iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** XII **Month of publication:** December 2021

DOI: https://doi.org/10.22214/ijraset.2021.39195

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue XII Dec 2021- Available at www.ijraset.com

A Bayesian Machine Learning Approach for Smart City

Ravi Prakash Malviya[1], Er. Paritosh Tripathi[2], Er. Vineet Kumar Singh[3], Vishal Sharma[4]
[1, 2, 3, 4]*IET, Dr. Rammanohar Lohia Avadh University, Ayodhya*

I. INTRODUCTION

Smart cities are one of the most active research fields today. It aims to provide solutions to problems in urban environments, in order to help humans live better and to increase productivity in cities. Fifty-five percent (55%) of the world's population live in urban areas, and the proportion is projected to reach 68% by 2050, adding 2.5 billion people to urban areas [1]. Furthermore, the current smart cities market size in 2020 is valued at USD $98.95 billion and is projected to reach USD $463.89 billion by 2027, demonstrating a Compound Annual GrowthRate (CAGR) of 24.7% [2]. These values demonstrate a demand and need

A large amount of the research around smart cities has also concentrated on deploying and using sensors [5], as well as on cloud and edge computing to leverage smart city applications, which includes Internet of Things (IoT) [6–8]. Furthermore, more research focused on handling big data and related analytics [9,10], and on developing services and middleware to facilitate data handling [11, 12]. In addition, to add intelligence to smart cities, the use of Artificial Intelligence (AI) methods have been explored in smart cities applications [13]. The use of AI has also been driven by the need to effectively and efficiently manage smart city infrastructure consisting of IoT, large data, networking and the cloud and smart city applications [14]. Smartcities can be broadly represented by five significant components as illustrated in Figure 1.1.

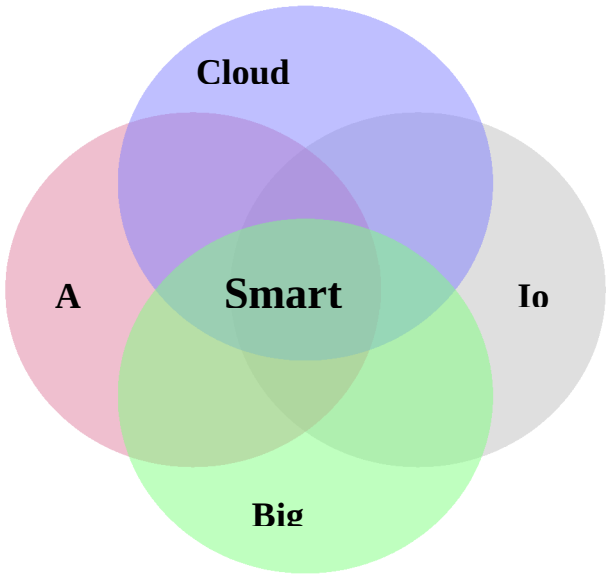


Figure 1.1: Venn Diagram of the Smart City Components

With the growth of IoT in smart cities, researchers began to focus on middleware to support sensors and related applications. The growing number of cities that are trying to exploit tech- nologies and applications that provide expanded services and enhanced living standards for itscitizens require software platforms to support development, deployment and use of smart city infrastructures.

The *computational infrastructure* for a smart city can be broadly viewed in terms of four mainlayers (see Figure 1.2). Starting from the bottom of the hierarchy, the first computing layeris the physical layer, also referred to as the device layer, which is comprised of the sensors that generate the raw data and the actuators which can be used to change settings of devices,

e.g. a thermostat. The layer above this is the IoT kernel layer, which collects data, monitors and controls the sensors and actuators. The third computing layer is the edge/fog layer, whichenables computing, i.e., applications and services, to take place at the network edge.

The last computing layer and at the top of the hierarchy is the cloud layer, which is the backbone of the IoT services and the initial

receiver of all the information sent by smart devices. These four computing layers are shown in Figure 1.2. This infrastructure provides the foundation forthe smart city applications and services. This can be monolithic software components, such as a database running in the cloud or fog computing system, or components of distributed applications and services or individual software components running on systems in the edge orIoT layer. One can think of the space of applications, services and their components as being overlayed on this computational infrastructure.

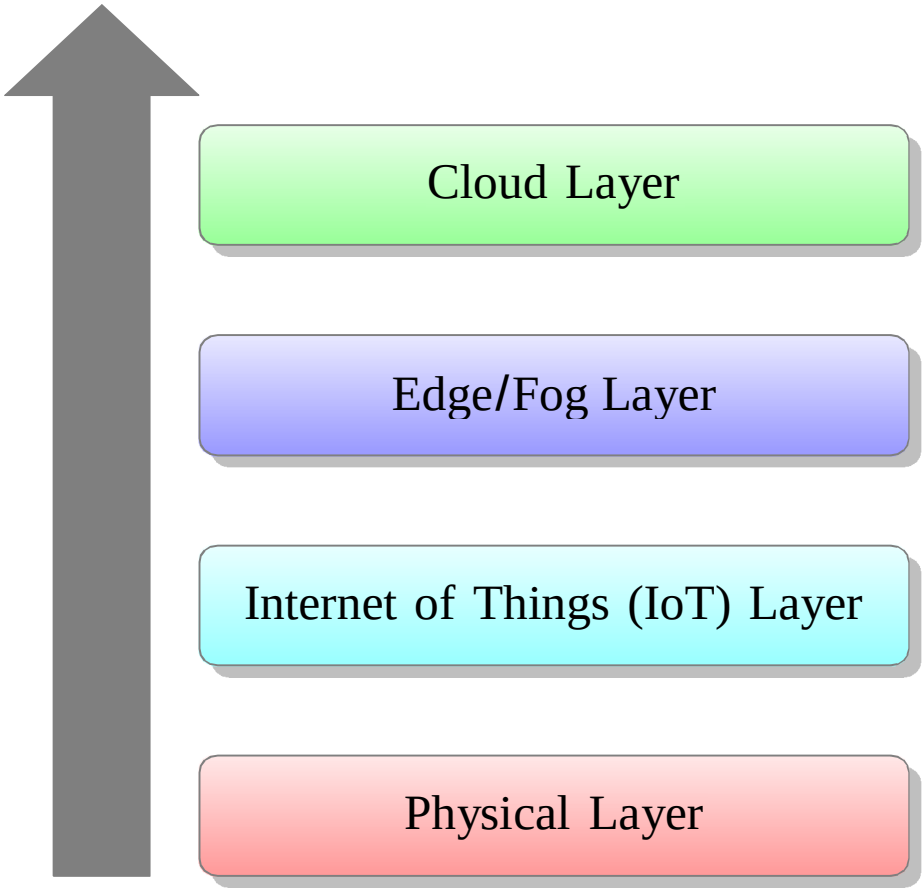


Figure 1.2: The Computing Layers of a Smart City

A. Problem Statement

Smart cities are capturing the attention of researchers and citizens around the world. Their emergence ignited extensive research on the approaches to support the development of smart city applications, specifically around handling sensors and their data, as well as on the associ- ated data analysis. With the complexity of smart city environments, namely the computational infrastructure and applications, ongoing research is needed to address the computational chal-lenges arising in smart city environments, particularly to help ensure efficient operations.

The physical layer consists of devices such as sensors and actuators, where different devices serve different purposes and come in various shapes and sizes. The sensors can measure dif- ferent aspects of the environment, such as temperature, humidity, air quality, or movement, whereas actuators can change their environment via movement and mechanism control. Smartcities heavily depend on sensors and actuators in order to get readings and enable applicationsto control houses, buildings, etc. The development of these types of smart city application re-quires methods and models that can facilitate the use of sensor data and enable use of actuators.

In summary, the model of the physical layer takes into consideration the set of integrated sen-sors and the readings they produce. The model provides a framework for integrating the vari-ables holding sensor readings, the role and use of intelligent agents, and the set of objectives to be met throughout during execution.

II. LITERATURE SURVEY

IN the early 1990s, the concept of smart city appeared for the first time. Technology, in-novation, and globalization have been emphasized upon in the urbanization process byresearchers [18]. Interest in smart cities has increased due to the growth of population in urbanareas and due to the many challenges being faced in order to meet the citizens' needs in areassuch as education, healthcare, security, water and many more. Smart cities aid in solving theseissues by embracing the help of real-time sensors, cloud computing and big data analytics. Inthis thesis, a physical layer model is proposed which can tackle the energy consumption chal-lenge faced in smart cities in order to boost the efficiency in a smart city. In this chapter, anoverview of smart cities and related work in the field of energy consumption management in a smart city are discussed.

A smart city's infrastructure consists of sensors, data platforms, applications and security com- ponents. These interconnected components and their provided services are illustrated in Figure

2.1. However, the adoption of smart city solutions and services come with other challenges, such as ensuring the existence of sufficient communication infrastructure, available funds, re-quired skills, and legal aspects to ensure that the privacy of citizens are not violated [19].

Stock Price Prediction Using Time Series

Figure 2.1: Smart City Components and Provided Services

Sensors have two types of network architectures: wired sensor networks and wireless sensor networks. The wired sensor network links the sensor directly to the device that will receive the input, by either using copper or fiber optic cables for the wiring [20, 21]. Wireless sensors are being widely used because of their efficiency, availability, increased mobility and low-cost nature. Wireless sensor networks are often preferred to be used in various smart city applications, where they have been extensively used in applications such as collecting data tomaintain property security and enhance quality of life [22]. The structure of wired and wireless sensor networks are shown in Figure 2.2.

Figure 2.2: (a) Wired Sensor Network (b) Wireless Sensor Network

Sensor-based applications can provide real-time data to help city authorities with their op- erations [23]. Many platforms have been developed to manage sensors efficiently, as it is a vital element in

III. PRELIMINARIES

IN this chapter, the aim is to introduce and define the layers of a smart city, which compriseof the cloud layer, the fog/edge layer, the IoT layer, and the physical layer. Afterwards, the field of machine learning, alongside some of its relevant techniques and methods is introduced.

Lastly, the proposed Bayesian Network Learning approach used in this work is described.

A. Proposed Approach

The objective in this thesis is to manage the sensor network layer to bring down the overall computational load in a smart city. The goal is to understand the sensors behavior and to understand the surrounding environment, as well as to be able to efficiently enhance the perfor-mance of the smart city layers. To achieve this goal, a probabilistic, semi-supervised Bayesian learning approach was utilized.

This approach handles uncertainties effectively and improves the decision making process, aswell as reduces the need and cost of collecting labeled data. A preliminary definition of the Bayesian learning approach is given in the next subsection, and a more detailed discussion is found in Chapter 4.

B. Bayesian Network Learning

Bayesian networks were introduced by Judea Pearl in 1980s with the development of AI, en- abling probabilistic beliefs to be systematically and locally assembled into a single, coherent whole [71]. Bayesian provide a structured, graphical representation of probabilistic relation- ships between several random variables and an explicit representation of conditional indepen-dencies via directed acyclic graphs (DAGs). For example, a Bayesian network can represent the probabilistic relationships between various different features and targets, where the proba-bilities of the presence of these targets are computed given the features. In this work, Bayesiannetworks will learn to model the behavior of the sensors to be able to control the sensor's en- ergy consumption. Due to their mathematically grounded framework, Bayesian networks are the most popular method for uncertain expert knowledge and ratiocination, where it is vastly applied in large number of research areas [72].

Bayesian networks make use of Bayes Theorem during inference and prior to learning. BayesTheorem is an approach for calculating the conditional probability of an event, and is definedas:

$$\textit{Posterior} = \frac{\textit{Likelihood} * \textit{Prior}}{\textit{Evidence}}$$

IV. PHYSICAL LAYER MODEL

In this chapter, a mathematical model for the physical layer ℵ is presented. The physical layer model ℵ is divided into two sections: the physical environment ℵp, and the operational environment ℵo. The physical environment ℵp consists of the environment E, the sensors S, and the actions Λ, whereas the operational environment ℵo consists of the agents A, the reading history H, and the objectives O. The learning algorithms using NB classifiers will be introduced and explained as well. Moreover, a new modified Bayesian for accumulative learning algorithm will be explained, which is an enhanced version of the NB classifier.

As mentioned earlier, the physical layer ℵ is built out of the physical model ℵp, and the opera- tional model ℵo. Mathematically, it is described as the following tuple:

$$\aleph = (\aleph_p, \aleph_o) \tag{4.1}$$

A. Physical Model ℵp

Physical layer model ℵp is the mathematical model for the physical layer. Mathematically, ℵp is described as follows:

$$\aleph_p = (E, S, \Lambda) \tag{4.2}$$

Where,

- E is the environment.
- S is the set of sensors in $\aleph_p$.
- Λ is the set of actions that are applied to the environment.

The following sections explain each component that belongs to $\aleph_p$ in details:

An environment E is defined to be a set of variables, where each variable takes on values thatdescribe a certain natural phenomena, such as temperature, pressure or humidity. Sensors S provide those reading values to the set of variables over time. The number of variables depends on the number of sensors that exist in $\aleph_p$. Each variable represents a type of a measurement for a physical phenomenon. Mathematically, the environment is modelled as follows:

$$E = \{V_1, V_2, \ldots, V_n\} \tag{4.3}$$

Where,

- $n = \|E\|$
- $V = \{v_1, \ldots, v_k, v_i \in V\}$ is the set of variables associated with the sensor, and each V_i isassociated with one $s_j \in S$.

The set of sensors S is a vital component in $\aleph_p$ since it provides readings to describe the envi-ronment over time. This is done by providing environment variables with values as mentionedpreviously. A sensor is modelled as follows:

$$s_i \in S = (V, \Psi_i, \tau, \Phi(t)) \tag{4.4}$$

Where,

- V is the set of variables associated with the sensor as defined previously.
- $\Psi_i = \{\Psi_{i1}, \quad, \Psi_{ik}\}, \Psi_{ij}(v_j \in V)$ is a transfer function that maps a variable v_j of the sensor to a value r_t at time t; r_t is a numeric value that belongs to 9^+; Ψ_{ij} describes sensorbehavior; there is a transfer function for each variable.
- τ is the rate of readings. In other words, $\tau_i = \frac{1}{f_i}$, where f_i is the frequency of the sensor s_i.
- $\Phi(t) = (r_1, r_2, \ldots, r_i)$, is the sensor state which is the tuple of values assigned to $v \in V$ attime t and $r_i = \Psi(v_i)$, where i is the number of states for each variable.

As described, it is assumed that at any point of time, a sensor s_i can provide a reading for eachvariable associated with that sensor. Consequently, at time t, an environment state E_t can be defined as:

$$E(t) = (r_1, r_2, \ldots r_n) \tag{4.5}$$

Where $r_i = \Psi_i(v_i)$, and Ψ_i is the transfer function defined previously. We can think of $E(t)$ as an*instance* of the environment E at some point of time t. It is worth noting that if there is a sensorthat deals with four different variables, then it is going to have four transfer functions inside it.

As mentioned above, for every sensor $s_i \in S$, there exists a sensor state Φ at time t. A sensor state $\Phi(t)$ is the set of readings assigned to sensor variables at time t. Note that:

$$\Phi(t) \subseteq E(t) \tag{4.6}$$

At any point of time, environment state $E(t)$ is the joining of all sensor states at time t.

An action is any activity applied to ℵp to achieve an objective. As will be seen later in this chapter, objectives are classified as either global objectives which represent an overarching strategy for the physical model or local objectives for a sensor. It should be emphasized that objectives may change based on the changes that occur in the environment states. For example, if the temperature decreases, then the objective will change accordingly.

In this thesis, it is assumed that a finite set of basic actions is predefined; a basic action is an action that cannot be divided any further. An action changes the environment from one state E(t1) to another state E(t2). An action is applied to a sensor in order to change the value of a variable in order to change the state of the environment. The set of actions is finite. Actions are represented as follows:

$$\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_l\} \tag{4.7}$$

Where $l = \|\Lambda\|$.

Examples of actions would be:

- Changing the desired humidifier setting to increase or decrease humidity levels gradually.
- Changing the desired thermostat setting to increase or decrease temperature levels grad- ually.

B. Operational Model $\aleph_o$

The operational model $\aleph_o$ is the representation of how decision making processes for achievingobjectives for the physical environment are modeled, as well as how the reading history H is stored. The operational model $\aleph_o$ is mathematically modelled as follows:

$$\aleph_o = (H, O, A) \tag{4.8}$$

where,

- H is the reading history of sensors.
- O is the set of objectives that $\aleph_o$ needs to reach.
- A is the set of intelligent agents corresponding to each sensor.

The description of these elements will be illustrated with more details in the subsections below:

C. Reading History H

History H is the previous readings that occurred in the past. Theoretically speaking, ‖H‖ could be infinite, however, in reality and for the sake of performance, and due to the limited memory capacity of embedded environments, history length has to be controlled. For every sensor, there is a history. Since every sensor has one or more variables, history keepstrack of the sequence of readings per variable. History is mathematically represented as:

$$H = ((t_0, v_1, r(t_0)), \ldots, (t_0, v_k, r(t_0)), \ldots, (t_\tau, v_1, r(t_\tau)) \ldots (t_\tau, v_k, r(t_\tau))) \tag{4.9}$$

Equation 4.9 shows how the history is stored. It shows that the history is a sequence of tuples.Every tuple consists of the timestamp, the variable and the reading of that variable. Note that atany time, the timestamp can have a single tuple for a single variable, tuples for some variables,tuples for all variables, or no tuples at all. It is a sequence of tuples sorted according to time *t*.

D. Objectives O

The set of objectives O is being passed to the operational model $\aleph_o$ in order to be met. An operational model $\aleph_o$ might have one or more objectives to achieve.

In this thesis, objectives are classified into two types:

- Global Objectives O_G
- Local Objectives O_L

Mathematically, it can be described as:

$$O = < O_L, O_G > \tag{4.10}$$

Objectives are complex Boolean expressions that the operational system tries to make true. Theformation of the objective depends on whether it is a global or a local objective. The formation of a global objective would be something like $e \leq e_m$, where e is the total energy consumption and e_m is the maximum amount of energy consumption that the city cannot exceed. e is com- puted by calculating the total energy consumption in a house in kilowatt-hours (kWhs), takinginto consideration the number of rooms, and the number of occupants in each house. Global objectives are set on the cloud layer.

The local objectives are set on the edge layer. If there is a sensor with three variables, v_t for temperature, v_h for humidity, and v_il for illumination, then one objective of the sensor could besomething similar to:

$$t_l \leq v_t \leq t_h \text{ and } h_l \leq v_h \leq h_h \text{ and } il_l \leq v_{il} \leq il_h$$

where,

- v_t is the variable having temperature readings.
- t_l is the minimum temperature in the objective.
- t_h is the maximum temperature in the objective.
- v_h is the variable having humidity readings.
- h_l is the minimum humidity in the objective.
- h_h is the maximum humidity in the objective.
- v_{il} is the variable having illumination readings.
- il_l is the minimum illumination in the objective.
- il_h is the maximum illumination in the objective.

E. Intelligent Agent A

Modern and sophisticated applications introduce the need to take IoT a step further to become what is known to be the Internet of Intelligent Things (IoIT) [73]. IoIT adds intelligence to "things" in a smart city. This reduces the need for intensive communication with the fog layer or with the cloud layer, since much of the learning and decision making is handled in the IoT layer. In this thesis, for this to be achieved, it is assumed that for every sensor s S there is a corresponding agent α A, where s and α are associated; it is also assumed that a single agent handles all the variables that a sensor provides readings for. Agent α tries to meet an objective o OL by changing an environment state E(t) at time t.

Agents are executable software that learn and take decisions. They run on the IoT or edge layer on devices such as cellphones, computers, Raspberry Pis, and so on. They do not run on sensors but they read from them and control the actuators. The sensors will be connected to the edge devices on the edge layer where the agent runs. Agents read from the environment usingsensors and act on the environment using the actuators.

An intelligent agent $\alpha \in A$ is mathematically described as follows:

$$(\alpha \in A) = (\Xi, \Omega, Y, \Lambda_\alpha \subseteq \Lambda, s_\alpha \in S) \quad (4.11)$$

where,

- Ξ is the learning algorithm that agent α uses to learn about the behaviour of the associatedsensor $s \in S$.
- Ω is the prediction algorithm that predicts the future readings of each v in a sensor s.
- Y is the action selection process that selects the suitable actions for each v in a sensor s.The selection is based on the prediction $\wp \in \Omega$ and the objective $o \in O_L$.
- $\Lambda_\alpha \subseteq \Lambda$ is the set of actions that can be taken by agent $\alpha \in A$.
- $s_\alpha \in S$ is the sensor associated with the agent.

Every agent can have one or more objectives depending on the number of sensor variables theagent needs to change. The granularity of agents with respect to sensors in this thesis is one to one. In other words, every agent is responsible for a sensor that might have one or more sensorvariables to avoid having a single point of failure. A single point failure is the case when a centralized system is controlled by a single agent, where the whole system fails if the agent fails. It is worth noting that the global objective O_G is announced to all agents in order for them to start working locally and autonomously towards the achievement of that objective.

Figure 4.1 represents a data flow explaining how the mathematical model relate to the smart city computational layers discussed previously.

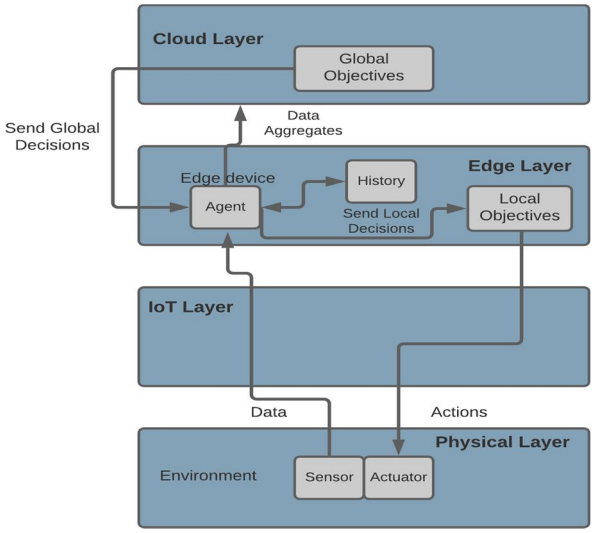


Figure 4.1: Data Flow

The following subsections describe the learning algorithm Ξ, the prediction process Ω, and the action selection process Y. The learning algorithm Ξ is achieved through an NB classifier. The learning process goes through a few steps as will be seen in the rest of the section. Next, the proposed modified Bayesian for accumulative learning is discussed, which is an enhancedversion of the NB classifier presented.

F. The Process of Learning

In this subsection, the motivation and discussion of the learning process is explained in detail.First, the problem is probabilistic since it depends on predicting and estimating future actionsthat can be taken. More specifically, the NB algorithm is considered since it is computationallyefficient and is able to achieve good results on a wide range of problems. The learning algo- rithm is general and can be used in many applications, however, it will be used in controlling the energy consumption in a city in this thesis.

The prediction helps us estimate future values of variables. This is important since we do not want to exceed a certain limit of any given objective. If we are able to predict future values, then situations can be handled better and objectives and limitations will not be exceeded. In this thesis, a new modified Bayesian for accumulative learning algorithm is presented, which is an improved version of the NB classifier. The algorithm is explained in Section 4.3.

1) *Statistical Summary of the Data:* The data used in our NB algorithm consists of sensor variables and their values. The data is initially separated into classes and the mean and the standard deviation of the data is calculatedto be able to get the probability. Algorithm 1 describes how the data set is summarized. It takesin a set of data and returns the mean μ and standard deviation σ per column. For descriptive purposes, the data is organized into columns as an input to the algorithm.

The algorithm works as follows:

Algorithm 1 : D Summary

```
Input : data set D.
Output : Stats data statistics.
Begin
  rCount — 0
  for all d ∈ D do
    cIndex — 0
4:  for all c ∈ d do
      μ[cIndex] — μ[cIndex] + c
      cIndex — cIndex + 1
    end for
8:  rCount — rCount + 1
  end for
  c — 0
  while c < cIndex do
12:  μ[c] — μ[c] / rCount
    c — c + 1
  end while
  for all d ∈ D do
16:  cIndex — 0
    for all c do
      σ[cIndex] — σ²[cIndex] + (d[cIndex] — μ[cIndex])²
      cIndex — cIndex + 1
20:  end for
    σ[cIndex] — sqrt(σ[cIndex] / rCount)
  end for
```

a) First, it iterates over each element of each column in the data set and gather all of thevalues for each column into a list.
b) The mean and the count of the list are then calculated.
c) Afterwards, the variance is calculated, and the standard deviation is outputted by takingthe square root of the variance.
d) The statistics are gathered into a list of tuples of statistics, and the operation is repeatedfor each column in the data set.
e) At the end, the summary of the data set is generated, where the mean, standard deviation,and count of each column are calculated, and a list of tuples of statistics is returned.

2) *Classification of Data D:* The classification of the data set is an essential component of the learning process. In Algorithm 1 the summary statistics for each column is calculated, where the statistics will be used after the classification.

3) *The Prediction Process:* The statistics calculated from our training data can now be used to calculate the probabilities of the new data. Calculating the probability or likelihood of a real value can be challenging. Hence, it can be achieved by assuming that the values are drawn from a Gaussian distribu- tion. This Gaussian distribution will have the mean and standard deviation values that were calculated in Algorithm 2.

Next, class predictions are performed on new data items by calculating the probability of each new data item belonging to each particular class, producing a list of probabilities. After an agent $\alpha \in A$ learns from history H, agent $\alpha \in A$ is ready for predicting the future behavior ofreadings. Algorithm 3 is the process of class prediction using probabilities calculation, and it processes as follows:
Target Readings Selection **Ω**
Target Readings Selection is the prediction algorithm that predicts the future readings of each *v* in a sensor *s*. Lets discuss the target readings selection process with an example. Assume we are trying to lower the energy consumption in a house by using a data that has a set of s and their v over time. It is also assumed that there are three sensor variables va, va, va in a
1 2 3
House with readings that lead to energy consumption that belongs to class label *a*. Moreover, the algorithm suggested to lower the target consumption to the class that belongs to label *b*. For example, let's say that the total consumption at class *a* is 1300 measure unit, and we wantto lower it to 1000 measure unit, which belongs to class *b*. Then, it is needed to figure out theoptimal reading from class *b* that will allow us to reach the required total consumption. Hence,the selected reading tuple in class *b* is the tuple that satisfies the following equation:

$$Min(\sqrt{(v_1^b[i] - v_1^a)^2 + (v_2^b[i] - v_2^a)^2 + (v_3^b[i] - v_3^a)^2}) \quad (4.12)$$

where $V_k[i]$ is reading number *i* for variable *k* in the sensor.
Algorithm 6 represents the getting new values process and it operates as follows:

a) The algorithm will take the target consumption and the current consumption as inputs.
b) It will then iterate over the classes, and select the class that is the closest to the target consumption.
c) Afterwards, it will go through that specific class, and the variable values that will provide us with the minimum Euclidean distance.
d) The selected variable values will provide us with the new tuple, allowing the target con-sumption to be reached.

G. The Action Selection Process Y

Action selection is a process that moves the sensor readings from one tuple to another to satisfya certain objective constraint. This happens through selecting the new class label that will try to make adjustments to move closer to the new consumption target. For every variable, there isan action, and the choice of actions will achieve the change of classes that will take place.

Algorithm 6: Get New Values
Input : Target Consumption (TC), current .
Output : $(v_1^a, \ldots, v_k^a)$, $(v_1^b, \ldots, v_k^b)$.

Begin
Max ← 0
Selectedc ← ∅
for all $c \in C$ **do**
4: **if** c.Consumption ≤ TC and c.Consumption > Max **then**
Max ← *c.Consumption*
Selectedc ← *c*
end if
8: **end for**
$v_1^a \ldots v_k^a$ ← *current.variables*
$min \leftarrow \infty$
for all $v_1^b \ldots v_k^b \in$ *selectedc.variables* **do**
12: $NT \leftarrow \sqrt{(v_1^b - v_1^a)^2 + (v_2^b - v_2^a)^2 + (v_3^b - v_3^a)^2}$
if $min \geq NT$ **then**
$min \leftarrow NT$
$(v_1^b, \ldots, v_k^b) \leftarrow (v_1^b, \ldots, v_k^b)$
16: **end if**
end for
return $(v_1^a, \ldots, v_k^a)$, $(v_1^b, \ldots, v_k^b)$
End.

Algorithm 7 represents the action selection process Y and it operates as follows:

1) The algorithm will take the newly chosen tuple from the Algorithm 6 as an input.
2) Afterwards, the algorithm will iterate through all actions for every variable and checks if the action will move the variable to the desired value.
3) Lastly, for every variable, the suitable action that will allow us to reach the target variablevalues will be selected.

Algorithm 7 : Action Selection

Input : $(v^s_1, \ldots, v^s_k), (v^b_1, \ldots, v^b_k), \Lambda$.
Output : Variable Actions (VA).

Begin
$VA - \emptyset$
for all $v_i \in (v_1, \ldots, v_k), v_i \in (v_1, \ldots, v_k)$ **do**
for all $\lambda \in \Lambda$ **do**
if $v^b_i - \delta(v^s_i, \lambda)$ **then**
$VA - VA \cup \lambda$
break
end if
end for
end for
return (VA)
End.

H. Modified Bayesian for Accumulative Learning

The proposed accumulative Bayesian learning algorithm updates the mean and the standard deviation of each class based on the new incoming data. The learning algorithm can be used in various applications and it will be used in controlling the energy consumption. The motivation behind the algorithm involves starting with historical data in the beginning, learning the mean and standard deviation of each class via supervised learning. Afterwards, the new data items are classified into different classes and the mean and standard deviation of all classes are updated. The algorithm is divided into two phases. In the first phase, the curve is enhanced by learningnew data items based on the training data with labels, followed by classifying the new data items. Afterwards, the second phase takes place, which occurs during the run time after the supervised learning training is completed. During that run time, the class of the generated data is unknown. The probability of each new data item belonging to each class is calculated,where the highest probability indicates the class that the data items belongs to. Subsequently, the mean and standard deviation get updated, and the curve gets smoother when more data items are added.

This section is dedicated to demonstrate the inner workings of the algorithm.

1) Updating the Mean

Since the mean is defined by:

$$\mu_n = \frac{\sum_{i=1}^{n} x_i}{n} \tag{4.13}$$

which also means that:

$$n\mu_n = \sum_{i=1}^{n} x_i \tag{4.14}$$

having a new value x_{n+1} makes the new μ as follows:

$$\mu_{n+1} = \frac{\sum_{i=1}^{n+1} x_i}{n+1} \tag{4.15}$$

therefore:

$$\mu_{n+1} = \frac{\sum_{i=1}^{n} x_i + x_{n+1}}{n+1} \tag{4.16}$$

From Equation 4.14 and Equation 4.15, it can be concluded that:

$$\mu_{n+1} = \frac{n\mu_n + x_{n+1}}{n+1} \tag{4.17}$$

$$\mu_{n+1} = \frac{n\mu_n + x_{n+1}}{n+1} \tag{4.18}$$

2) Updating the Standard Deviation

Standard deviation of n elements is defined by:

$$\sigma_n = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu_n)^2}{n}} \tag{4.19}$$

Standard deviation of $n + 1$ elements is defined by:

$$\sigma_{n+1} = \sqrt{\frac{\sum_{i=1}^{n+1}(x_i - \mu_{n+1})^2}{n+1}} \tag{4.20}$$

which makes the variance of $n + 1$ elements as:

$$\sigma^2_{n+1} = \frac{\sum^{n+1}(x_i - \mu_{n+1})^2}{i=1 \quad n+1} \tag{4.21}$$

This leads to:

$$\sigma^2_{n+1} = \frac{\sum_{i=1}^{n}(x_i - \mu_{n+1})^2 + (x_{n+1} - \mu_{n+1})^2}{n+1} \tag{4.22}$$

which makes the variance of $n + 1$ elements as:

$$\sigma^2_{n+1} = \frac{\Sigma^{n+1}(x_i - \mu_{n+1})^2}{n+1} \quad (4.21)$$

This leads to:

$$\sigma^2_{n+1} = \frac{\Sigma^{n}_{i=1}(x_i - \mu_{n+1})^2 + (x_{n+1} - \mu_{n+1})^2}{n+1} \quad (4.22)$$

$$\sigma^2_{n+1} = \frac{\Sigma^{n}_{i=1}(x_i^2 - 2x_i\mu_{n+1} + \mu^2_{n+1}) + (x_{n+1} - \mu_{n+1})^2}{n+1} \quad (4.23)$$

$$\sigma^2_{n+1} = \frac{\Sigma^{n}_{i=1}(x_i^2 - 2x_i\mu_{n+1} + \mu^2_{n+1}) + x^2_{n+1} - 2x_{n+1}\mu_{n+1} + \mu^2_{n+1}}{n+1} \quad (4.24)$$

$$\sigma^2_{n+1} = \frac{\Sigma^{n}_{i=1} x_i^2 - \Sigma^{n}_{i=1} 2x_i\mu_{n+1} + \Sigma^{n}_{i=1,n+1} \mu^2_{n+1} + x^2_{n+1} - 2x_{n+1}\mu_{n+1} + \mu^2_{n+1}}{n+1} \quad (4.25)$$

$$\Sigma^{n}_{i=1} 2x_i\mu_{n+1} = 2\mu_{n+1}\Sigma^{n}_{i=1} x_i = 2n\mu_n\mu_{n+1} \quad (4.26)$$

This gives the following formula:

$$\sigma^2_{n+1} = \frac{\Sigma^{n}_{i=1} x_i^2 - 2n\mu_n\mu_{n+1} + (n+1)\mu^2_{n+1} + x^2_{n+1} - 2x_{n+1}\mu_{n+1}}{n+1} \quad (4.28)$$

which leads to:

$$\sigma^2_{n+1} = \frac{\Sigma^{n+1}_{i=1} x_i^2 - 2n\mu_n\mu_{n+1} + (n+1)\mu^2_{n+1} - 2x_{n+1}\mu_{n+1}}{n+1} \quad (4.29)$$

So the standard accumulative formula is:

$$\sigma_{n+1} = \sqrt{\frac{\sum_{i=1}^{n+1} x_i^2 - 2n\mu_n\mu_{n+1} + (n+1)\mu_{n+1}^2 - 2x_{n+1}\mu_{n+1}}{n+1}} \quad (4.30)$$

I. Classification Algorithm

Unlike the original Bayesian's classifier, which is a supervised learning-based algorithm, the new proposed classifier is a semi-supervised learning-based approach. It starts as supervised learning at the beginning, then later, it keeps learning without the need for any supervision. Without the initial supervised learning step, classes could be separate for each row, since for every data element the mean is the value of the element and the standard deviation is zero. The classification is done in two phases, phase one is supervised when the class label is provided along with input data, and phase two is unsupervised when no label is provided.

Algorithm 8 describes phase one, and it operates as follows:

1) Initially, the algorithm takes the data and the labels as inputs.
2) Next, it iterates over each item in the data set, and check if the label of the data itembelong to a certain class.
3) If the label does not belong to any class, then a class is created for that label.
4) After that, loop through every class item, and for every class column in the classes, themean and standard deviation of that column is calculated.
5) At the end, all classes are returned as the output of the algorithm.

Phase 2 is the learning process during run time. This is done in two steps; the first step is the classification mechanism, and the second one is the distribution update. The classification is done based on the highest probability provided by a particular class.

Algorithm 9 describes phase two, and it operates as follows:

a) In this algorithm, the data and classes are taken as the inputs.
b) For every data item in the set, the probability for each class is calculated.
c) Whenever a new data item is added, check to see to which class that the data belongs toby calculating the probability.
d) The class with the highest probability are chosen.
e) After that, the mean and the standard deviation are updated by using Equations 4.18 and4.30.

To explain classes in a better way, an example showing temperature classes will be discussed.First, the data is classified based on predefined labels. The classes are as follows:

- Temperatures from -20 ˚C and below belongs to class 1.
- Temperatures between -19 ˚C to -10˚C belongs to class 2.
- Temperatures between -9 ˚C to 0˚C belongs to class 3.
- Temperatures between 1 ˚C to 10˚C belongs to class 4.
- Temperatures between 11 ˚C to 20˚C belongs to class 5.
- Temperatures between 21 ˚C to 30˚C belongs to class 6.
- Temperatures from 30 ˚C and above belongs to class 7.

V. SIMULATION AND EXPERIMENTS

ENergy consumption management is one of the major challenges faced in a smart city, dueto the lack of resources and the rapid growth of the world's population. In particular, governments have been trying to tackle the energy over-consumption issues by implementing different policies such as "the more you consume, the more you pay". While such policies help reduce energy consumption, it does not guarantee that the consumption does not exceed a certain target thresholds. It is becoming ever so important for smart cities to try to set maximumthresholds to avoid blackouts that result from over-consumption.

For example, countries in the Gulf Cooperation Council (GCC) area, where the temperature reaches above 50 degree Celsius, have high possibilities of blackouts and electricity grid cut offs, particularly in the summer when people over-consume energy due to the overuse of air- conditioners and dehumidifiers.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue XII Dec 2021- Available at www.ijraset.com

One potential solution includes setting energy consumption limit thresholds for each house across the smart city, in order to try to manage the energy consumption and ensure the electrical grid can keep providing what is necessary for the city tofunction.

Implementing such a policy or strategy raises many issues in governance, privacy, individual rights, etc.. This is used as a general scenario to illustrate our model and to illustrate the approach and algorithms. Such a scenario is not entirely hypothetical since a city would have control of the buildings it manages and would be able to introduce operational policies to reduce or manage energy consumption in its own buildings. For our experiments, they are described in terms of "houses".

One question that does arise is the maximum threshold boundary that needs to be set for each house or building. This is a probabilistic classification problem since every house might fall into different categories. It is worth noting that the consumption of every house is independentof the consumption of other houses and so there is no "global" control. Since the problem is probabilistic and houses are independent, NB classifier is a an excellent candidate to be used as a solution since it assumes that variables are independent. Bayesian classification is simple and fast which makes it very suitable for real-time systems, which is the case in energy grid control systems.

Note that classification is vital since it provides important information for categorizing houses based on their consumption for decision-making purposes. The algorithms learn which settings of devices, such as air conditioners, humidifiers, heaters, and dehumidifiers, lead to a particularconsumption class. Then, when it is required to move a certain house from one consumption level to another, the settings that are highly likely to lead to a certain consumption are chosen. Note that classes could have different lengths. The class is defined through supervised learningprocess when feeding the system with a certain input and the label associated to it. The input isbasically the current readings and the target readings. Values that have the same consumptionrange get the same label. That range is determined by the designer or administrator. In this thesis, it is assumed that the reading range takes place every 10 measurements; a single readingcould belong to multiple classes with different probabilities. Hence, the class that gives the highest probability is the class of this reading. If the following row of values are given:

1) Current temperature (T_c)
2) Target temperature (T_t)
3) Current humidity (H_c)
4) Target humidity (H_t)
5) Consumption (c)

Then this row belong to class C if the following is the maximum among all different classes, where P(x|y) is the probability of x given y:

$$P(T_c|C) \times P(T_t|C) \times P(H_c|C) \times P(H_t|C) \times P(c|C)$$

It is worth noting that labels are just unique class names that are assigned as identifiers for every class.

A. Experiments

In this section, the experiments are presented, where the focus will be driven towards the prob-lem of addressing the minimizing of energy across a city. Several experiments with different scenarios are presented along with their results. For all experiments, it is assumed that there is a city that has 100-1000 houses, where every house has a random number of rooms that canrange from 3 rooms, and up to 15 rooms. In every house there will be at least one sensor that has three variables, which are the temperature, humidity and motion sensors. Every house hasits own consumption based on the amount of energy used per house. The total city consump- tion is calculated per hour, where the whole city has a global objective. Moreover, the idea is to reduce the overall consumption only when it is more than the assumed maximum capacity.

The Total Consumption class in the simulator will always calculate the sum of the consumption of all the houses in the city per hour. Based on that, it will choose the houses that exceeded the the consumption rate and violated the local objective, and try to reduce their consumption to meet the global objective. This will be done by the agent which tries to predict on the targetreadings selection, and select the suitable actions to lower the consumption. Also, by taking into consideration the number of rooms in the houses, and the number of individuals who live in the house. Since, as the number of rooms and individuals in a certain house increases, this will lead to the increase of the consumption as more energy will be utilized.

The number of individuals in each house will be generated randomly. It is expected that each house will have 0 occupants and up to 6 occupants on average. Based on the number of roomsand individuals in each house, the objective will be set. The house shouldn't exceed that limitto avoid exceeding the global objective. When a decision is taken, the sensors start reading soif people change rooms or leave their houses, it will be recognized in the next run.

B. Data

From Figure 5.3, it can be seen that the temperature and humidity sensor variables' values are plotted since they affect the total consumption of the city. Hence, based on the temperature, the heating or air conditioner will be turned on or off, and based on the humidity, the humidifier or dehumidifier will be turned on or off. Accordingly, these electrical devices will affect the kilowatts-hour (kWhs) that we are trying to compute. The lighting of the rooms will be con- trolled by checking whether individuals are inside or outside the room, and depending on the time of the day.

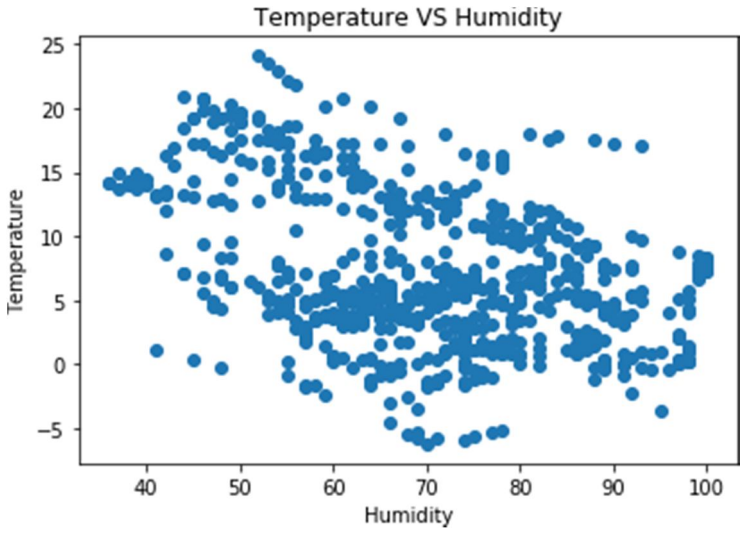
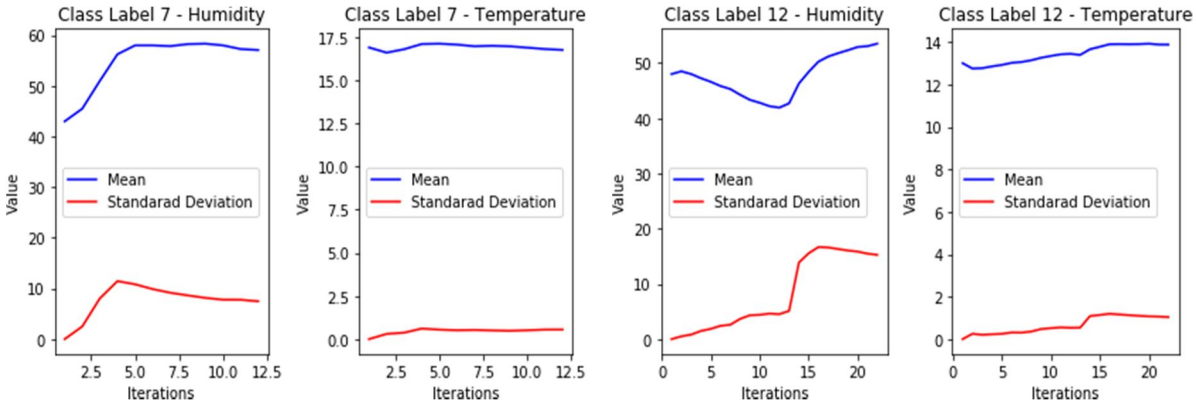


Figure 5.3: Temperature VS Humidity

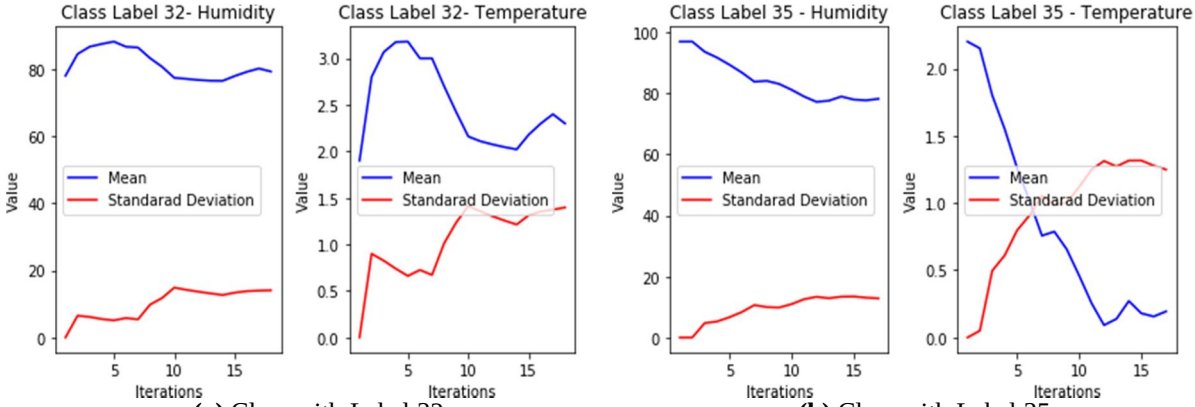
So, the temperature and humidity are controlled in these experiments and the variation of values of these variables is dependent on the external temperature and that is why the data on external temperature and humidity was extracted from The Weather Network [74]. Figure 5.3 is the pre-processed data that is used for temperature and humidity, and moving forward this data will be classified into different classes.

After classifying the data, the labels and classes are summarized in Tables 5.1 and 5.2. The tables include the class label, the name of the variable, the mean, the standard deviation, and the length of each class. Figures 5.4 and 5.5 illustrates how the classes learn the mean and the standard deviation. The learning take place whenever a new data item is added to a class. As can be seen, the curves illustrate that learning occurs while data is collected and training takesplace. The training data is limited and accordingly the learning can certainly be improved, if the training data was larger. In the second phase during the run time, the learning will move from supervised learning to unsupervised learning.



(a) Class with Label 7 **(b)** Class with Label 12

Figure 5.4: Classes with Labels 7 and 12



(a) Class with Label 32 **(b)** Class with Label 35

Figure 5.5: Classes with Labels 32 and 35

In the following subsections, the experiments and their results are explained. Each experiment constraints will be demonstrated and the results will be discussed. Every experiment is repeated and goes through 30 iterations, and the output is illustrated and summarized using graphs.

Table 5.1: Classes Statistics for Humidity and Temperature - Part 1

Label	Name	Mean	STD	Length
1	Humidity	48.85	5.30	7
1	Temperature	20.53	0.38	7
2	Humidity	51.10	4.06	10
2	Temperature	20.11	1.00	10
3	Humidity	50.56	5.54	9
3	Temperature	19.64	1.23	9
4	Humidity	52.0	3.60	8
4	Temperature	19.79	2.35	8
5	Humidity	54.83	5.58	6
5	Temperature	17.87	0.60	6
6	Humidity	51.56	4.52	9
6	Temperature	17.20	0.24	9
7	Humidity	57.08	7.47	12
7	Temperature	16.76	0.55	12
8	Humidity	55.63	7.24	8
8	Temperature	16.16	0.51	8
9	Humidity	62.33	15.40	6
9	Temperature	16.22	1.31	6
10	Humidity	61.43	14.59	14
10	Temperature	15.56	1.00	14
11	Humidity	56.94	16.18	17
11	Temperature	14.75	1.00	17
12	Humidity	53.50	15.26	22
12	Temperature	13.88	1.04	22
13	Humidity	53.83	9.63	6
13	Temperature	13.10	0.16	6
14	Humidity	66.29	7.67	14

14	Temperature	13.07	0.53	14
15	Humidity	68.25	3.19	8
15	Temperature	12.46	0.36	8
16	Humidity	68.90	5.52	10
16	Temperature	11.94	0.56	10
17	Humidity	70.50	8.51	10
17	Temperature	11.36	0.77	10
18	Humidity	73.00	12.13	7
18	Temperature	11.12	0.94	7
19	Humidity	69.13	13.47	15
19	Temperature	10.11	1.14	15
20	Humidity	73.64	10.90	14
20	Temperature	9.65	1.10	14
21	Humidity	65.91	14.30	22
21	Temperature	8.40	1.29	22
22	Humidity	66.80	14.38	15
22	Temperature	7.68	1.38	15

Table 5.2: Classes Statistics for Humidity and Temperature - Part 2

Label	Name	Mean	STD	Length
23	Humidity	71.00	13.07	17
23	Temperature	7.45	1.30	17
24	Humidity	67.19	10.21	26
24	Temperature	20.11	1.00	26
25	Humidity	67.90	10.846	48
25	Temperature	5.83	1.07	48
26	Humidity	78.46	12.39	35
26	Temperature	6.23	1.21	35
27	Humidity	72.25	14.50	40
27	Temperature	4.98	1.45	40
28	Humidity	75.36	13.63	39
28	Temperature	4.57	1.38	39
29	Humidity	77.05	10.23	21
29	Temperature	4.03	1.01	21
30	Humidity	80.89	11.46	17
30	Temperature	3.75	1.16	17
31	Humidity	71.53	14.80	19
31	Temperature	2.33	1.35	19
32	Humidity	79.28	14.07	18
32	Temperature	2.30	1.40	18
33	Humidity	74.30	7.70	27
33	Temperature	1.14	0.83	27
34	Humidity	77.77	9.11	22
34	Temperature	0.73	0.93	22
35	Humidity	78.24	12.83	17
35	Temperature	0.19	1.25	17
36	Humidity	79.05	11.11	20
36	Temperature	-0.45	1.08	20

37	Humidity	84.86	9.77	14
37	Temperature	-0.48	1.01	14
38	Humidity	94.00	7.76	16
38	Temperature	-0.20	0.83	16
39	Humidity	82.50	13.50	2
39	Temperature	-1.90	1.50	2
40	Humidity	66.00	0.00	1
40	Temperature	-4.50	0.00	1
41	Humidity	92.00	0.00	1
41	Temperature	-2.20	0.00	1
42	Humidity	68.67	0.47	3
42	Temperature	-5.50	0.16	3
43	Humidity	77.14	7.77	7
43	Temperature	-5.37	0.79	7

1) *Experiment 1:* The first experiment consists of 100 houses, where there is a random number of rooms and a random number of people in each house. The following are the assumptions for the experiment:

a) There is one sensor in each house and the sensor has one or more variables.

b) Each house has its own local objective based on the number of people and the relative energy consumption with respect to the mean, and the target consumption. Figure 5.6illustrates houses with different targets (local objectives) that they are trying to achieve.

```
house 1 has high consumption. It is consuming 4485.0 target is 4036.5 target class 40
house 3 has high consumption. It is consuming 3945.0 target is 3550.5 target class 35
house 4 has high consumption. It is consuming 4815.0 target is 4333.5 target class 43
house 5 has high consumption. It is consuming 4350.0 target is 3915.0 target class 39
house 9 has high consumption. It is consuming 3900.0 target is 3510.0 target class 35
house 15 has high consumption. It is consuming 3705.0 target is 3334.5 target class 33
house 17 has high consumption. It is consuming 4620.0 target is 4158.0 target class 41
house 21 has high consumption. It is consuming 3480.0 target is 3132.0 target class 31
house 23 has high consumption. It is consuming 3465.0 target is 3118.5 target class 31
house 24 has high consumption. It is consuming 2925.0 target is 2632.5 target class 26
house 26 has high consumption. It is consuming 3840.0 target is 3456.0 target class 34
house 27 has high consumption. It is consuming 4965.0 target is 4468.5 target class 44
house 30 has high consumption. It is consuming 3015.0 target is 2713.5 target class 27
```

Figure 5.6: Local Objectives of Houses

c) The global objective the city is trying to achieve is to not exceed a threshold limit of 90% of the original consumption which is 155096 kilowatts. This will reduce the energyconsumption by at least 10%.

d) Global objective is changed to have a threshold limit of 85% of the original consumptionwhen the experiment started. This will reduce the energy consumption by at least 15%.

e) Global objective is changed to have a threshold limit of 75% of the original consumptionwhen the experiment started. This will reduce the energy consumption by at least 25%.

- *Results for Experiment 1:* The results for the experiment with the different global objectives are illustrated in Figures 5.7,5.8, and 5.9. As can be seen, all the consumption are below the threshold, which validates our experiment. In Figure 5.9 the energy consumption percentage was very close to the threshold but it did not exceed it; it remains below 1% of the threshold and did not exceed it.

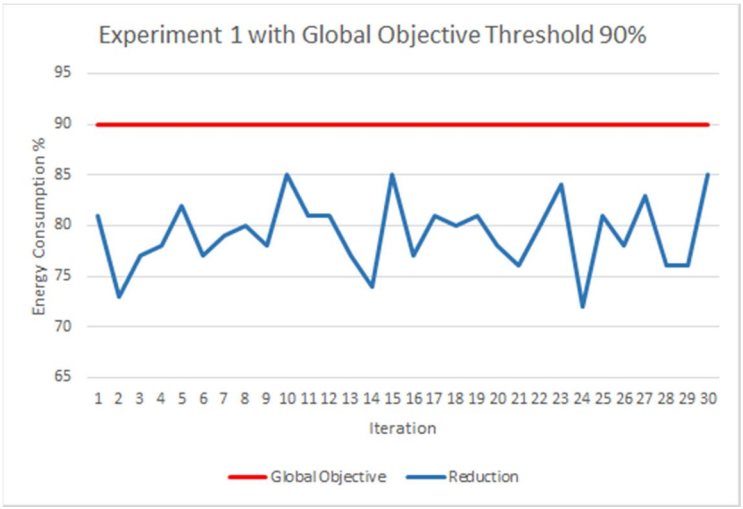


Figure 5.7: Experiment 1 - 90% Global Objective

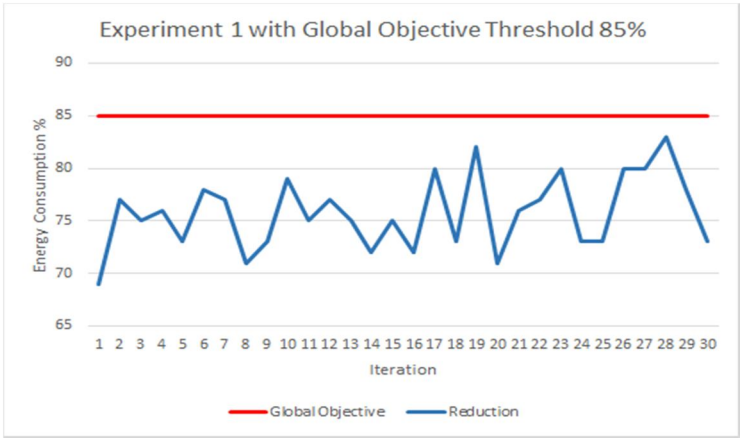


Figure 5.8: Experiment 1 - 85% Global Objective

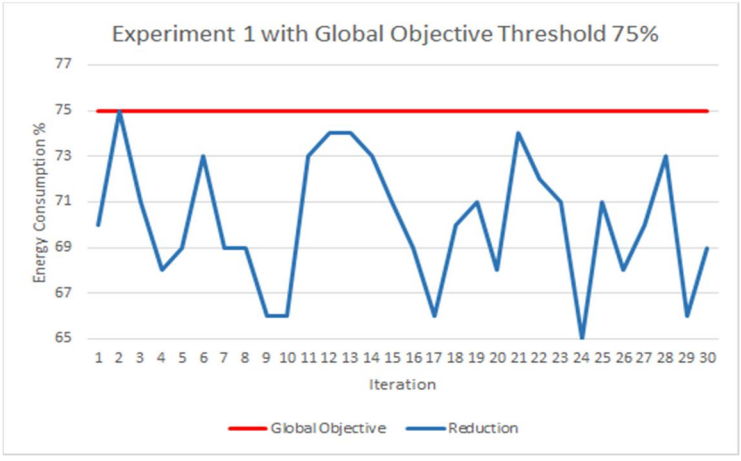


Figure 5.9: Experiment 1 - 75% Global Objective

2) *Experiment 2:* The second experiment is similar to the first experiment but uses 500 houses with randomnumbers of rooms and people instead. The following are the assumptions for the experiment:

a) There is one sensor in each house and the sensor has one or more variables.

b) Each house has its own local objective based on the number of people and the relativeenergy consumption with respect to the mean, and the desired target consumption.

c) The global objective the city is trying to achieve is to not exceed a threshold limit of 90% of the original consumption which is 772407 kilowatts. This will reduce the energyconsumption by at least 10%.

d) Global objective is changed to have a threshold limit of 85% of the original consumptionwhen the experiment started. This will reduce the energy consumption by at least 15%.

e) Global objective is changed to have a threshold limit of 75% of the original consumptionwhen the experiment started. This will reduce the energy consumption by at least 25%.

- *Results for Experiment 2:* The results for the experiment with different global objectives are illustrated in Figures 5.10, 5.11, and 5.12. As can be seen, all the consumption levels are below the threshold, which validates this experiment as well. This indicates that the model is able to control a city with anincreased number of houses.

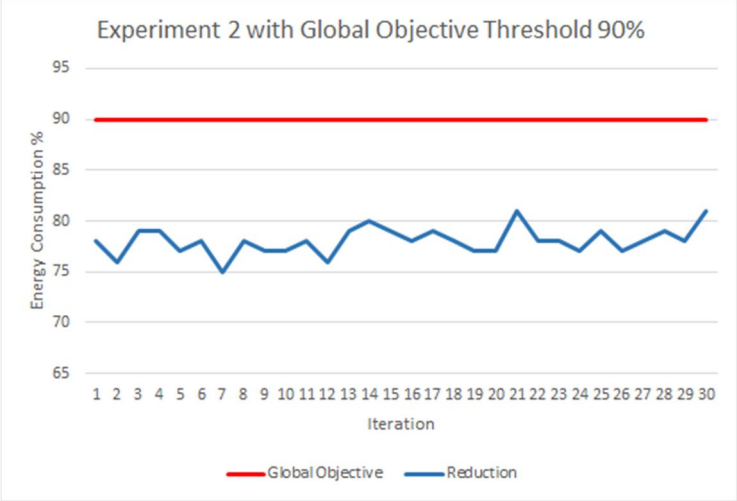


Figure 5.10: Experiment 2 - 90% Global Objective

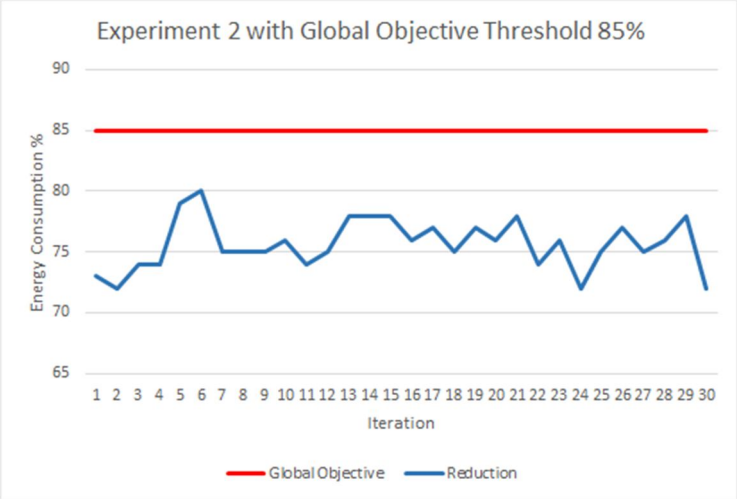


Figure 5.11: Experiment 2 - 85% Global Objective

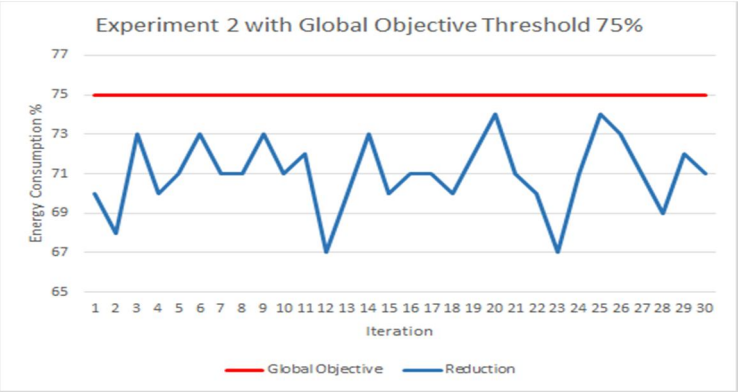


Figure 5.12: Experiment 2 - 75% Global Objective

VI. CONCLUSION AND FUTURE WORK

N conclusion, the proposed model and algorithms were evaluated by simulating a smartcity with different scenarios with the aim to reduce the energy consumption to meet a global objective. The results demonstrated the proposed model and algorithms effectivenessin managing and lowering the energy consumption in a city by around 34%. Six experimentssimulated diverse situations where multiple houses with different settings and environmentswere considered. In each experiment, a global objective that limited the energy consumption in a city was imposed and the energy consumption levels were decreased.

A. Summary of Contributions

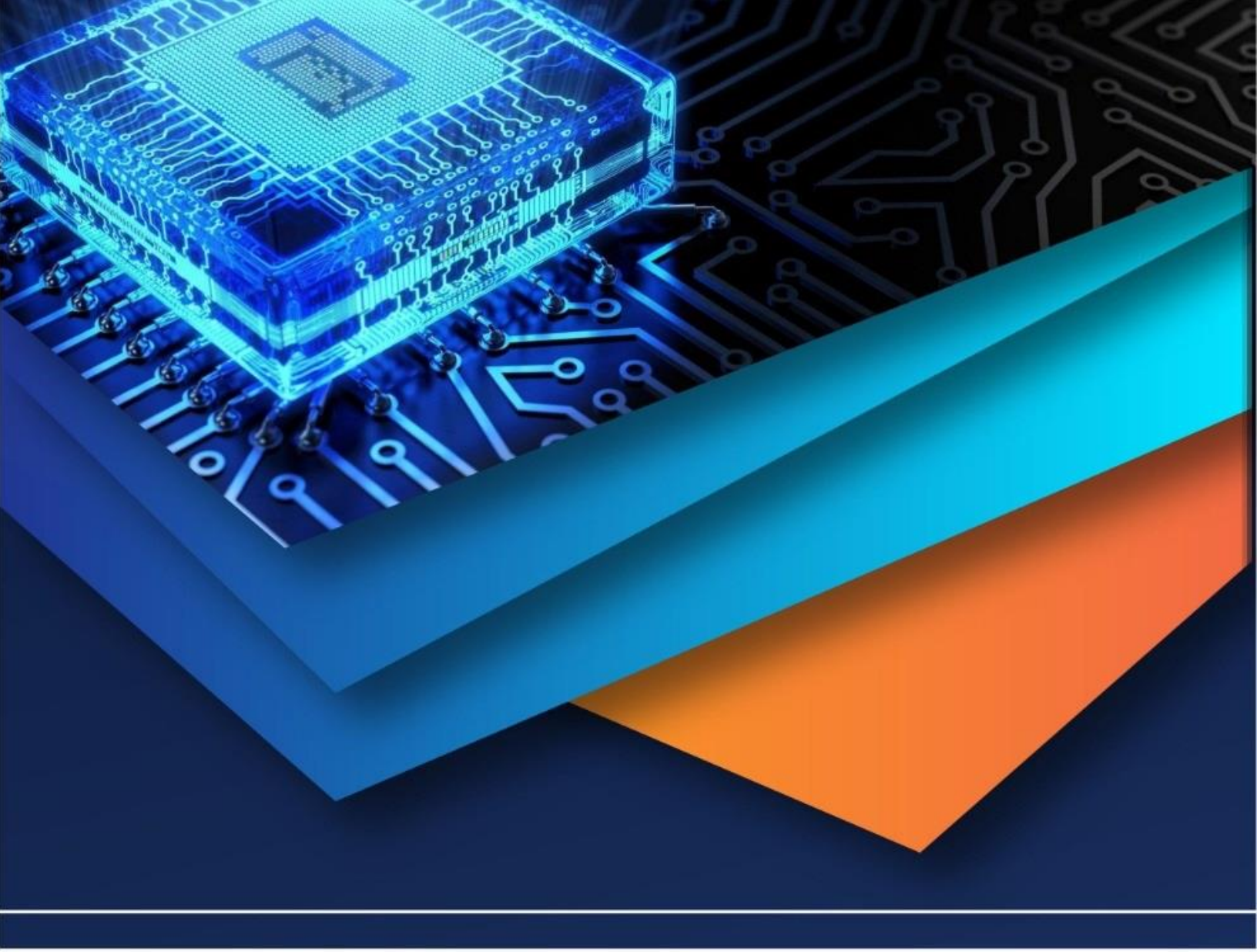
In this thesis, a model for a smart city physical layer is proposed. The model provided a math-ematical model for the framework components of the basic physical infrastructure in a smart city and their relationships with each other. The main components taken into consideration en-compass the physical environment and the operational environment. The physical environmentmodel includes the environment E, the sensors S, and the actions Λ, whereas, the operational environment model includes the intelligent agents A, the reading history H, and the objectives O. A novel semi-supervised Bayesian machine learning algorithm is proposed. The novel al- gorithm learns the behavior of a sensor to predict the future actions in order to reach a local orglobal objective. A simulator that simulates the physical layer model and the proposed learning algorithm was programmed and employed to test the model and algorithm. The main contributions in this thesis are the proposed physical layer framework and the en- hanced Bayesian machine learning algorithm, where both can be utilized to leverage smart cityservices and applications.

B. Future Work

There are some limitations in the proposed work and can be addressed in further extensions. Firstly, the proposed physical model can be utilized to target other objectives in a smart city, such as traffic management and smart streetlight systems. Therefore, a research extension canbe conducted to target various objectives besides energy consumption management. Experi- mentation with larger data sets can be done to explore whether the novel learning algorithm can learn more interesting features or produce better classification and prediction results. Ad-ditionally, policies can be identified for different types of managed objects in a smart city, e.g.,policies dealing with configuration management and adaptation could be examined, as sensorsmay fail or stop working at any given time, affecting the whole system or subsystem.

BIBLIOGRAPHY

[1] United Nations. 68% of the world population projected to live in urban areas by 2050, 2018. https://www.un.org/development/desa/en/news/population/ 2018-revision-of-world-urbanization-prospects.html.
[2] Grand View Research. Smart cities market size, share & trends analysis re- port, 2020. https://www.grandviewresearch.com/industry-analysis/smart- cities-market.
[3] Irene Celino and Spyros Kotoulas. Smart cities [guest editors' introduction]. IEEE Inter- net Computing, 17(6):8–11, 2013.
[4] Renata Paola Dameri. Searching for smart city definition: a comprehensive proposal. international Journal of computers & technology, 11(5):2544–2551, 2013.
[5] Giuseppe Anastasi, Michela Antonelli, Alessio Bechini, Simone Brienza, Eleonora D'Andrea, Domenico De Guglielmo, Pietro Ducange, Beatrice Lazzerini, Francesco Mar- celloni, and Armando Segatori. Urban and social sensing for sustainable mobility in smart cities. In 2013 Sustainable Internet and ICT for Sustainability (SustainIT), pages 1–4. IEEE, 2013.
[6] Jos e´ Antonio Galache, Takuro Yonezawa, Levent Gurgen, Daniele Pavia, Marco Grella, and Hiroyuki Maeomichi. Clout: Leveraging cloud computing techniques for improving



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY