,H,H	H	H,M,H,M,H	H	C	0.1	0.87	0.03	0.17	0.82	0.01	0.47	0.52	0.01	0.52	0.47	0.01	0.05	0.05	0.05	0.76	0.81	0.83	0.24	0.2	0.15	0.05	0.05	0.05
2019/11/16	H,M,M,M,M,M,M,H	H,M,H,H,H,H,H	H	H,M,H,M,H	H	C	0.09	0.88	0.03	0.19	0.8	0.01	0.5	0.49	0.01	0.55	0.44	0.01	0.05	0.05	0.05	0.7	0.76	0.8	0.3	0.25	0.19	0.05	0.05	0.05
2019/11/17	M,M,M,M,M,M,H,H	M,H,H,H,H,H,H	H	H,M,H,H,H	H	C	0.1	0.87	0.02	0.2	0.79	0.01	0.52	0.47	0.01	0.57	0.42	0.01	0.05	0.05	0.05	0.66	0.71	0.75	0.34	0.3	0.24	0.05	0.05	0.05
2019/11/18	M,M,M,M,M,H,H,H	H,H,H,H,H,H,H	H	H,M,H,H,H	H	C	0.11	0.87	0.02	0.21	0.79	0.01	0.53	0.46	0.01	0.58	0.41	0.01	0.05	0.05	0.05	0.63	0.68	0.72	0.37	0.33	0.28	0.05	0.05	0.05
2019/11/19	M,M,M,M,H,H,H,H	H,H,H,H,H,H,H	H	H,H,H,H,H	H	C	0.12	0.86	0.02	0.21	0.78	0.01	0.53	0.46	0.01	0.58	0.41	0.01	0.05	0.05	0.05	0.62	0.66	0.7	0.38	0.35	0.3	0.05	0.05	0.05
2019/11/20	M,M,M,H,H,H,H,H	H,H,H,H,H,H,H	H	H,H,H,H,H	H	C	0.13	0.86	0.02	0.21	0.78	0.01	0.53	0.46	0.01	0.58	0.41	0.01	0.05	0.05	0.05	0.62	0.65	0.69	0.38	0.36	0.31	0.05	0.05	0.05
2019/11/21	M,M,H,H,H,H,H,H	H,H,H,H,H,H,H	L	H,H,H,H,H	H	I	0.13	0.85	0.02	0.2	0.78	0.01	0.53	0.46	0.01	0.58	0.41	0.01	0.05	0.05	0.05	0.64	0.66	0.69	0.36	0.35	0.31	0.05	0.05	0.05
2019/11/22	M,H,H,H,H,H,H,H	H,H,H,H,H,H,L	L	H,H,H,H,H	H	I	0.13	0.85	0.02	0.2	0.79	0.01	0.52	0.47	0.01	0.57	0.42	0.01	0.05	0.05	0.05	0.67	0.68	0.7	0.33	0.33	0.3	0.05	0.05	0.05
2019/11/23	H,H,H,H,H,H,H,H	H,H,H,H,H,L,L	H	H,H,H,H,M	M	B	0.13	0.85	0.02	0.2	0.79	0.01	0.52	0.47	0.01	0.57	0.42	0.01	0.05	0.05	0.05	0.7	0.71	0.73	0.3	0.3	0.27	0.05	0.05	0.05
2019/11/24	H,H,H,H,H,H,H,H	H,H,H,H,L,L,H	H	H,H,H,M,H	H	C	0.13	0.85	0.02	0.2	0.79	0.01	0.52	0.47	0.01	0.57	0.42	0.01	0.05	0.05	0.05	0.73	0.74	0.76	0.27	0.27	0.24	0.05	0.05	0.05
2019/11/25	H,H,H,H,H,H,H,H	H,H,H,L,L,H,H	H	L,H,H,H,H	H	C	0.13	0.85	0.02	0.2	0.79	0.01	0.52	0.47	0.01	0.57	0.42	0.01	0.05	0.05	0.05	0.76	0.77	0.78	0.24	0.24	0.22	0.05	0.05	0.05
2019/11/26	H,H,H,H,H,H,H,H	H,H,L,L,H,H,H	H	L,H,H,H,H	H	C	0.13	0.85	0.02	0.2	0.79	0.01	0.52	0.47	0.01	0.57	0.42	0.01	0.05	0.05	0.05	0.78	0.79	0.8	0.22	0.22	0.2	0.05	0.05	0.05
2019/11/27	H,H,H,H,H,H,H,H	H,L,L,H,H,H,H	H	H,H,H,H,H	H	C	0.13	0.85	0.02	0.2	0.79	0.01	0.52	0.47	0.01	0.57	0.42	0.01	0.05	0.05	0.05	0.8	0.81	0.82	0.2	0.2	0.18	0.05	0.05	0.05
2019/11/28	H,H,H,H,H,H,H,H	L,L,H,H,H,H,H	H	H,H,H,H,H	H	C	0.13	0.85	0.02	0.2	0.79	0.01	0.52	0.47	0.01	0.57	0.42	0.01	0.05	0.05	0.05	0.82	0.83	0.84	0.18	0.18	0.16	0.05	0.05	0.05
2019/11/29	H,H,H,H,H,H,H,H	L,H,H,H,H,H,H	H	H,H,H,H,H	H	C	0.13	0.85	0.02	0.2	0.79	0.01	0.52	0.47	0.01	0.57	0.42	0.01	0.05	0.05	0.05	0.84	0.85	0.86	0.16	0.16	0.14	0.05	0.05	0.05
2019/11/30	H,H,H,H,H,H,H,H	H,H,H,H,H,H,H	H	H,H,H,H,H	H	C	0.13	0.85	0.02	0.2	0.79	0.01	0.52	0.47	0.01	0.57	0.42	0.01	0.05	0.05	0.05	0.86	0.86	0.87	0.14	0.14	0.13	0.05	0.05	0.05
2019/12/01	H,H,H,H,H,H,H,H	H,H,H,H,H,H,H	H	H,H,H,H,H	H	C	0.13	0.85	0.02	0.2	0.79	0.01	0.52	0.47	0.01	0.57	0.42	0.01	0.05	0.05	0.05	0.87	0.87	0.88	0.13	0.13	0.12	0.05	0.05	0.05
2019/12/02	H,H,H,H,H,H,H,H	H,H,H,H,H,H,H	H	H,H,H,H,H	H	C	0.13	0.85	0.02	0.2	0.79	0.01	0.52	0.47	0.01	0.57	0.42	0.01	0.05	0.05	0.05	0.88	0.88	0.89	0.12	0.12	0.11	0.05	0.05	0.05
2019/12/03	H,H,H,H,H,H,H,H	H,H,H,H,H,H,H	H	H,H,H,H,H	H	C	0.13	0.85	0.02	0.2	0.79	0.01	0.52	0.47	0.01	0.57	0.42	0.01	0.05	0.05	0.05	0.89	0.89	0.9	0.11	0.11	0.1	0.05	0.05	0.05
2019/12/04	H,H,H,H,H,H,H,Cl	H,H,H,H,H,H,H	L	H,H,H,H,M	M	B	0.13	0.85	0.02	0.2	0.79	0.01	0.52	0.47	0.01	0.57	0.42	0.01	0.06	0.06	0.06	0.89	0.89	0.9	0.1	0.1	0.09	0.05	0.05	0.05
2019/12/05	H,H,H,H,H,H,Cl,Cl	H,H,H,H,H,H,L	L	H,H,H,M,M	M	B	0.13	0.85	0.02	0.2	0.79	0.01	0.52	0.47	0.01	0.57	0.42	0.01	0.08	0.08	0.08	0.87	0.88	0.89	0.09	0.09	0.08	0.05	0.05	0.05
2019/12/06	H,H,H,H,H,Cl,Cl,Cl	H,H,H,H,H,L,L	L	H,H,M,M,M	M	B	0.13	0.85	0.02	0.2	0.79	0.01	0.52	0.47	0.01	0.57	0.42	0.01	0.11	0.11	0.11	0.84	0.86	0.87	0.08	0.08	0.07	0.05	0.05	0.05
2019/12/07	H,H,H,H,Cl,Cl,Cl,Cl	H,H,H,H,L,L,L	L	H,M,M,M,M	M	B	0.13	0.85	0.02	0.2	0.79	0.01	0.52	0.47	0.01	0.57	0.42	0.01	0.15	0.15	0.14	0.8	0.83	0.84	0.07	0.07	0.06	0.05	0.05	0.05
2019/12/08	H,H,H,Cl,Cl,Cl,Cl,Cl	H,H,H,L,L,L,L	L	L,H,M,M,M	M	B	0.13	0.85	0.02	0.2	0.79	0.01	0.52	0.47	0.01	0.57	0.42	0.01	0.2	0.2	0.18	0.75	0.79	0.8	0.06	0.06	0.05	0.05	0.05	0.05
2019/12/09	H,H,Cl,Cl,Cl,Cl,Cl,Cl	H,H,L,L,L,L,L	L	L,H,M,M,M	M	B	0.13	0.85	0.02	0.2	0.79	0.01	0.52	0.47	0.01	0.57	0.42	0.01	0.26	0.25	0.23	0.7	0.74	0.76	0.05	0.05	0.05	0.05	0.05	0.05
2019/12/10	H,Cl,Cl,Cl,Cl,Cl,Cl,Cl	H,L,L,L,L,L,L	L	L,H,M,M,M	M	B	0.12	0.86	0.02	0.21	0.79	0.01	0.53	0.46	0.01	0.58	0.41	0.01	0.33	0.31	0.28	0.63	0.68	0.71	0.05	0.05	0.05	0.05	0.05	0.05
2019/12/11	Cl,Cl,Cl,Cl,Cl,Cl,Cl,H	L,L,L,L,L,L,L	M	L,H,M,M,H	H	B	0.13	0.86	0.02	0.21	0.78	0.01	0.54	0.45	0.01	0.59	0.4	0.01	0.39	0.37	0.34	0.58	0.63	0.65	0.05	0.05	0.05	0.05	0.05	0.05
2019/12/12	Cl,Cl,Cl,Cl,Cl,Cl,M,H	L,L,L,L,L,L,M	H	L,H,M,M,H	H	C	0.13	0.85	0.02	0.21	0.78	0.01	0.55	0.45	0.01	0.6	0.4	0.01	0.43	0.41	0.38	0.54	0.58	0.6	0.06	0.06	0.06	0.05	0.05	0.05

Appendix N: Sample of Baum-Welch related Geyser solar heating calculations

Date	ObsWeather	GeyserTempRec	Geyser Temp PredRec	GeyserTemp PredPast	Geyser Temp Pred	Predict Correct	PrH	PrM	PrL	TrHH	TrHM	TrHl	TrMH	TrMM	TrML	TrLH	TrLM	TrLL	EmCH	EMCM	EmCL	EmHH	EmHM	EmHL	EmMH	EmMM	EmML	EmLH	EmLM	EmLL
2018/01/22	H,H,H,H, H,H,H,H	H,H,H,H, H,H,H	H	H,H,H, H,H	H	C	0.64	0.24	0.11	0.81	0.11	0.08	0.39	0.54	0.07	0.45	0.26	0.29	0.01	0.07	0.19	0.8	0.37	0.25	0.18	0.51	0.38	0.01	0.05	0.19
2018/01/23	H,H,H,H, H,H,H,H	H,H,H,H, H,H,H	H	H,H,H, H,H	H	C	0.69	0.21	0.09	0.84	0.1	0.07	0.45	0.49	0.06	0.51	0.24	0.25	0.01	0.06	0.15	0.87	0.49	0.4	0.11	0.41	0.3	0.01	0.04	0.15
2018/01/24	H,H,H,H, H,H,H,M	H,H,H,H, H,H,H	M	H,H,H, H,H	H	B	0.72	0.19	0.08	0.85	0.1	0.07	0.46	0.48	0.06	0.53	0.24	0.24	0.01	0.05	0.12	0.9	0.52	0.44	0.08	0.4	0.31	0.01	0.02	0.12
2018/01/25	H,H,H,H, H,H,M,M	H,H,H,H, H,H,M	H	H,H,H, H,M	M	B	0.75	0.17	0.07	0.84	0.11	0.07	0.43	0.51	0.06	0.5	0.26	0.25	0.01	0.04	0.1	0.92	0.47	0.44	0.06	0.47	0.36	0.01	0.02	0.1
2018/01/26	H,H,H,H, H,M,M,M	H,H,H,H, H,M,H	H	H,H,H, M,M	M	B	0.78	0.15	0.06	0.82	0.12	0.07	0.37	0.57	0.06	0.44	0.3	0.26	0.01	0.02	0.08	0.9	0.4	0.42	0.08	0.55	0.43	0.01	0.02	0.08
2018/01/27	H,H,H,H, M,M,M,L	H,H,H,H, M,H,H	L	H,M,M,M,M	M	B	0.81	0.13	0.05	0.79	0.14	0.08	0.32	0.61	0.07	0.38	0.32	0.3	0.01	0.02	0.06	0.85	0.34	0.37	0.12	0.59	0.44	0.02	0.05	0.12
2018/01/28	H,H,H,M,M, M,L,M	H,H,H,M,H ,H,L	H	M,M,M,M,M	M	B	0.84	0.11	0.04	0.76	0.17	0.09	0.27	0.65	0.08	0.33	0.35	0.32	0.01	0.02	0.05	0.8	0.28	0.31	0.17	0.63	0.46	0.02	0.07	0.18
2018/01/29	H,H,M,M,M,L,M,M	H,H,M,H,H ,L,H	M	H,H,M,M, M	M	C	0.86	0.1	0.04	0.71	0.21	0.1	0.23	0.69	0.08	0.28	0.38	0.34	0.01	0.02	0.04	0.72	0.22	0.25	0.25	0.67	0.48	0.02	0.08	0.23
2018/01/30	H,M,M,M,L,M,M,M	H,M,H,H,L,H,M	M	H,L M,M,M	M	C	0.87	0.09	0.04	0.64	0.26	0.11	0.2	0.72	0.08	0.24	0.41	0.35	0.01	0.02	0.02	0.61	0.16	0.2	0.36	0.73	0.51	0.02	0.08	0.27
2018/01/31	M,M,M,L M,M,M,M	M,H,H,L,H,M,M	H	L,M,M ,M,M	M	B	0.84	0.12	0.04	0.59	0.3	0.12	0.18	0.74	0.08	0.21	0.43	0.36	0.01	0.02	0.02	0.44	0.11	0.15	0.53	0.78	0.52	0.02	0.08	0.3
2018/02/21	M,Cl,H,H, H,M,H,M	H,L,H, H,H,H,L	L	H,H,H, H,M	M	B	0.76	0.16	0.07	0.67	0.21	0.11	0.41	0.49	0.09	0.38	0.32	0.3	0.02	0.21	0.12	0.64	0.3	0.61	0.32	0.47	0.24	0.02	0.02	0.02
2018/02/22	Cl,H,H,H ,M,H,M,M	L,H,H, H,H,L,L	L	H,H,H, M,M	M	B	0.67	0.24	0.09	0.68	0.21	0.11	0.43	0.48	0.09	0.39	0.31	0.3	0.02	0.21	0.12	0.65	0.3	0.6	0.31	0.47	0.26	0.02	0.02	0.02
2018/02/23	H,H,H,M,H ,M,M,L	H,H,H,H, L,L,L	L	H,H,M,M, M	M	B	0.7	0.21	0.09	0.68	0.21	0.11	0.43	0.48	0.09	0.39	0.31	0.3	0.02	0.17	0.09	0.65	0.3	0.57	0.3	0.48	0.31	0.03	0.05	0.03
2018/02/24	H,H,M,H,M, M,L,M	H,H,H,L ,L,L,L	H	L,M,M ,M,M	M	B	0.72	0.19	0.09	0.67	0.22	0.11	0.42	0.5	0.09	0.38	0.32	0.3	0.02	0.14	0.06	0.62	0.27	0.49	0.32	0.51	0.37	0.03	0.08	0.08
2018/02/25	H,M,H,M,M,L,M,H	H,H,L, L,L,L,H	H	L,M,M ,M,H	H	C	0.74	0.17	0.09	0.66	0.23	0.11	0.41	0.51	0.09	0.37	0.33	0.3	0.02	0.11	0.04	0.57	0.25	0.43	0.37	0.53	0.4	0.03	0.1	0.12
2018/02/26	M,H,M,M,L,M,H,M	H,L,L ,L,L,H,H	M	L,L,M,H,H	H	B	0.73	0.18	0.09	0.65	0.24	0.11	0.4	0.52	0.09	0.36	0.34	0.3	0.02	0.09	0.02	0.5	0.22	0.38	0.44	0.56	0.44	0.03	0.13	0.16
2018/02/27	H,M,M,L ,M,H,M,M	L,L ,L,L,H ,H,M	M	L,M,H ,H,M	M	C	0.75	0.16	0.09	0.63	0.25	0.11	0.39	0.53	0.09	0.35	0.35	0.3	0.02	0.07	0.02	0.48	0.19	0.33	0.48	0.6	0.47	0.02	0.13	0.18
2018/02/28	M,M,L,M, H,M,M,H	L,L ,L,H,H,M, M	M	H,H,H, M,H	H	B	0.74	0.17	0.09	0.62	0.26	0.11	0.39	0.53	0.09	0.35	0.35	0.3	0.02	0.06	0.02	0.45	0.16	0.28	0.51	0.64	0.49	0.02	0.14	0.2
2018/03/01	M,L,M,H ,M,M,H,H	L,L ,H,H,M,M,M	H	H,H,M,H ,H	H	C	0.72	0.19	0.1	0.62	0.26	0.11	0.4	0.52	0.09	0.36	0.34	0.3	0.02	0.04	0.02	0.46	0.18	0.26	0.49	0.63	0.48	0.02	0.15	0.24
2018/03/02	L,M,H,M ,M,H,H,H	L,H,H, M,M,M,H	H	M,M,H,H, H	H	C	0.62	0.24	0.15	0.64	0.25	0.1	0.42	0.5	0.09	0.38	0.33	0.29	0.02	0.02	0.02	0.51	0.21	0.26	0.44	0.62	0.43	0.02	0.15	0.29
2018/03/03	M,H,M,M,H,H ,H,H	H,H,M,M,M,H ,H	H	M,H,H,H ,H	H	C	0.62	0.25	0.14	0.66	0.23	0.09	0.45	0.48	0.08	0.41	0.32	0.28	0.02	0.02	0.02	0.57	0.26	0.32	0.39	0.6	0.43	0.02	0.11	0.22
2018/03/04	H,M,M,H,H, H,H,H	H,M,M,M,H,H ,H	H	M,H,H,H ,H	H	C	0.64	0.23	0.13	0.68	0.22	0.08	0.47	0.46	0.08	0.43	0.31	0.27	0.02	0.02	0.02	0.66	0.32	0.39	0.3	0.58	0.41	0.02	0.08	0.18
2018/03/05	M,M,H,H,H, H,H,H	M,M,M,H,H,H ,H	H	H,H,H, H,H	H	C	0.61	0.26	0.13	0.71	0.2	0.07	0.49	0.44	0.08	0.45	0.3	0.26	0.02	0.02	0.02	0.72	0.36	0.42	0.24	0.57	0.41	0.02	0.06	0.14
2018/03/06	M,H,H,H,H ,H,H,H	M,M,H,H,H, H,H	H	H,H,H, H,H	H	C	0.58	0.29	0.13	0.74	0.18	0.06	0.53	0.41	0.08	0.49	0.28	0.25	0.02	0.02	0.03	0.78	0.43	0.48	0.18	0.51	0.38	0.02	0.05	0.11
2018/03/07	H,H,H,H, H,H,H,M	M,H,H,H,H ,H,H	L	H,H,H, H,H	H	I	0.61	0.27	0.12	0.75	0.17	0.06	0.54	0.4	0.08	0.51	0.27	0.24	0.02	0.02	0.02	0.83	0.49	0.54	0.13	0.45	0.35	0.02	0.04	0.09
2018/03/08	H,H,H,H, H,H,M,Cl	H,H,H,H, H,H,L	L	H,H,H, H,M	M	B	0.64	0.25	0.11	0.76	0.17	0.06	0.54	0.39	0.08	0.51	0.27	0.24	0.03	0.05	0.05	0.85	0.51	0.54	0.1	0.42	0.34	0.02	0.03	0.07
2018/03/09	H,H,H,H, H,M,Cl,H	H,H,H,H, H,L,L	H	H,H,H, M,L	L	I	0.66	0.23	0.1	0.76	0.17	0.06	0.54	0.39	0.08	0.51	0.27	0.24	0.03	0.08	0.07	0.87	0.51	0.53	0.07	0.39	0.34	0.02	0.02	0.06
2018/03/10	H,H,H,H, M,Cl,H,M	H,H,H,H, L,L,H	M	H,M,M,H, M	M	C	0.69	0.21	0.09	0.74	0.19	0.06	0.51	0.41	0.08	0.48	0.28	0.25	0.03	0.1	0.08	0.88	0.48	0.49	0.06	0.4	0.37	0.02	0.02	0.05
2018/03/11	H,H,H,M,Cl,H,M,M	H,H,H,L ,L,H,M	M	L,M,L ,M,M	M	C	0.72	0.19	0.08	0.7	0.22	0.07	0.47	0.45	0.09	0.44	0.3	0.27	0.03	0.11	0.09	0.88	0.41	0.44	0.07	0.46	0.44	0.02	0.02	0.04
2018/04/07	M,Cl,Cl,Cl,M,M,M,H	M,M,M,M,M,L,H	H	M,M,M,M,H	H	C	0.61	0.28	0.1	0.53	0.22	0.25	0.37	0.49	0.15	0.25	0.26	0.48	0.24	0.07	0.46	0.2	0.2	0.06	0.53	0.71	0.45	0.02	0.02	0.02
2018/04/08	Cl,Cl,Cl,M,M,M,H,H	M,M,M,M,L,H,H	H	M,M,M,H,H	H	C	0.62	0.24	0.13	0.53	0.22	0.25	0.37	0.5	0.14	0.25	0.26	0.48	0.26	0.07	0.48	0.24	0.22	0.06	0.48	0.69	0.44	0.02	0.02	0.02
2018/04/09	Cl,Cl,M,M ,M,H,H,M	M,M,M,L, H,H,H	L	L,L,H,H,H	H	I	0.63	0.2	0.16	0.53	0.23	0.24	0.36	0.52	0.13	0.25	0.28	0.47	0.24	0.07	0.45	0.29	0.23	0.06	0.44	0.68	0.46	0.02	0.02	0.02
2018/04/10	Cl,M,M,M,H,H,M,M	M,M,L,H,H,H,L	M	H,H,H,H,L	L	B	0.63	0.17	0.19	0.53	0.25	0.23	0.35	0.54	0.12	0.25	0.31	0.44	0.2	0.06	0.37	0.34	0.23	0.06	0.44	0.69	0.55	0.02	0.02	0.02
2018/04/11	M,M,M,H,H,M ,M,M	M,L,H,H,H,L,M	M	H,H,H,L,M	M	C	0.61	0.19	0.19	0.53	0.26	0.22	0.34	0.55	0.11	0.25	0.33	0.42	0.15	0.05	0.28	0.37	0.21	0.06	0.46	0.72	0.64	0.02	0.02	0.02
2018/04/12	M,M,H,H,M,M ,M,L	L,H,H,H,L,M,M	L	H,H,L,M,M	M	B	0.59	0.21	0.2	0.53	0.27	0.21	0.34	0.56	0.11	0.25	0.34	0.41	0.11	0.03	0.21	0.4	0.2	0.06	0.45	0.73	0.69	0.03	0.03	0.05
2018/04/13	M,H,H,M,M,M ,L,L	H,H,H,L,M,M,L	M	L,L,M,M,M	M	C	0.58	0.22	0.2	0.53	0.27	0.21	0.34	0.56	0.11	0.25	0.33	0.42	0.08	0.02	0.16	0.43	0.19	0.06	0.43	0.72	0.64	0.06	0.07	0.14
2018/04/14	H,H,M,M,M,L,L,M	H,H,L, M,M,L,M	M	M,M,M,M,M	M	C	0.63	0.2	0.17	0.52	0.27	0.21	0.32	0.56	0.12	0.23	0.33	0.44	0.06	0.02	0.12	0.48	0.16	0.06	0.39	0.7	0.6	0.07	0.12	0.22
2018/04/15	H,M,M,M,L,L,M,M	H,L,M,M ,L,M,M	M	M,M,L,M,M	M	C	0.68	0.18	0.14	0.49	0.29	0.23	0.29	0.57	0.13	0.21	0.33	0.46	0.05	0.02	0.09	0.48	0.11	0.05	0.4	0.71	0.59	0.07	0.15	0.27
2018/04/16	M,M,M,L L,M,M,M	L,M,M,L,M,M,M	L	L,L,M,M,M	M	B	0.65	0.2	0.15	0.46	0.3	0.25	0.27	0.59	0.14	0.19	0.33	0.48	0.04	0.02	0.07	0.38	0.07	0.03	0.5	0.73	0.59	0.08	0.17	0.3
2018/04/17	M,M,L,L,M,M,M,M	M,M,L,M, M,M,L	M	M,M,M,M,M	M	C	0.63	0.21	0.15	0.43	0.31	0.27	0.25	0.6	0.15	0.17	0.33	0.49	0.03	0.02	0.06	0.3	0.05	0.02	0.58	0.75	0.59	0.09	0.18	0.33
2018/04/18	M,L,L,M,M,M,M,M	M,L,M,M, M,L,M	M	M,M,M,M,M	M	C	0.62	0.21	0.16	0.4	0.32	0.29	0.24	0.61	0.15	0.16	0.34	0.5	0.02	0.02	0.05	0.24	0.03	0.02	0.64	0.77	0.57	0.1	0.18	0.36
2018/04/19	L,L,M,M,M,M,M,H	L,M,M,M, L,M,M	M	M,M,M,M,H	H	B	0.57	0.21	0.21	0.41	0.32	0.28	0.25	0.61	0.14	0.16	0.35	0.49	0.02	0.02	0.03	0.25	0.02	0.02	0.6	0.78	0.56	0.12	0.18	0.38
2018/04/20	L,M,M,M, M,M,H,M	M,M,M,L,M,M,M	M	L,L,M,H,M	M	C	0.53	0.21	0.25	0.42	0.33	0.26	0.26	0.61	0.13	0.17	0.36	0.47	0.02	0.02	0.02	0.26	0.04	0.02	0.59	0.77	0.61	0.12	0.16	0.34
2018/04/21	M,M,M,M,M,H,M,M	M,M,L,M, M,M,M	M	M,M,H,M,M	M	C	0.52	0.22	0.25	0.42	0.34	0.25	0.27	0.61	0.12	0.18	0.37	0.45	0.02	0.02	0.02	0.26	0.06	0.02	0.62	0.81	0.69	0.1	0.1	0.27
2018/04/22	M,M,M,M,H,M, M,H	M,L,M,M, M,M,M	M	M,H,M,M,H	H	B	0.51	0.23	0.25	0.43	0.34	0.23	0.29	0.6	0.11	0.2	0.38	0.42	0.02	0.02	0.02	0.29	0.09	0.02	0.61	0.82	0.74	0.08	0.06	0.21
2018/04/23	M,M,M,H,M,M, H,H	L,M,M,M, M,M,M	H	M,M,M,H,H	H	C	0.5	0.24	0.25	0.46	0.33	0.21	0.31	0.59	0.1	0.22	0.39	0.39	0.02	0.02	0.02	0.34	0.16	0.02	0.57	0.77	0.78	0.06	0.04	0.17

Date	ObsWeather	GeyserTempRec	Geyser Temp PredRec	GeyserTemp PredPast	Geyser Temp Pred	Predict Correct	PrH	PrM	PrL	TrHH	TrHM	TrHl	TrMH	TrMM	TrML	TrLH	TrLM	TrLL	EmCH	EMCM	EmCL	EmHH	EmHM	EmHL	EmMH	EmMM	EmML	EmLH	EmLM	EmLL
2018/05/12	H,H,H,H,H,H,H,M	M,H,M,M,M,H,M	M	M,H,H,H,M	M	C	0.51	0.35	0.15	0.73	0.19	0.08	0.5	0.45	0.05	0.35	0.38	0.27	0.02	0.02	0.02	0.93	0.77	0.68	0.02	0.19	0.27	0.02	0.02	0.02
2018/05/13	H,H,H,H,H,H,M,H	H,M,M,M,H,M,M	M	M,H,H,M,H	H	B	0.52	0.34	0.14	0.72	0.2	0.08	0.5	0.45	0.05	0.35	0.38	0.27	0.02	0.02	0.02	0.93	0.76	0.68	0.02	0.2	0.27	0.02	0.02	0.02
2018/05/14	H,H,H,H,H,M,H,Cl	M,M,M,H,M,M,M	L	H,H,M,H,M	M	B	0.53	0.33	0.13	0.71	0.21	0.08	0.49	0.45	0.05	0.35	0.38	0.27	0.04	0.03	0.04	0.92	0.74	0.67	0.02	0.2	0.27	0.02	0.02	0.02
2018/05/15	H,H,H,H,M,H,Cl,L	M,M,H,M,M,M,L	L	M,M,H,H,M	M	B	0.55	0.32	0.12	0.7	0.22	0.09	0.49	0.45	0.05	0.35	0.38	0.27	0.06	0.03	0.05	0.88	0.72	0.64	0.02	0.22	0.27	0.04	0.03	0.04
2018/05/16	H,H,H,M,H,Cl,L,L	M,H,M,M,M,L,L	L	M,H,H,L,L	L	C	0.57	0.31	0.11	0.69	0.22	0.09	0.49	0.45	0.05	0.35	0.37	0.27	0.07	0.03	0.06	0.82	0.68	0.57	0.02	0.23	0.27	0.08	0.05	0.11
2018/05/17	H,H,M,H,Cl,L,L,M	H,M,M,M,L,L,L	L	M,H,L,L,L	L	C	0.58	0.31	0.11	0.66	0.23	0.11	0.48	0.46	0.05	0.33	0.37	0.29	0.09	0.02	0.06	0.77	0.62	0.47	0.02	0.28	0.27	0.12	0.07	0.21
2018/05/18	H,M,H,Cl,L,L,M,H	M,M,M,L,L,L,L	L	L,L,L,L,L	L	C	0.57	0.32	0.11	0.63	0.24	0.13	0.48	0.46	0.05	0.32	0.37	0.31	0.12	0.02	0.06	0.7	0.59	0.37	0.02	0.32	0.28	0.15	0.07	0.29
2018/05/19	M,H,Cl,L,L,M,H,M	M,M,L,L,L,L,L	L	L,L,L,L,L	L	C	0.47	0.4	0.13	0.6	0.24	0.16	0.48	0.46	0.05	0.3	0.37	0.33	0.15	0.02	0.05	0.64	0.51	0.31	0.03	0.4	0.3	0.18	0.07	0.34
2018/05/20	H,Cl,L,L,M,H,M,L	M,L,L,L,L,L,L	L	L,M,L,L,L	L	C	0.5	0.39	0.12	0.58	0.23	0.19	0.48	0.45	0.06	0.28	0.36	0.36	0.18	0.02	0.04	0.57	0.48	0.25	0.02	0.41	0.32	0.22	0.08	0.39
2018/05/21	Cl,L,L,M,H,M,L,L	L,L,L,L,L,L,L	M	L,M,M,L,L	L	B	0.58	0.32	0.11	0.55	0.22	0.23	0.47	0.45	0.07	0.26	0.35	0.39	0.2	0.03	0.03	0.48	0.43	0.2	0.03	0.44	0.32	0.29	0.1	0.45
2018/05/22	L,L,M,H,M,L,L,H	L,L,L,L,L,L,M	M	L,M,L,L,M	M	C	0.6	0.27	0.13	0.52	0.22	0.25	0.47	0.45	0.08	0.25	0.35	0.4	0.16	0.02	0.02	0.44	0.4	0.18	0.03	0.43	0.32	0.37	0.14	0.48
2018/05/23	L,M,H,M,L,L,H,H	L,L,L,L,L,M,M	M	L,L,L,M,H	H	B	0.6	0.24	0.16	0.5	0.24	0.26	0.46	0.45	0.08	0.25	0.35	0.4	0.13	0.02	0.02	0.44	0.41	0.18	0.03	0.4	0.34	0.41	0.16	0.46
2018/05/24	M,H,M,L,L,H,H,H	L,L,L,L,M,M,M	M	L,L,M,H,H	H	B	0.5	0.31	0.19	0.5	0.25	0.25	0.46	0.45	0.08	0.26	0.35	0.39	0.1	0.02	0.02	0.48	0.45	0.2	0.04	0.36	0.37	0.39	0.17	0.41
2018/05/25	H,M,L,L,H,H,H,H	L,L,L,M,M,M,M	M	M,H,H,H,H	H	B	0.51	0.31	0.18	0.5	0.26	0.24	0.46	0.45	0.08	0.27	0.35	0.38	0.07	0.02	0.02	0.52	0.52	0.23	0.04	0.28	0.34	0.37	0.17	0.4
2018/06/30	H,H,H,L,M,H,H,M	M,M,M,L,M,M,L	M	L,M,M,H,L	L	B	0.29	0.4	0.31	0.39	0.26	0.36	0.43	0.52	0.06	0.07	0.68	0.26	0.02	0.02	0.02	0.7	0.75	0.31	0.09	0.21	0.44	0.19	0.02	0.24
2018/07/01	H,H,L,M,H,H,M,H	M,M,L,M,M,L,M	M	M,H,H,L,M	M	C	0.29	0.43	0.27	0.38	0.25	0.38	0.44	0.51	0.06	0.07	0.69	0.25	0.02	0.02	0.02	0.7	0.72	0.31	0.09	0.24	0.42	0.19	0.02	0.25
2018/07/02	H,L,M,H,H,M,H,M	M,L,M,M,L,M,M	L	M,H,L,M,M	M	B	0.33	0.43	0.24	0.36	0.24	0.41	0.44	0.51	0.06	0.07	0.69	0.25	0.02	0.02	0.02	0.69	0.67	0.3	0.1	0.28	0.42	0.19	0.02	0.26
2018/07/03	L,M,H,H,M,H,M,L	L,M,M,L,M,M,L	L	L,M,M,M,L	L	C	0.35	0.35	0.29	0.34	0.24	0.43	0.44	0.5	0.06	0.07	0.68	0.26	0.02	0.02	0.02	0.63	0.62	0.27	0.12	0.34	0.42	0.22	0.02	0.29
2018/07/04	M,H,H,M,H,M,L,L	M,M,L,M,M,L,L	L	M,H,M,L,L	L	C	0.3	0.37	0.32	0.34	0.23	0.44	0.45	0.5	0.06	0.08	0.66	0.27	0.02	0.02	0.02	0.59	0.58	0.26	0.13	0.38	0.41	0.26	0.02	0.31
2018/07/05	H,H,M,H,M,L,L,M	M,L,M,M,L,L,L	L	M,M,L,L,M	M	B	0.31	0.39	0.29	0.33	0.22	0.45	0.45	0.49	0.06	0.08	0.64	0.29	0.02	0.02	0.02	0.56	0.56	0.25	0.14	0.4	0.39	0.28	0.02	0.34
2018/07/06	H,M,H,M,L,L,M,L	L,M,M,L,L,L,L	L	L,L,L,M,L	L	C	0.32	0.4	0.27	0.32	0.21	0.47	0.47	0.47	0.06	0.08	0.61	0.32	0.02	0.02	0.02	0.49	0.51	0.22	0.15	0.44	0.38	0.33	0.02	0.38
2018/07/07	M,H,M,L,L,M,L,L	M,M,L,L,L,L,L	L	L,L,M,L,L	L	C	0.28	0.43	0.28	0.32	0.19	0.49	0.5	0.44	0.06	0.09	0.58	0.35	0.02	0.02	0.02	0.41	0.45	0.15	0.15	0.5	0.37	0.41	0.02	0.46
2018/07/08	H,M,L,L,M,L,L,L	M,L,L,L,L,L,L	L	L,M,L,L,L	L	C	0.28	0.46	0.25	0.33	0.17	0.5	0.54	0.4	0.06	0.1	0.53	0.39	0.02	0.02	0.02	0.35	0.44	0.1	0.14	0.51	0.32	0.49	0.02	0.56
2018/07/09	M,L,L,M,L,L,L,L	L,L,L,L,L,L,L	L	L,L,L,L,L	L	C	0.25	0.51	0.23	0.34	0.14	0.51	0.61	0.33	0.06	0.11	0.47	0.43	0.02	0.03	0.02	0.27	0.35	0.06	0.12	0.59	0.28	0.57	0.03	0.64
2018/07/10	L,L,M,L,L,L,L,L	L,L,L,L,L,L,L	L	L,L,L,L,L	L	C	0.31	0.41	0.27	0.35	0.12	0.52	0.67	0.27	0.06	0.12	0.42	0.47	0.02	0.02	0.02	0.22	0.27	0.04	0.1	0.62	0.22	0.65	0.08	0.72
2018/07/11	L,M,L,L,L,L,L,M	L,L,L,L,L,L,L	L	L,L,L,L,M	M	B	0.32	0.34	0.33	0.35	0.11	0.52	0.7	0.24	0.06	0.12	0.4	0.49	0.02	0.02	0.02	0.18	0.22	0.02	0.09	0.65	0.2	0.71	0.1	0.76
2018/07/12	M,L,L,L,L,L,M,M	L,L,L,L,L,L,L	L	L,L,L,M,M	M	B	0.28	0.42	0.3	0.36	0.1	0.53	0.69	0.25	0.06	0.12	0.38	0.51	0.02	0.02	0.02	0.14	0.18	0.02	0.1	0.7	0.22	0.74	0.1	0.75
2018/07/13	L,L,L,L,L,M,M,M	L,L,L,L,L,L,L	L	L,L,M,M,M	M	B	0.32	0.36	0.32	0.36	0.09	0.54	0.63	0.31	0.06	0.12	0.36	0.53	0.02	0.02	0.02	0.11	0.14	0.02	0.11	0.72	0.23	0.75	0.11	0.73
2018/07/14	L,L,L,L,M,M,M,L	L,L,L,L,L,L,L	L	L,M,M,M,L	L	C	0.36	0.3	0.33	0.36	0.08	0.55	0.6	0.34	0.06	0.12	0.35	0.54	0.02	0.02	0.02	0.09	0.11	0.02	0.12	0.75	0.26	0.77	0.11	0.7
2018/07/15	L,L,L,M,M,M,L,L	L,L,L,L,L,L,L	L	L,M,M,L,L	L	C	0.41	0.25	0.33	0.36	0.07	0.56	0.57	0.37	0.06	0.11	0.35	0.54	0.02	0.02	0.02	0.07	0.09	0.02	0.12	0.78	0.29	0.78	0.11	0.67
2018/07/16	L,L,M,M,M,L,L,L	L,L,L,L,L,L,L	L	L,M,L,L,L	L	C	0.45	0.21	0.33	0.36	0.06	0.56	0.55	0.4	0.06	0.11	0.35	0.54	0.02	0.02	0.02	0.06	0.07	0.02	0.12	0.79	0.31	0.79	0.11	0.65
2018/07/17	L,M,M,M,L,L,L,L	L,L,L,L,L,L,L	L	L,L,L,L,L	L	C	0.45	0.18	0.36	0.37	0.06	0.56	0.53	0.42	0.06	0.11	0.35	0.54	0.02	0.02	0.02	0.05	0.06	0.02	0.11	0.8	0.33	0.81	0.11	0.63
2018/07/18	M,M,M,L,L,L,L,L	L,L,L,L,L,L,L	L	L,L,L,L,L	L	C	0.38	0.23	0.38	0.4	0.05	0.54	0.52	0.43	0.06	0.12	0.33	0.55	0.02	0.02	0.02	0.04	0.05	0.02	0.11	0.81	0.34	0.82	0.11	0.62
2018/07/19	M,M,L,L,L,L,L,M	L,L,L,L,L,L,L	L	L,L,L,L,L	L	C	0.32	0.3	0.38	0.42	0.05	0.53	0.53	0.42	0.06	0.13	0.32	0.56	0.02	0.02	0.02	0.02	0.04	0.02	0.11	0.82	0.32	0.84	0.11	0.64
2018/07/20	M,L,L,L,L,L,M,M	L,L,L,L,L,L,L	L	L,L,L,L,L	L	C	0.28	0.37	0.35	0.44	0.04	0.52	0.54	0.41	0.06	0.13	0.31	0.57	0.02	0.02	0.02	0.02	0.02	0.02	0.11	0.84	0.31	0.84	0.11	0.65
2018/07/21	L,L,L,L,L,M,M,M	L,L,L,L,L,L,L	L	L,L,L,L,M	M	B	0.34	0.31	0.35	0.45	0.04	0.51	0.48	0.47	0.06	0.13	0.31	0.57	0.02	0.02	0.02	0.02	0.02	0.02	0.11	0.84	0.29	0.84	0.11	0.67
2018/07/22	L,L,L,L,M,M,M,M	L,L,L,L,L,L,L	M	L,M,M,M,M	M	C	0.4	0.26	0.35	0.45	0.04	0.51	0.41	0.55	0.06	0.12	0.32	0.57	0.02	0.02	0.02	0.02	0.02	0.02	0.11	0.85	0.31	0.84	0.1	0.65
2018/07/23	L,L,L,M,M,M,M,M	L,L,L,L,L,L,M	M	L,M,M,M,M	M	C	0.45	0.21	0.34	0.44	0.04	0.52	0.34	0.62	0.05	0.11	0.34	0.56	0.02	0.02	0.02	0.02	0.02	0.02	0.12	0.88	0.34	0.83	0.07	0.62
2018/07/24	L,L,M,M,M,M,M,M	L,L,L,L,L,M,M	M	L,M,M,M,M	M	C	0.49	0.17	0.34	0.4	0.05	0.55	0.28	0.68	0.05	0.09	0.38	0.53	0.02	0.02	0.02	0.02	0.02	0.02	0.12	0.9	0.39	0.82	0.06	0.57
2018/07/25	L,M,M,M,M,M,M,H	L,L,L,L,M,M,M	M	L,M,M,M,H	H	B	0.49	0.14	0.37	0.34	0.08	0.58	0.24	0.72	0.05	0.07	0.43	0.49	0.03	0.02	0.02	0.08	0.03	0.02	0.13	0.9	0.48	0.77	0.04	0.48
2018/07/26	M,M,M,M,M,M,H,M	L,L,L,M,M,M,M	M	M,M,M,H,M	M	C	0.41	0.19	0.4	0.3	0.1	0.6	0.21	0.75	0.05	0.06	0.47	0.46	0.02	0.02	0.02	0.17	0.07	0.02	0.2	0.89	0.62	0.61	0.02	0.34
2018/07/27	M,M,M,M,M,H,M,M	L,L,M,M,M,M,M	M	M,M,H,M,M	M	C	0.35	0.23	0.42	0.26	0.12	0.62	0.19	0.78	0.05	0.05	0.5	0.44	0.02	0.02	0.02	0.22	0.09	0.02	0.27	0.86	0.7	0.48	0.02	0.25
2018/07/28	M,M,M,M,H,M,M,H	L,M,M,M,M,M,M	M	M,H,M,M,H	H	B	0.31	0.25	0.44	0.24	0.14	0.62	0.18	0.79	0.05	0.05	0.53	0.42	0.03	0.02	0.02	0.32	0.12	0.02	0.28	0.84	0.76	0.38	0.02	0.19
2018/07/29	M,M,M,H,M,M,H,H	M,M,M,M,M,M,M	M	M,M,M,H,H	H	B	0.27	0.27	0.46	0.26	0.15	0.59	0.18	0.79	0.04	0.05	0.56	0.39	0.02	0.02	0.02	0.42	0.19	0.02	0.26	0.77	0.8	0.3	0.02	0.15
2018/07/30	M,M,H,M,M,H,H,H	M,M,M,M,M,M,M	M	M,M,M,H,H	H	B	0.23	0.28	0.48	0.3	0.16	0.54	0.18	0.79	0.04	0.05	0.59	0.36	0.02	0.02	0.02	0.51	0.28	0.02	0.22	0.68	0.82	0.24	0.02	0.12

Date	ObsWeather	GeyserTempRec	Geyser Temp PredRec	GeyserTemp PredPast	Geyser Temp Pred	Predict Correct	PrH	PrM	PrL	TrHH	TrHM	TrHl	TrMH	TrMM	TrML	TrLH	TrLM	TrLL	EmCH	EMC M	EmCL	EmHH	EmHM	EmHL	EmMH	EmMM	EmML	EmLH	EmLM	EmLL
2018/08/17	L,L,M,H,H,H,H,H	L,L,M,M,M,M,M	M	M,M,H,H,H	H	B	0.38	0.38	0.25	0.39	0.15	0.46	0.19	0.78	0.02	0.04	0.59	0.37	0.02	0.02	0.02	0.48	0.74	0.02	0.05	0.18	0.25	0.45	0.06	0.7
2018/08/18	L,M,H,H,H,H,H,H	L,M,M,M,M,M,M	L	M,M,H,H,H	H	I	0.38	0.32	0.31	0.36	0.19	0.45	0.16	0.81	0.02	0.03	0.63	0.34	0.02	0.02	0.02	0.49	0.78	0.02	0.04	0.16	0.28	0.45	0.04	0.67
2018/08/19	M,H,H,H,H,H,H,M	M,M,M,M,M,M,L	M	M,M,H,M,L	L	B	0.31	0.34	0.36	0.34	0.24	0.42	0.14	0.84	0.02	0.03	0.7	0.27	0.02	0.02	0.02	0.55	0.79	0.05	0.07	0.16	0.41	0.35	0.02	0.52
2018/08/20	H,H,H,H,H,H,M,H	M,M,M,M,M,L,M	M	M,M,M,L,M	M	C	0.29	0.42	0.3	0.33	0.29	0.38	0.13	0.86	0.02	0.03	0.75	0.22	0.02	0.02	0.02	0.62	0.81	0.14	0.08	0.15	0.44	0.27	0.02	0.4
2018/08/21	H,H,H,H,H,M,H,H	M,M,M,M,L,M,M	M	M,M,L,M,H	H	B	0.27	0.48	0.26	0.32	0.33	0.35	0.12	0.87	0.02	0.03	0.78	0.19	0.02	0.02	0.02	0.68	0.82	0.24	0.07	0.14	0.43	0.22	0.02	0.31
2018/08/22	H,H,H,H,M,H,H,M	M,M,M,L,M,M,M	M	L,M,M,H,M	M	C	0.26	0.52	0.23	0.31	0.34	0.35	0.11	0.87	0.02	0.03	0.8	0.17	0.02	0.02	0.02	0.72	0.8	0.29	0.08	0.16	0.45	0.18	0.02	0.24
2018/08/23	H,H,H,M,H,H,M,H	M,M,L,M,M,M,M	M	M,M,M,M,M	M	C	0.25	0.55	0.2	0.3	0.35	0.35	0.11	0.87	0.02	0.03	0.81	0.16	0.02	0.02	0.02	0.76	0.76	0.34	0.08	0.2	0.46	0.14	0.02	0.18
2018/08/24	H,H,M,H,H,M,H,H	M,L,M,M,M,M,M	M	M,M,M,M,H	H	B	0.25	0.57	0.18	0.29	0.35	0.35	0.11	0.87	0.02	0.03	0.82	0.15	0.02	0.02	0.02	0.79	0.73	0.39	0.08	0.23	0.46	0.11	0.02	0.13
2018/08/25	H,M,H,H,M,H,H,M	L,M,M,M,M,M,M	M	M,M,M,H,M	M	C	0.26	0.57	0.17	0.27	0.35	0.38	0.11	0.87	0.02	0.03	0.82	0.15	0.02	0.02	0.02	0.8	0.69	0.4	0.09	0.27	0.48	0.09	0.02	0.1
2018/08/26	M,H,H,M,H,H,M,H	M,M,M,M,M,M,M,M	M	M,M,M,M,H	H	B	0.22	0.58	0.2	0.26	0.35	0.39	0.11	0.87	0.02	0.03	0.83	0.14	0.02	0.02	0.02	0.79	0.65	0.37	0.11	0.31	0.52	0.07	0.02	0.08
2018/08/27	H,H,M,H,H,M,H,L	M,M,M,M,M,M,M,M	M	M,M,M,M,M	M	C	0.23	0.59	0.18	0.26	0.34	0.4	0.12	0.86	0.02	0.03	0.83	0.14	0.02	0.02	0.02	0.78	0.66	0.38	0.1	0.3	0.49	0.1	0.02	0.11
2018/08/28	H,M,H,H,M,H,L,M	M,M,M,M,M,M,M,M	M	M,M,M,M,M	M	C	0.24	0.59	0.17	0.25	0.33	0.43	0.13	0.85	0.02	0.03	0.83	0.14	0.02	0.02	0.02	0.75	0.62	0.35	0.1	0.31	0.51	0.13	0.05	0.12
2018/08/29	M,H,H,M,H,L,M,M	M,M,M,M,M,M,M,M	M	M,M,M,M,M	M	C	0.21	0.6	0.19	0.24	0.32	0.45	0.13	0.84	0.02	0.03	0.83	0.14	0.02	0.02	0.02	0.68	0.54	0.3	0.12	0.36	0.56	0.17	0.08	0.12
2018/08/30	H,H,M,H,L,M,M,M	M,M,M,M,M,M,M,M	M	M,M,M,M,M	M	C	0.22	0.61	0.17	0.23	0.31	0.46	0.13	0.84	0.02	0.03	0.82	0.15	0.02	0.02	0.02	0.66	0.49	0.28	0.12	0.41	0.57	0.19	0.09	0.12
2018/08/31	H,M,H,L,M,M,M,H	M,M,M,M,M,M,M,M	M	M,M,M,M,H	H	B	0.24	0.61	0.16	0.21	0.3	0.48	0.13	0.84	0.02	0.03	0.82	0.15	0.02	0.02	0.02	0.65	0.42	0.26	0.12	0.46	0.59	0.2	0.1	0.13
2018/09/01	M,H,L,M,M,M,H,H	M,M,M,M,M,M,M,M	M	M,M,M,H,H	H	B	0.21	0.63	0.17	0.21	0.29	0.48	0.13	0.84	0.02	0.03	0.82	0.15	0.02	0.02	0.02	0.64	0.39	0.24	0.12	0.49	0.61	0.21	0.11	0.13
2018/10/14	H,H,H,H,H,H,Cl,L	M,H,H,H,H,H,L	L	H,H,H,H,L	L	C	0.19	0.65	0.16	0.6	0.17	0.22	0.35	0.63	0.02	0.07	0.8	0.14	0.03	0.06	0.11	0.93	0.55	0.51	0.02	0.25	0.27	0.02	0.13	0.11
2018/10/15	H,H,H,H,H,Cl,L,M	H,H,H,H,H,L,L	H	H,H,H,L,M	M	B	0.22	0.64	0.15	0.6	0.16	0.23	0.34	0.64	0.02	0.06	0.8	0.15	0.03	0.08	0.16	0.93	0.53	0.48	0.02	0.25	0.22	0.02	0.14	0.15
2018/10/16	H,H,H,H,Cl,L,M,H	H,H,H,H,L,L,H	H	H,L,M,M,H	H	C	0.25	0.62	0.14	0.59	0.15	0.24	0.34	0.64	0.02	0.05	0.8	0.16	0.02	0.08	0.21	0.91	0.52	0.42	0.04	0.25	0.18	0.02	0.15	0.19
2018/10/17	H,H,H,Cl,L,M,H,H	H,H,H,L,L,H,H	H	L,M,M,H,H	H	C	0.28	0.6	0.13	0.58	0.14	0.26	0.34	0.64	0.02	0.04	0.79	0.17	0.02	0.08	0.27	0.89	0.51	0.37	0.06	0.25	0.14	0.02	0.16	0.22
2018/10/18	H,H,Cl,L,M,H,H,H	H,H,L,L,H,H,H	H	L,M,H,H,H	H	C	0.32	0.57	0.11	0.58	0.13	0.28	0.34	0.64	0.02	0.04	0.78	0.18	0.02	0.08	0.31	0.88	0.5	0.33	0.07	0.25	0.12	0.02	0.18	0.24
2018/10/19	H,Cl,L,M,H,H,H,H	H,L,L,H,H,H,H	H	H,H,H,H,H	H	C	0.38	0.51	0.1	0.58	0.12	0.29	0.34	0.64	0.02	0.04	0.78	0.19	0.02	0.08	0.34	0.87	0.49	0.31	0.08	0.25	0.1	0.02	0.19	0.25
2018/10/20	Cl,L,M,H,H,H,H,H	L,L,H,H,H,H,H	H	H,H,H,H,H	H	C	0.32	0.49	0.19	0.62	0.11	0.26	0.34	0.64	0.02	0.04	0.78	0.19	0.02	0.09	0.36	0.87	0.47	0.3	0.09	0.25	0.07	0.02	0.2	0.27
2018/10/21	L,M,H,H,H,H,H,M	L,H,H,H,H,H,H	M	H,H,H,H,H	H	B	0.26	0.52	0.22	0.65	0.11	0.23	0.35	0.63	0.02	0.04	0.8	0.17	0.02	0.07	0.28	0.85	0.46	0.31	0.11	0.27	0.08	0.02	0.2	0.33
2018/10/22	M,H,H,H,H,H,M,M	H,H,H,H,H,H,M	M	H,H,H,H,M	M	C	0.24	0.57	0.19	0.67	0.11	0.21	0.36	0.62	0.02	0.04	0.81	0.15	0.02	0.06	0.22	0.84	0.43	0.38	0.12	0.36	0.14	0.02	0.15	0.26
2018/10/23	H,H,H,H,H,M,M,M	H,H,H,H,H,M,M	M	H,H,H,M,M	M	C	0.28	0.55	0.17	0.67	0.12	0.2	0.34	0.64	0.02	0.04	0.82	0.14	0.02	0.05	0.18	0.85	0.39	0.42	0.11	0.45	0.19	0.02	0.11	0.21
2018/10/24	H,H,H,H,M,M,M,M	H,H,H,H,M,M,M	H	H,M,M,M,M	M	B	0.33	0.51	0.15	0.66	0.13	0.2	0.31	0.67	0.02	0.04	0.83	0.13	0.02	0.03	0.14	0.85	0.33	0.45	0.11	0.56	0.24	0.02	0.08	0.18
2018/10/25	H,H,H,M,M,M,M,H	H,H,H,M,M,M,H	H	M,M,M,M,H	H	C	0.4	0.46	0.14	0.64	0.14	0.21	0.29	0.69	0.02	0.04	0.84	0.12	0.02	0.02	0.11	0.83	0.29	0.48	0.13	0.63	0.28	0.02	0.06	0.14
2018/10/26	H,H,M,M,M,M,H,H	H,H,M,M,M,H,H	H	M,M,M,H,H	H	C	0.45	0.41	0.13	0.62	0.15	0.22	0.28	0.7	0.02	0.04	0.85	0.12	0.02	0.02	0.09	0.82	0.25	0.51	0.14	0.69	0.3	0.02	0.04	0.11
2018/10/27	H,M,M,M,M,H,H,H	H,M,M,M,H,H,H	H	M,M,H,H,H	H	C	0.45	0.39	0.15	0.63	0.15	0.21	0.27	0.71	0.02	0.04	0.86	0.11	0.02	0.02	0.07	0.82	0.22	0.52	0.14	0.74	0.32	0.02	0.02	0.09
2018/10/28	M,M,M,M,H,H,H,H	M,M,M,H,H,H,H	H	H,H,H,H,H	H	C	0.37	0.47	0.15	0.67	0.13	0.19	0.27	0.71	0.02	0.04	0.85	0.11	0.02	0.02	0.06	0.82	0.18	0.52	0.14	0.78	0.35	0.02	0.02	0.07
2018/10/29	M,M,M,H,H,H,H,H	M,M,H,H,H,H,H	H	H,H,H,H,H	H	C	0.3	0.54	0.15	0.72	0.11	0.17	0.29	0.69	0.02	0.05	0.84	0.11	0.02	0.02	0.05	0.83	0.16	0.53	0.13	0.79	0.36	0.02	0.02	0.06
2018/10/30	M,M,H,H,H,H,H,H	M,H,H,H,H,H,H	H	H,H,H,H,H	H	C	0.25	0.6	0.14	0.76	0.09	0.15	0.34	0.65	0.02	0.07	0.82	0.11	0.02	0.02	0.04	0.84	0.15	0.56	0.12	0.8	0.35	0.02	0.02	0.05
2018/10/31	M,H,H,H,H,H,H,H	H,H,H,H,H,H,H	M	H,H,H,H,H	H	B	0.22	0.65	0.12	0.8	0.07	0.13	0.44	0.55	0.02	0.13	0.75	0.12	0.02	0.02	0.03	0.85	0.18	0.63	0.11	0.77	0.3	0.02	0.02	0.04
2018/11/01	H,H,H,H,H,H,H,Cl	H,H,H,H,H,H,M	L	H,H,H,H,H	H	I	0.31	0.57	0.11	0.82	0.06	0.12	0.51	0.48	0.02	0.18	0.7	0.12	0.03	0.06	0.09	0.88	0.32	0.65	0.07	0.6	0.24	0.02	0.02	0.03
2018/11/02	H,H,H,H,H,H,Cl,M	H,H,H,H,H,M,L	H	H,H,H,M,M	M	B	0.37	0.52	0.1	0.82	0.06	0.13	0.51	0.48	0.02	0.17	0.71	0.12	0.02	0.07	0.16	0.9	0.35	0.6	0.05	0.55	0.21	0.02	0.02	0.02
2018/11/03	H,H,H,H,H,Cl,M,M	H,H,H,H,M,L,H	M	H,H,L,M,M	M	C	0.43	0.47	0.09	0.8	0.06	0.15	0.46	0.52	0.02	0.15	0.73	0.12	0.02	0.07	0.24	0.88	0.34	0.54	0.07	0.56	0.19	0.02	0.02	0.03
2018/11/04	H,H,H,H,Cl,M,M,M	H,H,H,M,L,H,M	L	M,M,M,M,M	M	B	0.49	0.42	0.08	0.78	0.06	0.18	0.4	0.58	0.02	0.13	0.75	0.12	0.02	0.07	0.32	0.87	0.31	0.48	0.09	0.6	0.18	0.02	0.02	0.02
2018/12/12	H,H,H,M,M,M,M,H	H,H,H,H,M,H,H	H	H,H,M,M,H	H	C	0.65	0.25	0.09	0.69	0.05	0.26	0.5	0.48	0.02	0.25	0.65	0.11	0.02	0.02	0.02	0.73	0.42	0.52	0.22	0.53	0.42	0.02	0.02	0.02
2018/12/13	H,H,M,M,M,M,H,H	H,H,H,M,H,H,H	H	M,M,M,H,H	H	C	0.68	0.23	0.08	0.67	0.05	0.27	0.48	0.5	0.02	0.23	0.66	0.11	0.02	0.02	0.02	0.67	0.37	0.49	0.29	0.58	0.46	0.02	0.02	0.02
2018/12/14	H,M,M,M,M,H,H,H	H,H,M,H,H,H,H	H	H,H,H,H,H	H	C	0.69	0.22	0.08	0.66	0.05	0.28	0.47	0.51	0.02	0.22	0.67	0.11	0.02	0.02	0.02	0.63	0.33	0.47	0.33	0.62	0.48	0.02	0.02	0.02
2018/12/15	M,M,M,M,H,H,H,H	H,M,H,H,H,H,H	H	H,H,H,H,H	H	C	0.65	0.25	0.09	0.67	0.05	0.28	0.47	0.51	0.02	0.22	0.67	0.11	0.02	0.02	0.02	0.6	0.31	0.47	0.36	0.64	0.48	0.02	0.02	0.02
2018/12/16	M,M,M,H,H,H,H,H	M,H,H,H,H,H,H	H	H,H,H,H,H	H	C	0.61	0.28	0.1	0.68	0.05	0.27	0.48	0.5	0.02	0.24	0.66	0.11	0.02	0.02	0.02	0.63	0.31	0.49	0.33	0.64	0.46	0.02	0.02	0.02
2018/12/17	M,M,H,H,H,H,H,H	H,H,H,H,H,H,H	H	H,H,H,H,H	H	C	0.57	0.32	0.11	0.7	0.05	0.25	0.5	0.48	0.02	0.27	0.63	0.11	0.02	0.02	0.02	0.67	0.33	0.54	0.29	0.62	0.41	0.02	0.02	0.02
2018/12/18	M,H,H,H,H,H,H,H	H,H,H,H,H,H,H	H	H,H,H,H,H	H	C	0.55	0.35	0.11	0.72	0.04	0.23	0.54	0.44	0.02	0.31	0.59	0.11	0.02	0.02	0.02	0.74	0.38	0.62	0.23	0.57	0.33	0.02	0.02	0.02
2018/12/19	H,H,H,H,H,H,H,H	H,H,H,H,H,H,H	H	H,H,H,H,H	H	C	0.58	0.32	0.11	0.74	0.04	0.22	0.58	0.4	0.02	0.34	0.55	0.11	0.02	0.02	0.02	0.82	0.49	0.69	0.14	0.46	0.26	0.02	0.02	0.02

Appendix O: Sample of Geyser Usage Training Calculations

Date	Fin1Best	PredRec1Best	Correct1	Fin2Best	PredRec2Best	Correct2	Fin3Best	PredRec3Best	Correct3	Fin4Best	PredRec4Best	Correct4	Fin5Best	PredRec5Best	Correct5	Fin6Best	PredRec6Best	Correct6	Fin7Best	PredRec7Best	Correct7	Fin8Best	PredRec8Best	Correct8
2018/01/08	Small	None	H1	Small	Medium	I	Small	Medium	I	Medium	None	H2	Small	Small	C	Small	Small	C	Large	Small	H2	Small	Small	C
2018/01/09	Small	Small	C	Small	Medium	I	Medium	Medium	C	Large	Small	H2	Small	Medium	I	Small	Large	I	Large	Medium	H1	Small	Small	C
2018/01/10	Small	Small	C	Medium	Medium	C	Medium	Medium	C	Large	Small	H2	Small	Medium	I	Small	Large	I	Large	Large	C	Small	Small	C
2018/01/11	Small	Small	C	Medium	Medium	C	Medium	Small	H1	Large	Small	H2	Medium	Medium	C	Large	Large	C	Large	Large	C	Small	Small	C
2018/01/12	Small	Small	C	Medium	Medium	C	Medium	Small	H1	Large	Small	H2	Medium	Small	H1	Large	Medium	H1	Large	Large	C	Small	None	H1
2018/01/13	Small	None	H1	Medium	Small	H1	Medium	Small	H1	Large	Small	H2	Medium	Small	H1	Large	Medium	H1	Large	Medium	H1	Small	Small	C
2018/01/14	Small	None	H1	Medium	Medium	C	Medium	Small	H1	Large	Medium	H1	Medium	Small	H1	Large	Medium	H1	Large	Medium	H1	Small	Small	C
2018/01/15	Small	None	H1	Medium	Medium	C	Medium	Small	H1	Medium	Medium	C	Medium	Small	H1	Large	Small	H2	Large	Small	H2	Small	Small	C
2018/01/16	Small	Small	C	Medium	Medium	C	Medium	Medium	C	Medium	Medium	C	Medium	Small	H1	Large	Large	C	Large	Small	H2	Small	Small	C
2018/01/17	Small	Small	C	Medium	Medium	C	Medium	Medium	C	Medium	Medium	C	Small	Small	C	Large	Large	C	Large	Large	C	Small	Small	C
2018/01/29	Small	None	H1	Medium	Medium	C	Medium	Medium	C	Large	None	H3	Large	None	H3	Small	Medium	I	Large	Small	H2	Small	Small	C
2018/01/30	Small	None	H1	Medium	Medium	C	Medium	Medium	C	Small	Small	C	Medium	Small	H1	Small	Large	I	Large	Medium	H1	Small	Small	C
2018/01/31	None	None	C	Medium	Medium	C	Medium	Medium	C	Small	Small	C	Medium	Small	H1	Medium	Large	I	Large	Medium	H1	Small	Small	C
2018/02/01	None	None	C	Medium	Medium	C	Medium	Medium	C	Small	Small	C	Small	Small	C	Large	Large	C	Large	Large	C	Small	Medium	I
2018/02/02	None	None	C	Medium	Medium	C	Medium	Medium	C	Small	Small	C	Small	Small	C	Large	Medium	H1	Large	Large	C	Small	Medium	I
2018/02/03	None	Small	I	Medium	Medium	C	Medium	Medium	C	Small	Small	C	Small	Medium	I	Large	Medium	H1	Large	Large	C	Medium	Medium	C
2018/02/04	None	Small	I	Medium	Medium	C	Medium	Medium	C	Small	Medium	I	Small	Medium	I	Large	Medium	H1	Large	Medium	H1	Medium	Small	H1
2018/02/05	Small	Small	C	Medium	Medium	C	Medium	Large	I	Small	Medium	I	Medium	Medium	C	Large	Small	H2	Large	Medium	H1	Medium	Small	H1
2018/02/06	Small	Small	C	Medium	Medium	C	Medium	Large	I	Medium	Medium	C	Medium	Medium	C	Large	Medium	H1	Large	Small	H2	Medium	Small	H1
2018/02/07	Small	Small	C	Medium	Medium	C	Large	Large	C	Medium	Medium	C	Medium	Medium	C	Medium	Medium	C	Large	Medium	H1	Medium	Small	H1
2018/02/08	Small	Small	C	Medium	Medium	C	Large	Large	C	Medium	Small	H1	Medium	Small	H1	Medium	Medium	C	Large	Medium	H1	Medium	Small	H1
2018/02/09	Small	Small	C	Medium	Medium	C	Large	Large	C	Medium	Small	H1	Medium	Small	H1	Medium	Medium	C	Medium	Medium	C	Small	Small	C
2018/02/10	Small	None	H1	Medium	Medium	C	Large	Large	C	Medium	Medium	C	Medium	Small	H1	Medium	Medium	C	Medium	Medium	C	Small	Small	C
2018/02/11	Small	None	H1	Medium	Medium	C	Large	Large	C	Medium	Medium	C	Medium	Medium	C	Medium	Small	H1	Medium	Medium	C	Small	Small	C
2018/02/12	Small	Small	C	Medium	Medium	C	Large	Large	C	Medium	Medium	C	Medium	Large	I	Medium	Small	H1	Medium	Large	I	Small	Small	C
2018/02/13	Small	Small	C	Medium	Medium	C	Large	Large	C	Medium	Medium	C	Medium	Large	I	Medium	Small	H1	Medium	Large	I	Small	Small	C
2018/02/14	Small	Small	C	Medium	Medium	C	Large	Medium	H1	Medium	Small	H1	Large	Large	C	Medium	Small	H1	Large	Large	C	Small	Small	C
2018/02/15	Small	None	H1	Medium	Medium	C	Large	Medium	H1	Medium	Small	H1	Large	Medium	H1	Medium	Medium	C	Large	Large	C	Small	Small	C
2018/02/16	Small	None	H1	Medium	Medium	C	Large	Medium	H1	Medium	Small	H1	Large	Medium	H1	Medium	Medium	C	Large	Large	C	Small	Small	C
2018/03/25	Small	None	H1	Medium	Medium	C	Large	Medium	H1	Medium	Medium	C	Small	Small	C	Small	Small	C	Large	Large	C	Small	Small	C
2018/03/26	None	None	C	Medium	Medium	C	Large	Medium	H1	Medium	Medium	C	Small	Small	C	Small	Small	C	Large	Large	C	Small	Small	C
2018/03/27	None	None	C	Medium	Medium	C	Medium	Large	I	Medium	Medium	C	Small	Small	C	Small	Medium	I	Large	Large	C	Small	None	H1
2018/03/28	None	None	C	Medium	Medium	C	Large	Large	C	Medium	Medium	C	Small	Small	C	Small	Large	I	Large	Large	C	Small	None	H1
2018/03/29	None	None	C	Medium	Medium	C	Large	Large	C	Medium	Small	H1	Small	Medium	I	Medium	Large	I	Large	Medium	H1	Small	None	H1
2018/03/30	None	None	C	Medium	Small	H1	Large	Small	H2	Medium	Small	H1	Small	Medium	I	Large	Large	C	Large	Small	H2	Small	Small	C
2018/03/31	None	None	C	Medium	Small	H1	Large	Small	H2	Medium	Small	H1	Medium	Medium	C	Large	Small	H2	Large	Small	H2	Small	Small	C
2018/04/01	None	None	C	Medium	Small	H1	Large	Medium	H1	Medium	Small	H1	Medium	Small	H1	Large	Small	H2	Large	Large	C	Small	Small	C
2018/04/02	None	None	C	Medium	Small	H1	Large	Medium	H1	Medium	Medium	C	Medium	Small	H1	Large	Small	H2	Large	Large	C	Small	Small	C
2018/04/03	None	Small	I	Medium	Small	H1	Large	Medium	H1	Medium	Medium	C	Medium	Small	H1	Large	Medium	H1	Large	Large	C	Small	Small	C
2018/04/04	None	Small	I	Medium	Small	H1	Medium	Medium	C	Medium	Medium	C	Medium	Small	H1	Large	Medium	H1	Large	Large	C	Small	Small	C
2018/04/05	Small	Small	C	Small	Small	C	Medium	Small	H1	Medium	Medium	C	Medium	Small	H1	Medium	Medium	C	Large	Large	C	Small	Small	C
2018/04/06	Small	Small	C	Small	Small	C	Medium	Small	H1	Medium	Medium	C	Small	Small	C	Medium	Medium	C	Large	Medium	H1	Small	Small	C
2018/04/07	Small	Medium	I	Small	Small	C	Medium	Small	H1	Medium	Medium	C	Small	Small	C	Medium	Medium	C	Large	Medium	H1	Small	Small	C
2018/04/08	Small	Medium	I	Small	Small	C	Medium	Small	H1	Medium	Medium	C	Small	Small	C	Medium	Medium	C	Large	Medium	H1	Small	Medium	I
2018/04/09	Medium	Medium	C	Small	Small	C	Medium	Medium	C	Medium	Medium	C	Small	Small	C	Medium	Small	H1	Large	Large	C	Small	Medium	I
2018/04/10	Medium	None	H2	Small	Medium	I	Medium	Medium	C	Medium	Medium	C	Small	Small	C	Medium	Small	H1	Large	Large	C	Medium	Medium	C
2018/04/11	Medium	Small	H1	Small	Medium	I	Medium	Medium	C	Medium	Small	H1	Small	Small	C	Medium	Small	H1	Large	Large	C	Medium	Medium	C
2018/04/12	Medium	Small	H1	Medium	Medium	C	Medium	Medium	C	Medium	Small	H1	Small	Small	C	Medium	Small	H1	Large	Large	C	Medium	Medium	C
2018/04/13	Medium	Small	H1	Medium	Medium	C	Medium	Medium	C	Medium	Small	H1	Small	Medium	I	Medium	Small	H1	Large	Large	C	Medium	Small	H1
2018/05/26	None	None	C	Medium	Medium	C	Medium	Large	I	Medium	Large	I	Medium	Medium	C	Small	Small	C	Large	Large	C	Small	Small	C
2018/05/27	None	Small	I	Medium	Medium	C	Large	Large	C	Medium	Large	I	Medium	Medium	C	Small	Large	I	Large	Large	C	Small	Small	C
2018/05/28	None	Small	I	Medium	Medium	C	Large	Medium	H1	Large	Large	C	Medium	Medium	C	Small	Large	I	Large	Large	C	Small	Small	C
2018/05/29	Small	Small	C	Medium	Medium	C	Large	Medium	H1	Large	Medium	H1	Medium	Medium	C	Large	Large	C	Large	Large	C	Small	None	H1
2018/05/30	Small	None	H1	Medium	Medium	C	Large	Medium	H1	Large	Medium	H1	Medium	Medium	C	Large	Medium	H1	Large	Large	C	Small	Small	C
2018/05/31	Small	None	H1	Medium	Medium	C	Large	Medium	H1	Large	Medium	H1	Medium	Medium	C	Large	Medium	H1	Large	Large	C	Small	Small	C
2018/06/01	Small	None	H1	Medium	Medium	C	Large	Medium	H1	Large	Medium	H1	Medium	Medium	C	Large	Medium	H1	Large	Large	C	Small	Small	C
2018/06/02	Small	None	H1	Medium	Medium	C	Medium	Medium	C	Large	Small	H2	Medium	Medium	C	Large	Medium	H1	Large	Large	C	Small	Small	C
2018/06/03	Small	None	H1	Medium	Medium	C	Medium	Medium	C	Medium	Medium	C	Medium	Medium	C	Large	Small	H2	Large	Large	C	Small	Small	C
2018/06/04	None	None	C	Medium	Medium	C	Medium	Medium	C	Medium	Medium	C	Medium	Medium	C	Medium	Small	H1	Large	Large	C	Small	Small	C
2018/06/05	None	None	C	Medium	Medium	C	Medium	Medium	C	Medium	Medium	C	Medium	Medium	C	Medium	Small	H1	Large	Large	C	Small	None	H1
2018/06/06	None	None	C	Medium	Medium	C	Medium	Large	I	Medium	Medium	C	Medium	Medium	C	Medium	Small	H1	Large	Large	C	Small	None	H1
2018/06/07	None	None	C	Medium	Medium	C	Medium	Large	I	Medium	Medium	C	Medium	Medium	C	Medium	Small	H1	Large	Large	C	Small	Small	C
2018/06/08	None	None	C	Medium	Medium	C	Large	Large	C	Medium	Medium	C	Medium	Medium	C	Small	Small	C	Large	Large	C	Small	Medium	I
2018/06/09	None	Small	I	Medium	Medium	C	Large	Medium	H1	Medium	Small	H1	Medium	Medium	C	Small	Small	C	Large	Large	C	Small	Medium	I
2018/06/10	None	Small	I	Medium	Medium	C	Large	Medium	H1	Medium	Large	I	Medium	Small	H1	Small	Large	I	Large	Medium	H1	Medium	Medium	C
2018/06/11	Small	Small	C	Medium	Medium	C	Large	Small	H2	Medium	Large	I	Medium	Small	H1	Small	Large	I	Large	Medium	H1	Medium	Medium	C
2018/06/12	Small	Small	C	Medium	Medium	C	Large	Small	H2	Large	Large	C	Medium	Medium	C	Large	Large	C	Large	Large	C	Medium	Medium	C

Date	Fin1Best	PredRec1Best	Correct1	Fin2Best	PredRec2Best	Correct2	Fin3Best	PredRec3Best	Correct3	Fin4Best	PredRec4Best	Correct4	Fin5Best	PredRec5Best	Correct5	Fin6Best	PredRec6Best	Correct6	Fin7Best	PredRec7Best	Correct7	Fin8Best	PredRec8Best	Correct8
2018/07/09	Small	Small	C	Medium	Medium	C	Large	Large	C	Large	Large	C	Medium	Medium	C	Medium	Small	H1	Large	Large	C	Medium	Small	H1
2018/07/10	Small	None	H1	Medium	Medium	C	Large	Large	C	Large	Medium	H1	Medium	Medium	C	Medium	Small	H1	Large	Large	C	Medium	Small	H1
2018/07/11	Small	None	H1	Medium	Medium	C	Large	Medium	H1	Large	Medium	H1	Medium	Medium	C	Medium	Small	H1	Large	Large	C	Medium	Medium	C
2018/07/12	Small	None	H1	Medium	Medium	C	Large	Medium	H1	Large	Medium	H1	Medium	Medium	C	Medium	Small	H1	Large	Large	C	Medium	Medium	C
2018/07/13	Small	None	H1	Medium	Medium	C	Large	Medium	H1	Large	Medium	H1	Medium	Medium	C	Medium	Small	H1	Large	Medium	H1	Medium	Medium	C
2018/07/14	Small	None	H1	Medium	Medium	C	Large	Large	C	Large	Medium	H1	Medium	Large	I	Small	Medium	I	Large	Medium	H1	Medium	Medium	C
2018/07/15	Small	Small	C	Medium	Medium	C	Large	Large	C	Medium	Medium	C	Large	Large	C	Medium	Medium	C	Large	Medium	H1	Medium	Medium	C
2018/07/16	Small	Small	C	Medium	Medium	C	Large	Large	C	Medium	Large	I	Large	Large	C	Medium	Medium	C	Large	Large	C	Medium	Medium	C
2018/07/17	Small	Medium	I	Medium	Medium	C	Large	Medium	H1	Large	Large	C	Large	Medium	H1	Medium	Medium	C	Large	Large	C	Medium	Medium	C
2018/07/18	Medium	Medium	C	Medium	Large	I	Large	Medium	H1	Large	Large	C	Large	Medium	H1	Medium	Medium	C	Large	Large	C	Medium	Medium	C
2018/07/19	Medium	Medium	C	Medium	Large	I	Large	Medium	H1	Large	Medium	H1	Large	Small	H2	Medium	Medium	C	Large	Large	C	Medium	Medium	C
2018/07/20	Medium	Medium	C	Large	Large	C	Large	Medium	H1	Large	Medium	H1	Large	Small	H2	Medium	Large	I	Large	Large	C	Medium	Medium	C
2018/08/25	Small	Small	C	Medium	Medium	C	Medium	Medium	C	Medium	Small	H1	Small	Medium	I	Large	Small	H2	Large	Large	C	Medium	Small	H1
2018/08/26	Small	Medium	I	Medium	Medium	C	Medium	Medium	C	Medium	Medium	C	Medium	Medium	C	Large	Small	H2	Large	Medium	H1	Medium	Small	H1
2018/08/27	Small	Medium	I	Medium	Medium	C	Medium	Large	I	Medium	Medium	C	Medium	Medium	C	Large	Small	H2	Large	Large	C	Medium	Small	H1
2018/08/28	Medium	Medium	C	Medium	Medium	C	Medium	Large	I	Medium	Medium	C	Medium	Medium	C	Small	Small	C	Large	Large	C	Medium	Small	H1
2018/08/29	Medium	Small	H1	Medium	Medium	C	Large	Large	C	Medium	Medium	C	Medium	Medium	C	Small	Small	C	Large	Large	C	Medium	Small	H1
2018/08/30	Medium	None	H2	Medium	Medium	C	Large	Medium	H1	Medium	Medium	C	Medium	Small	H1	Small	Small	C	Large	Large	C	Medium	Small	H1
2018/08/31	Medium	None	H2	Medium	Medium	C	Large	Medium	H1	Medium	None	H2	Medium	Small	H1	Small	Small	C	Large	Medium	H1	Small	Small	C
2018/09/01	Medium	None	H2	Medium	Medium	C	Large	Medium	H1	Medium	Large	I	Medium	None	H2	Small	Small	C	Large	Small	H2	Small	Medium	I
2018/09/02	Medium	Small	H1	Medium	Medium	C	Large	Medium	H1	Medium	Large	I	Medium	Small	H1	Small	Small	C	Large	Small	H2	Medium	Medium	C
2018/09/03	Small	Small	C	Medium	Medium	C	Large	Medium	H1	Large	Large	C	Medium	Large	I	Small	Small	C	Large	Large	C	Medium	Medium	C
2018/09/04	Small	Small	C	Medium	Medium	C	Medium	Medium	C	Large	Medium	H1	Medium	Large	I	Small	Small	C	Large	Large	C	Medium	Medium	C
2018/09/05	Small	None	H1	Medium	Medium	C	Medium	Medium	C	Large	Medium	H1	Large	Large	C	Small	Small	C	Large	Large	C	Medium	Small	H1
2018/09/06	Small	None	H1	Medium	Medium	C	Medium	Medium	C	Large	Medium	H1	Large	Small	H2	Small	Medium	I	Large	Large	C	Medium	Small	H1
2018/09/07	Small	None	H1	Medium	Large	I	Medium	Medium	C	Large	Medium	H1	Large	Small	H2	Small	Medium	I	Large	Large	C	Medium	Medium	C
2018/09/08	Small	Small	C	Medium	Large	I	Medium	Medium	C	Large	Small	H2	Large	Small	H2	Medium	Medium	C	Large	Large	C	Medium	Medium	C
2018/09/09	Small	Small	C	Large	Large	C	Medium	Medium	C	Medium	Large	I	Large	Small	H2	Medium	Medium	C	Large	Medium	H1	Medium	Medium	C
2018/09/10	Small	Small	C	Large	Medium	H1	Medium	Large	I	Large	Large	C	Large	Small	H2	Medium	Large	I	Large	Large	C	Medium	Medium	C
2018/09/11	Small	Small	C	Large	Medium	H1	Medium	Large	I	Large	Large	C	Small	Small	C	Medium	Large	I	Large	Large	C	Medium	Small	H1
2018/09/12	Small	Small	C	Large	Medium	H1	Large	Large	C	Large	Medium	H1	Small	Medium	I	Large	Large	C	Large	Large	C	Medium	Small	H1
2018/09/13	Small	None	H1	Large	Medium	H1	Large	Medium	H1	Large	Small	H2	Small	Medium	I	Large	Large	C	Large	Large	C	Medium	Small	H1
2018/09/14	Small	None	H1	Large	Medium	H1	Large	Medium	H1	Large	Small	H2	Medium	Medium	C	Large	Large	C	Large	Large	C	Medium	Small	H1
2018/09/15	Small	Small	C	Medium	Medium	C	Large	Medium	H1	Large	Medium	H1	Medium	Small	H1	Large	Large	C	Large	Large	C	Medium	Small	H1
2018/09/16	Small	Small	C	Medium	Medium	C	Large	Medium	H1	Large	Medium	H1	Medium	Small	H1	Large	Small	H2	Large	Small	H2	Medium	Small	H1
2018/09/17	Small	Small	C	Medium	Medium	C	Large	Large	C	Medium	Medium	C	Medium	Small	H1	Large	Small	H2	Large	Large	C	Small	Small	C
2018/09/18	Small	Small	C	Medium	Medium	C	Large	Large	C	Medium	Medium	C	Medium	Small	H1	Large	Medium	H1	Large	Large	C	Small	Small	C
2018/09/19	Small	Small	C	Medium	Medium	C	Large	Large	C	Medium	Small	H1	Medium	None	H2	Large	Medium	H1	Large	Large	C	Small	Small	C
2018/09/20	Small	Small	C	Medium	Medium	C	Large	Medium	H1	Medium	Small	H1	Small	None	H1	Large	Medium	H1	Large	Large	C	Small	Small	C
2018/09/21	Small	Small	C	Medium	Medium	C	Large	Medium	H1	Medium	Small	H1	Small	None	H1	Medium	Small	H1	Large	Large	C	Small	Small	C
2018/10/29	Small	None	H1	Medium	Small	H1	Large	Large	C	Medium	Medium	C	Small	Small	C	Large	Large	C	Large	Medium	H1	Small	Small	C
2018/10/30	Small	None	H1	Medium	Medium	C	Large	Large	C	Medium	Medium	C	Small	Small	C	Large	Large	C	Large	Large	C	Small	Small	C
2018/10/31	Small	None	H1	Medium	Medium	C	Large	Large	C	Medium	Medium	C	Small	Small	C	Large	Small	H2	Large	Large	C	Small	Small	C
2018/11/01	None	Small	I	Medium	Medium	C	Large	Medium	H1	Medium	Medium	C	Small	Small	C	Large	Small	H2	Large	Large	C	Small	Small	C
2018/11/02	Small	Small	C	Medium	Medium	C	Large	Medium	H1	Medium	Medium	C	Small	Small	C	Large	Small	H2	Large	Large	C	Small	Small	C
2018/11/03	Small	Small	C	Medium	Medium	C	Large	Medium	H1	Medium	Medium	C	Small	Small	C	Large	Medium	H1	Large	Large	C	Small	Small	C
2018/11/04	Small	Small	C	Medium	Medium	C	Large	Medium	H1	Medium	Small	H1	Small	Small	C	Large	Medium	H1	Large	Large	C	Small	Small	C
2018/11/05	Small	Small	C	Medium	Small	H1	Large	Small	H2	Medium	Small	H1	Small	Small	C	Large	Medium	H1	Large	Large	C	Small	Small	C
2018/11/06	Small	Small	C	Medium	Medium	C	Large	Medium	H1	Medium	Small	H1	Small	Small	C	Medium	Small	H1	Large	Large	C	Small	Small	C
2018/11/07	Small	Small	C	Medium	Medium	C	Medium	Large	I	Medium	Small	H1	Small	None	H1	Medium	Medium	C	Large	Large	C	Small	Medium	I
2018/11/08	Small	Small	C	Medium	Medium	C	Large	Large	C	Medium	Medium	C	Small	None	H1	Medium	Medium	C	Large	Large	C	Small	Medium	I
2018/11/09	Small	Small	C	Medium	Medium	C	Large	Large	C	Medium	Medium	C	Small	Medium	I	Medium	Medium	C	Large	Large	C	Medium	Medium	C
2018/12/16	Small	Small	C	Medium	Large	I	Medium	Small	H1	Medium	Small	H1	Large	None	H3	Small	Small	C	Large	Small	H2	Large	Large	C
2018/12/17	Small	Small	C	Large	Large	C	Medium	None	H2	Medium	Small	H1	Large	Small	H2	Small	Small	C	Large	Medium	H1	Large	Small	H2
2018/12/18	Small	Small	C	Large	Small	H2	Small	None	H1	Medium	Medium	C	Large	Small	H2	Small	Large	I	Large	Large	C	Large	Small	H2
2018/12/19	Small	None	H1	Large	Small	H2	Small	Small	C	Medium	Medium	C	Large	Medium	H1	Small	Large	I	Large	Large	C	Large	Small	H2
2018/12/20	Small	Small	C	Large	Small	H2	Small	Small	C	Medium	Medium	C	Large	Medium	H1	Large	Large	C	Large	Large	C	Large	Large	C
2018/12/21	Small	Small	C	Large	Medium	H1	Small	Small	C	Medium	Medium	C	Medium	Medium	C	Large	Small	H2	Large	Large	C	Large	Large	C
2018/12/22	Small	Small	C	Large	Medium	H1	Small	Small	C	Medium	Medium	C	Medium	Small	H1	Large	Small	H2	Large	Large	C	Large	Large	C
2018/12/23	Small	Small	C	Medium	Medium	C	Small	Small	C	Medium	None	H2	Medium	Small	H1	Large	Small	H2	Large	Small	H2	Large	None	H3
2018/12/24	Small	Small	C	Medium	Small	H1	Small	None	H1	Medium	None	H2	Medium	Small	H1	Large	Small	H2	Large	Small	H2	Large	Small	H2
2018/12/25	Small	Small	C	Medium	Small	H1	Small	Small	C	Medium	None	H2	Medium	Small	H1	Large	Small	H2	Large	None	H3	Large	Small	H2
2018/12/26	Small	None	H1	Medium	Small	H1	Small	Small	C	Medium	None	H2	Medium	Small	H1	Small	Small	C	Large	None	H3	Large	Small	H2
2018/12/27	Small	None	H1	Medium	None	H2	Small	Small	C	Medium	None	H2	Small	Small	C	Small	None	H1	Large	None	H3	Large	None	H3
2018/12/28	Small	None	H1	Medium	None	H2	Small	None	H1	Medium	None	H2	Small	None	H1	Small	Small	C	Large	Small	H2	Small	Small	C
2018/12/29	Small	None	H1	Small	None	H1	Small	Small	C	None	Small	I	Small	None	H1	Small	Small	C	Small	Medium	I	Small	Small	C
2018/12/30	Small	Small	C	Small	Small	C	Small	Large	I	Small	Medium	I	Small	Small	C	Small	Medium	I	Medium	Large	I	Small	Small	C
2018/12/31	Small	Small	C	Small	Small	C	Large	Large	C	Medium	Medium	C	Small	Medium	I	Medium	Medium	C	Large	Large	C	Small	Small	C