Lappeenranta University of Technology
School of Engineering Science
Erasmus Mundus Master's Programme in Pervasive Computing & Communications for Sustainable Development PERCCOM

Carl Makafui Kugblenu

PREDICTIVE MODELLING FOR IOT ENABLED WASTE MANAGEMENT SYSTEM

Supervisors: Professor Arkady Zaslavsky (CSIRO)
Dr. Sylvain Kubler (Université de Lorraine)

Examiners : Professor Eric Rondeau (Université de Lorraine)
Professor Jari Porras (Lappeenranta University of Technology)
Associate Professor Karl Anderson (Luleå University of Technology)

This thesis is prepared as part of an European Erasmus Mundus programme PERCCOM – Pervasive Computing & COMmunications for Sustainable Development.

This thesis has been accepted by partner institutions of the consortium (cf. UDL-DAJ, n°1524, 2012 PERCCOM agreement).

Successful defense of this thesis is obligatory for graduation with the following national diplomas:

- Master in Complex Systems Engineering (University of Lorraine)
- Master of Science in Technology (Lappeenranta University of Technology)
- Degree of Master of Science (120 credits) –Major: Computer Science and Engineering, Specialisation: Pervasive Computing and Communications for Sustainable Development (Luleå University of Technology)

Lappeenranta University of Technology
School of Engineering Science
PERCCOM Master Program

Carl Makafui Kugblenu

Predictive Modelling for IoT Enabled Waste Management System

Master's Thesis 2018

Examiners: Professor Eric Rondeau (Université de Lorraine*)*
Professor Jari Porras (Lappeenranta University of Technology)
Associate Professor Karl Anderson (Luleå University of Technology)

Keywords: municipal solid waste, classification, machine learning, modelling, decision trees

ABSTRACT

The research presents a web based Decision Support System(DSS) that operates within the waste management and IoT ecosystem. The DSS leverages temperature, humidity, fill-level and weight about the Smart Garbage Bins(SGB) to improve waste collection and reduce the number of messages being sent by the SGBs by predicting the send intervals based on frequency of the same data. We demonstrate the efficiency of our approach with a decision tree and compare with four machine learning predictive classifiers: i) Random Forest ii) Naïve Bayes iii)K Nearest Neighbour iv) Support Vector Classifier. Results demonstrate that decision perform best based on accuracy, precision and recall. It also improves the waste collection process and can reduce the number of messages sent by the SGB by 44% to 45% of the original message count.

ACKNOWLEDGEMENTS

The research reported here was supported and funded by the PERCCOM Erasmus Mundus Program of the European Union (PERCCOM- FPA 2013-0231). The authors would like to express their gratitude to all the partner institutions, sponsors, and researchers involved in the PERCCOM program [1].

I would like to take this opportunity to thank to the PERCCOM Selection committee, the host universities, such as University of Lorraine, Lappeenranta University of Technology, ITMO University, Luleå University of Technology and Leeds Beckett University, and especially Professor Eric Rondeau for the efforts that have been invested in PERCCOM.

I would like to express my deep gratitude to my supervisors Professor Arkady Zaslavsky and Dr. Sylvain Kubler for all kinds of support, meaningful feedbacks and encouragement throughout this master thesis.

I would like to also thank my family for their encouragement and support throughout the period of my thesis. The people who have helped and supported me during the past two years, PERCCOM family and consortium, the European Commission for financial support.

Above all, I give thanks to The Almighty, I would not have made it this far if not by His grace and mercy

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

DSS	Decision Support System
SGB	Smart Garbage Bin
FN	False Negative
FP	False Positive
IoT	Internet of Things
LoRaWAN	Long Range Wide Area Network
ML	Machine Learning
MSW	Municipal Waste Management
TN	True Negative
TP	True Positive

1 INTRODUCTION

According to the United Nations [2], the world population will increase to about 8.6 billion people in 2030 and 9.8 billion people in 2050, this implies an increase in production, consumption and waste generation. Currently, the amount of waste generated per day globally is 3.3 million tons; a growing concern for many major cities [3].

The traditional model of waste management is not resource efficient (personnel and vehicles) and time-consuming which has brought about the need to develop intelligent systems that can efficiently and effectively alert authorities on the status and threshold levels of the smart bins, that is, their weight, fill level, odour and temperature. Many developed cities have sensors embedded into the bins to provide important information on the status of the bin, their contents. However, the data gathered is not effectively utilized to manage collection. Internet of Things enabled smart bins provide real-time information on the content as well as how disturbing the content could be to the citizens, e.g., stench because of rotten onion and weight of waste which could impact their well-being and quality of life. The forecasting and gathering of information on the waste generated provides data that can be used to anticipate future scaling and optimization of services to accommodate new demands [4].

Sustainable waste management does not just involve collection of waste, it also focuses on the efficiency of waste transportation from their respective locations, incorporates feedback loops and ensures responsible disposal. This is to reduce harm to the environment and the impact on the quality of life for citizens [5]. The rise in commodity consumption creates a challenge where users have more items that they rarely use which is not viable. Appropriate waste management also promotes waste prevention or minimization, recycling and composting. In many countries, a bigger proportion of waste is hardly reintroduced into reuse or recycling thus ends up in landfills or is incineration [6].

Predictive Modelling of waste management provides more accuracy than traditional techniques especially in processing of large amounts of data. The city of Calgary in Canada Municipal Waste Management [7] employed machine learning techniques on 63,000 records on waste which outperformed traditional practices. They applied hybrid prediction combination of weighted k-means for clustering and linear regression for the final predictors.

The computational performance on the 63,000 records took 20 minutes on a PC with 4GB of memory. The quality and validity of results enabled the managers to make informed decisions on optimization of costs and operational services for guaranteed quality service provision to their customers (households) who rely on Waste and Recycling Services (WRS) such as residential waste, recycling and organic collection. The results were also utilized for future prediction and planning purposes in the city [8].

With the integration of IoT in cities, traditional approach in service provision can be altered into smart services and other new services. This includes services in waste management, environmental monitoring and optimizing energy consumption, healthcare and other services [9].

Forecasting historical data on waste management using machine learning reduces the challenges in waste management and provides opportunities to better manage the prioritization of waste collection according to features such as fill-level, odour, and weight. Despite the much-vaunted merits of machine learning, the effectiveness of its algorithms still heavily rely on what data is fed into them. An abundance of relevant data can impressively fuel a machine learning algorithm much like useful clues can help a detective to reach wiser conclusions. It is exactly for this reason that IoT can make an ideal use case for the technology. A wide range of IoT devices can highly frequently generate data which can then be placed into the algorithm. Data from sensors in the smart bins provide vital information which influences decisions on matters such as planning for collection thus minimizing the amount of fuel consumed by the trucks, reducing emissions of harmful gases from the bins and from the trucks during collection and reducing the overall operational costs.

1.1 Problem Definition

According to the Global Waste Management Outlook report [10], waste is a global issue that poses a threat to public health and the environment if not properly dealt with. Due to rising quality of life and high resource consumption, waste generation levels have reached an unprecedented state that is beyond the capacities of urban government and agencies. Cities are therefore faced with the problems of high volumes of waste which include higher costs and adverse effects of waste both on the local and global environment. There is a need for the traditional approach of waste disposal - that is focused on municipalities and their inefficient use of energy and technologies; to move towards efficient use of technologies for waste processing, recycling and management [11]. As shown in figure 1, Waste generation rates are highly dependent on the income level, socio-cultural patterns and the climate conditions. Organic waste has the highest percentage in all countries based on income level and will be the focus of our work in terns SGBs for organic waste.

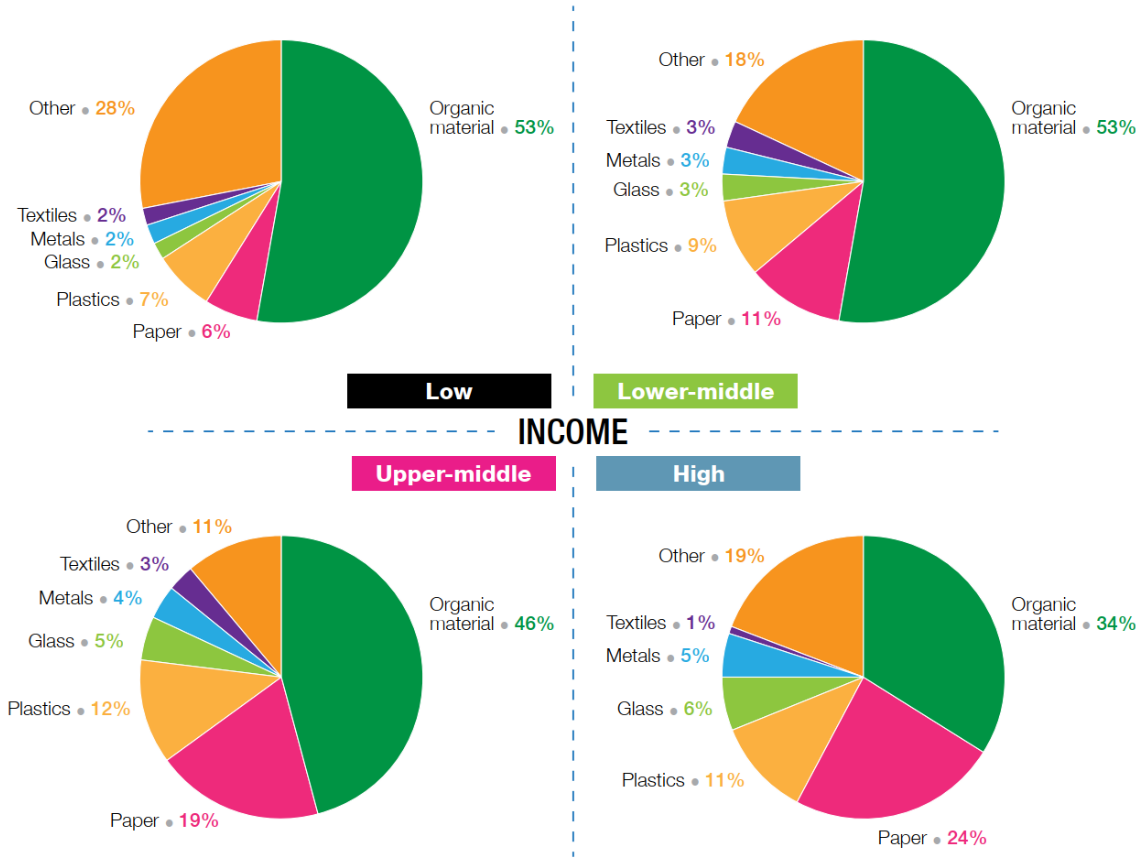

Figure 1 Variation in MSW composition grouped by country income levels [10]

The composition of the different kinds of waste impacts on the density and content which also determines the kind of technology that will be used to collect, treat, reduce, reuse and recycle the waste.

1.2 Motivation

Smart cities' major challenge with waste management is efficiency in waste collection. Many developed cities have sensors embedded into the bins to provide important information on the status of the bin, their contents, although the data gathered is not effectively utilized to manage collection. This traditional approach of waste collection is not practical, time consuming and also costly especially when the bins are not full. IoT enabled smart bins to provide real-time information about the content as well as how disturbing the content could be to the citizens, e.g., stench because of rotten onion and weight of waste which could impact their well-being and quality of life. The waste management system must prioritize waste collection based not only on fill level, but also on other parameters such as temperature, humidity and weight.

A second motivation is to determine when to increase the interval of the sensor updates by the SGB based on the frequency of the same data appearing. For example, if the same temperature, humidity, fill-level and weight values appear more than a specified time, the interval between updates could be increased and the same values predicted. This could help in energy efficiency terms as less energy will be needed when the intervals are increased.

1.3 Research Objectives and Questions

This study focuses on a prioritization of waste collection in St. Petersburg, Russia based on the sensor data from the smart bin and using predictive modelling techniques reduce the number of messages sent from the bins to the server. The successful implementation of the proposed technique creates efficiency in the number of messages sent which impacts the overall energy consumption. It also creates high prioritization of garbage collection depending on parameters such as odour and weight which might impact the well-being of of nearby residents.

The research questions are:

- What predictive model is best suited for waste collection and prioritization?

Objective: Investigate the current prediction methods used in waste management and identify gaps. Determine a suitable model for waste collection prediction.

- How does the proposed approach increase the efficiency of the SGBs?
  Objective: Develop a web based predictive system that reduces the number of messages sent by predicting when to increase the interval of sensor updates.
- How does the proposed system support sustainability?
  Objective: Determine how the proposed approach supports sustainability.

1.4 Research Contribution

The research contribution of this study is the machine learning classification approach to waste collection prioritization. We developed a web-based waste prediction system based on sensor data coming from the SGBs to predict which SGB is full. We also minimized the number of messages and message size sent to the server(cloud) by only sending the collection status of the SGBs. Five classification algorithms were compared, and Decision Tree Classifier was chosen based on accuracy, precision, recall and log loss. With the application of machine learning and IoT, we provide a machine learning approach to reduce the number of messages sent by the SGB by predicting when to increase the interval of sensor updates. The significance of the research is to improve the energy efficiency of the waste management system by reducing the number of messages sent using predictive models.

1.5 Scope and Delimitation

For this thesis, the time of day of waste collection is not taken into account. We also consider only SGBs filled with organic waste in our study to detect. We faced the challenge of a lack of availability of waste collection datasets for use in our study, so it was generated using the python random module. Our dataset was used to pre-train the model based on realistic thresholds of sensor values in SGBs. Decision Trees are known to be culprits of overfitting and this was taken into account by having a training dataset and a test dataset to circumvent this. Finally, it is assumed in our system that sensor data is coming from a LoRaWAN, infrastructure to make reducing the number of messages (by increasing the intervals) feasible for energy consumption.

1.6 Structure of the thesis

This thesis is structured as follows:

Chapter 2 presents the background and related works in the areas of the prediction in waste management.

Chapter 3 describes the Research Methodology, Supervised Machine Learning and Decision Trees in detail.

Chapter 4 gives a detailed description of the Architecture and Implementation.

Chapter 5 presents the results and analysis the results based on our scenario.

Chapter 6 gives a summary of the outcome of the thesis and outline future work needed to be done.

2 RELATED WORK

In this chapter, we discuss in detail the concepts of Internet of Things, Waste Management and Machine Learning and Prediction in Waste Management.

2.1 Internet of Things

The IoT is a novel technology that aims to bring together different digital devices with the Internet, and thus enabling new services in several application domains especially smart cities. It is seen as a global infrastructure for the information society capable of providing advanced services by interconnecting (physical and virtual) devices based on existing and evolving interoperable information and communication technologies [12]. With the aim of making the internet more immersive and pervasive [13], IoT is a revolutionary communication paradigm that will bring forth an invisible and innovative framework to connect a plethora of digital devices with the internet. IoT remains an active research area as it has been identified as one of the key enabling technologies for future smart cities.
In this section, the concept of the IoT and its high-level requirements are reviewed. A detailed discussion of the IoT is provided in [12].

2.1.1 Concept of the IoT

IoT allows people and things to be connected anytime, anywhere, with anything and anyone, ideally using any path/network for any service [14] as illustrated in In Fig. 2. The rapid advancement in the field of wireless communications has made “any TIME communication” and “any PLACE communication” possible with the support of high-speed internet. The IoT seeks to add the dimension “any THING communication”. Through the exploitation of identification, data capturing, processing and communication capabilities, which can be supported by fully integrating leading technologies like advanced device-to-device communication, autonomic networking, data mining and decision making. The IoT will make full use of “THINGs” to offer services to all kinds of applications.

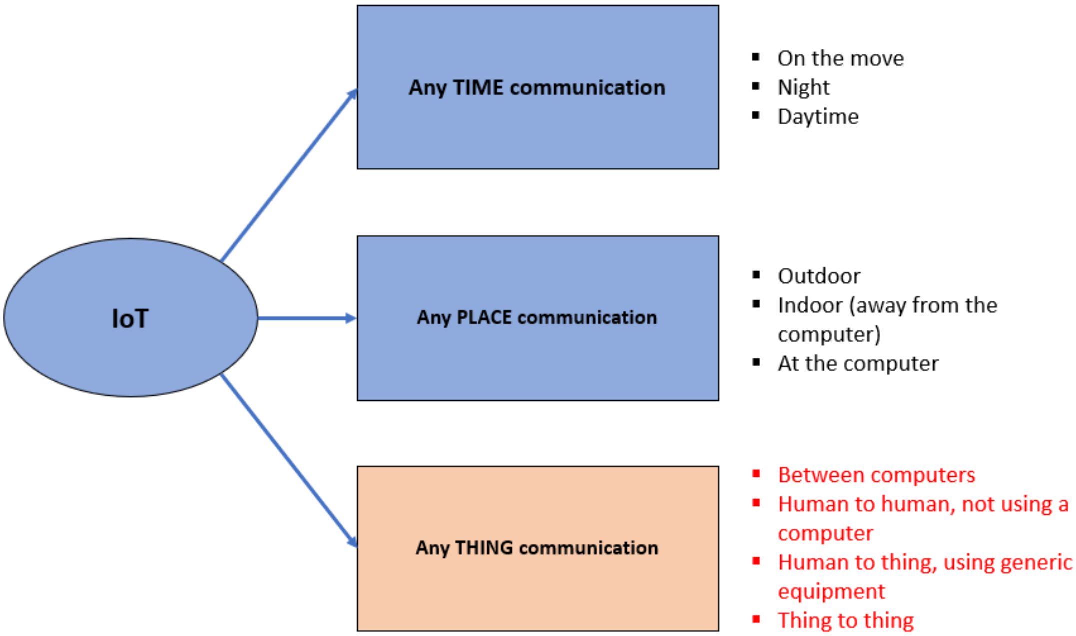

Figure 2 The new dimension introduced in the IoT.

Regarding the IoT, "THINGs" can be physical entities (objects of the physical world) or virtual entities (of the information world) capable of being identified and integrated into a communication network. Physical things such as industrial robots, goods and electrical equipment can be sensed, actuated and connected, while virtual things are capable of being stored, processed and accessed. Typical example of virtual things includes multimedia content and application software.

IoT promises to offer diverse applications while offering numerous benefits like low cost, low energy consumption, high quality-of-service, wider coverage, increased flexibility, increased security, ultra-dense deployments, and multi-vendor interoperability. According to the International Data Corporation (IDC) forecast, the worldwide IoT market is expected to reach US $1.7 trillion in 2020 up from US $655.8 billion in 2014 [15].

2.1.2 Technical overview and requirements of the IoT

This subsection begins with describing the components of the IoT and later discusses the requirements of the IoT. Fig. 3 shows a snapshot of the technical overview and components of the IoT.

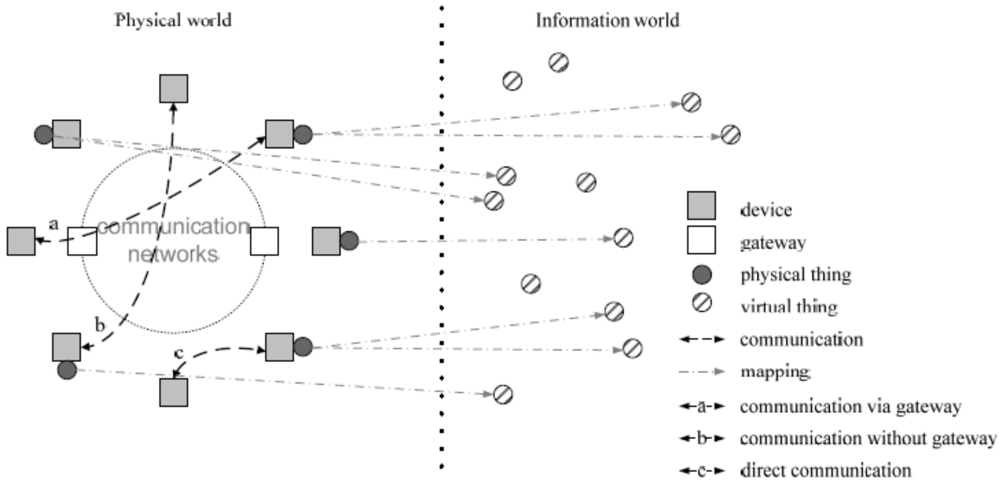

Figure 3 Technical overview of the IoT

The basic components of the IoT are a device, gateway, physical thing, virtual thing and a communication network. A device can be any piece of equipment that has the mandatory capabilities of communication (that is why it is an IoT device) and one or more optional capabilities like sensing, actuation, data capture, data storage and data processing. A device can communicate with other devices primarily in three different ways as illustrated in Fig. 3;

Case a: by communicating with other devices through a communication network via an internet gateway,

Case b: by communicating with other devices through a communication network without a gateway, and

Case c: by communicating with other devices directly without using a communication network.

It is important to mention that a combination of cases a and c as well as cases b and c are also possible. In the IoT, the minimum requirement of a device is their communication capabilities. Fig. 4 shows the different categories of an IoT device and their respective functions.

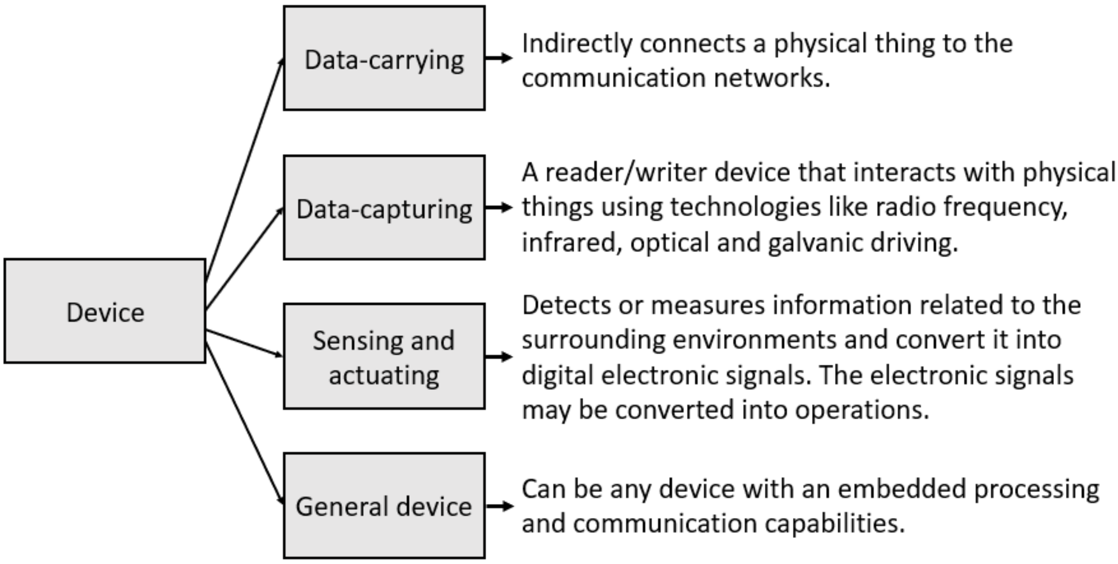

Figure 4 different type of IoT devices and their functions.

A "thing" can be any object of the physical world (physical thing) or the information world (virtual thing) that has the software and hardware capabilities of being identified and integrated into a communication network. Although interactions between physical things are shown in Fig. 4, there can be exchanges between virtual things in the information world and between a physical thing and a virtual thing. Physical things are normally mapped to virtual things unlike virtual things that can exist without being associated to any physical thing. The communication network is responsible for transferring data captured by devices (sensing/actuating, data capturing devices) to applications and/or other devices as well as instructions from applications to devices [12]. The communication network is one of the critical components of the IoT since it determines how reliable and efficient data transfer can be in an IoT infrastructure. This communication network can be realized using a conventional TCP/IP-based networks and/or cellular network.

2.2 Waste Management and IoT

IoT is a perfect fit for waste management services because of the decreasing cost of sensors; paving way for larger and cost-effective deployments. A smart garbage bin (SGB) is a waste bin outfitted with sensors that can detect fill-level, temperature, humidity and weight among others.

The vast amount of data from SGBs offer some immediate insights and obvious benefits like collecting a bin when it is full based on fill level sensor to prevent overflow or monitoring the internal state of the bins. One of these applications is waste prioritization based on fill-level, temperature, humidity and weight and not just on fill-level. This can help curtail bad odours coming from the bins and damage to the bins due to heavy material in the bins [16].

The authors in [17] present a RFID and sensor model for designing a real time waste management system. They further describe an architecture that can track waste identity(customer), weight of bin, missing/stolen bins more efficiently. These methods and techniques are shown to reduce waste management costs.

With RFID and sensor technology, waste management systems can be optimized to increase productivity and save costs. IoT-enabled applications are effectively improving the waste management operations. Pre-defined routes and outdated methods of waste collection are increasingly being replaced with sensor-enabled bins and sophisticated waste management applications based on IoT.

Waste management is becoming a severe issue in the present century. Population increase, lack of responsibility towards the environment, improper resource management, etc. are the primary reasons for excessive generation of garbage [18]. In every subset of the waste management concept, few difficulties emerge and settling them is conceivable given that they are delivered to processing plants, and effectively processed quickly. The challenge in rubbish accumulation and checking is quite complicated as the rubbish is generally appropriated, widely created, and for the most part unmanaged [19].

[20] presents a smart waste collection system with IoT enabled waste bins. Fill level data of the waste bins are sent via the Internet to a server for storage and decision making based on the data collected. Optimized collection routes are generated for the truck drivers. The paper sets up three cases for waste collection; vehicle sent to all locations, vehicle sent to identified locations based on fill level thresholds and vehicle sent to identified locations based on

historic fill level frequency. The strength of the paper is the clear description of the components of the waste collection system and the proposed approach.

[21] presents a multi-agent system for waste collection that uses a modified ant colony optimization method to solve the dynamic vehicle routing problem. Real time data about the utilized capacity of the trucks after visiting a waste collection point is used in generating dynamic routes. The system utilizes clustering algorithms to determine the location of waste collection points. A performance evaluation of static routes and the proposed dynamic approached is done and evaluated based on distance travelled by trucks and the dynamic approach was shown to be better. The authors do not consider IoT enabled bins in their multi agent system.

One waste collection approach without prediction is [22], a model for waste collection that takes into account real time traceability data. The sources of the real time data mentioned in the paper include volumetric sensors, GPS, RFID and GPRS. Constraints on the waste bin and trucks are outlined as serve as inputs to the model for optimal routes. The objective of the routing model is to minimize total distance covered, travel time, number of trucks per fleet and impact to environment. The results of evaluating the model showed clear benefits in utilizing real time traceability data in generating routes for waste collection.

2.3 Machine Learning Algorithms

Machine learning gives devices the capability to learn through pattern recognition and computational learning. It improves the efficiency of systems through data mining and analysis [23]. Learning can be categorised into supervised learning which samples input data and its corresponding target data, unsupervised learning which does not require target data and reinforcement learning which takes the applicable action depending on the situation at hand. Several algorithms exist whose different features can be compared as shown in the figure 5. The strengths and weaknesses of each approach is highlighted in comparison with the rest. The performance metrics like accuracy, precision and recall can be determined based on how well they perform on specific datasets. Decision Trees and Rule learners show a close operational profile and SVM and Neural Networks show a close operational profile based on the comparisons.

	Decision Trees	Neural Networks	Naïve Bayes	kNN	SVM	Rule-learners
Accuracy in general	**	***	*	**	****	**
Speed of learning with respect to number of attributes and the number of instances	***	*	****	****	*	**
Speed of classification	****	****	****	*	****	****
Tolerance to missing values	***	*	****	*	**	**
Tolerance to irrelevant attributes	***	*	**	**	****	**
Tolerance to redundant attributes	**	**	*	**	***	**
Tolerance to highly interdependent attributes (e.g. parity problems)	**	***	*	*	***	**
Dealing with discrete/binary/continuous attributes	****	***(not discrete)	***(not continuous)	***(not directly discrete)	**(not discrete)	***(not directly continuous)
Tolerance to noise	**	**	***	*	**	*
Dealing with danger of overfitting	**	*	***	***	**	**
Attempts for incremental learning	**	***	****	****	**	*
Explanation ability/transparency of knowledge/classifications	****	*	****	**	*	****
Model parameter handling	***	*	****	***	*	***

Figure 5 Comparing learning algorithms (** represent the best and * represent the worst performance)**[23]

It is not easy to constructively appraise each algorithm in terms of accuracy and performance, this is because each algorithm has its own strength which is dependent on the situation at hand. Combination of algorithms for different problems are in some cases better than individual algorithms running in isolation as they improve performance and also generate more certain and accurate system results. However, the integration of multiple algorithms creates a complexity for non-experts users to comprehend the processes leading to the final decisions (reduces comprehensibility), require more storage space, and need more computational performance.

This study considered several classifiers before settling on decision trees for the prediction decision support system. This include:

- K-Nearest Neighbour Algorithm

It is a simple pattern classification algorithm whose performance depends on the distance metric used to identify nearest neighbours. In situations where there is no prior information, the classifiers use Euclidean metric to measure the contrast in the vector inputs.

Equation 1 KNN Equation

$$p(t = c|x, K) = \frac{1}{K} \sum_{i \in N_k(x)} 1(t_i = c)$$

$$y = \arg_c \max p\ (t = c|x, K)$$

- Support Vector Machine

It's a supervised learning algorithm that is used to solve both classification and regression issues. It identifies the dividing hyperplane which best splits the training set from the maximum margin. The distance between the support vector and the hyperplane need to be as far as possible.

Equation 2 SVM Constrained Optimization Problem

$$minimize_{w,b,\xi} \quad f(w, b, \xi) = \frac{1}{2} w^T w + c \sum_{i=1}^{n} \xi_i$$

$$subject\ to \quad y_i(w^T x_i + b) - 1 + \xi_i \geq 0 \quad 1 = 1, \cdots, n, \xi_i \geq 0 \quad i = 1, \cdots, n$$

- Naive Bayesian

It is a simple stochastic classifier based on Bayes theorem with strong independent assumptions. Naive Bayesian assumes that the classification features are independent and do not have any interrelationships in any way. Bayes' theorem or Bayesian Networks make definite assumptions on variable dependencies.

Equation 3 Naive Bayes Classifier

$$y = \arg_c \max p(t = c) \prod_{j=1}^{M} p(z_j|t = c)$$

- Random Forest

A classification and regression algorithm which operates by constructing a multitude of decision trees at training time. The decision of the majority of the trees is chosen as the final decision. The advantages of random forest is its ability to attain high accuracy during predictions and takes less training time.

- Decision Tree

This algorithm is widely used technique effective for classification and regression. Decision Trees represent possible observations, occurrence or reaction in the branches and target values (conclusion, classification or decisions) in the leaves. The decision tree breaks down the dataset broken into subsets and any associated decision tree is developed incrementally. To avoid a situation of overfitting, the decision tree is pruned so as to not grow to its full size.

Equation 4 Decision Tree Classification

$$p(t = c|k) = \frac{1}{|R_k|} \sum_{i \in R_k} 1(t_i = c),$$

$$y = \arg_c \max p\ (t = c|x) = \arg_c \max p\ (t = c|k)$$

An example of machine learning algorithms performance comparison on network applications QoS and dynamic access control reveals that the computational performance metrics which are build time and classification speed were different for each algorithm. For instance, Naive Bayesian performed better in build time than decision trees while in classification speed decision trees identified network flows faster than Naive Bayesian. This is to emphasise that each algorithm under certain conditions and applications performs better than the other.

2.4 Prediction in Waste Management

IoT will be the most impactful sources of new data, and data science will contribute to making IoT services and applications more intelligent. Data science is the amalgamation of data mining, machine learning and other techniques to find patterns and gain new insights from data. Data models including neural networks, regression and clustering methods are applied to match with the data characteristics and gain insights. The argument is made for smart data to be a good representation for IoT data. “Smart data provides value from harnessing the challenges posed by volume, velocity, variety and veracity of big data and in turn providing actionable information and improving decision making [24].

Using smart cities as a use case, the authors in [24] assess different machine learning approaches that focus on challenges when dealing with IoT data. They present a taxonomy of algorithms detailing how different techniques are used with the data to get higher level information. The key contribution by them is the way data characteristics and application specifics could lead to choosing the right analytics algorithm.

Decision trees and neural networks were applied to build models in [7]. Neural networks performed better with 72% of variation in the data. The authors created tools that assist in regional waste planning by pre-processing public data from various sources taking into consideration socio-economic explanatory variables. Their approach shows the high feasibility for waste prediction systems.

In [4], the authors use machine learning to gather information about habits of waste generation in a specified region to predict the amount of waste generated in the future saving time and money. Immediate benefits of this approach such as less air pollution was highlighted.

The objective of the system described in [13] is to utilize accurate fill-level predictions to predict future routes and improve costs by minimizing distance travelled by a truck to collect SGBs. An improved quality of service is also highlighted as an advantage because the system counteracts overflowing bins.

An intelligent predictive software system designed to use the knowledge and technology that follows current trends in designing intelligent systems. We are convinced that our intelligent system will help to improve the quality of life of the city where it is implemented, due to the reduction of costs and the efficient treatment of the waste generated, which will constitute a way to achieve the sustainability in a city.

3 METHODOLOGY

A variant of the waterfall model [25] is used in managing our study due to our requirements clearly outlined in the beginning.

In Phase I, we focused on investigating the gaps in the literature with an emphasis on machine learning in waste management. The background and related work on Internet of Things, Waste Management and Machine Learning are presented. The research questions, objectives and scope are formulated as well.

In Phase II, Machine learning models chosen for this thesis are presented and discussed. The supervised machine learning was chosen as the preferred learning approach for the work.

In Phase III, we investigated the data using exploratory data analysis techniques to gather initial insights from the data.

In Phase IV, we gave a detailed view of the implementation of the Waste Prediction System for Decision Support, run experiments and graph the results.

In Phase V, we presented an analysis and validation of the results. We provided insights gained from the results, drew conclusions and gave avenues for further research.

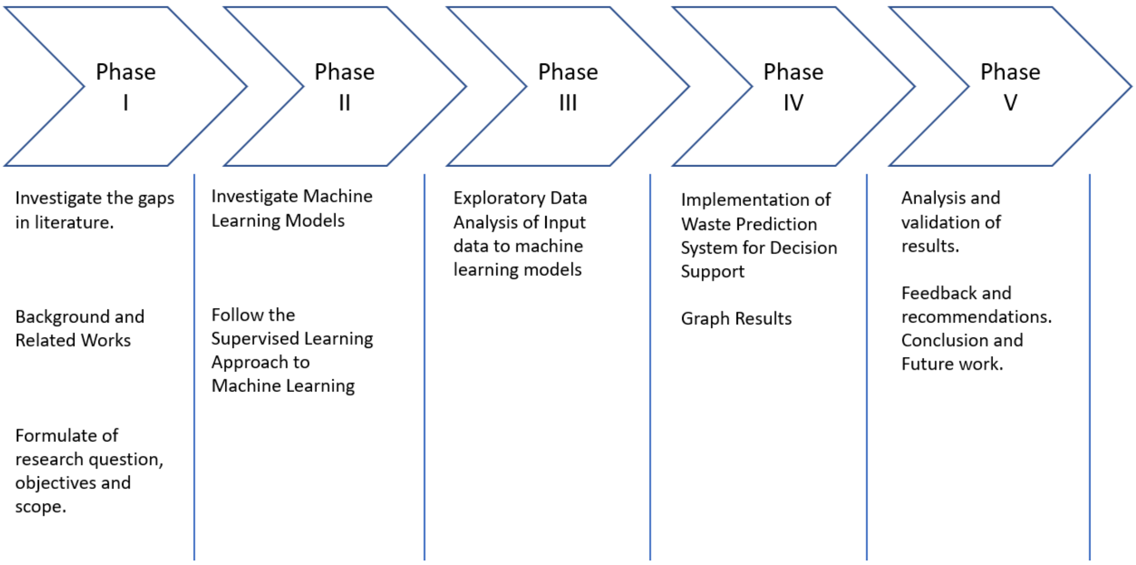

Figure 6 Waterfall Model

4 IMPLEMENTATION

In this section, we discuss the implementation of the decision support system and the choice of technologies used for the study.

4.1 Web Application

We implemented a web-based frontend for the predictive engine to provide a visual interface to interact with the engine for input and output. The frontend interacts with the predictive engine to generate sensor information, interval information and show alerts from the predictive engine. The web-based frontend was developed with a web framework called VueJS [26].

VueJS is an open-source JavaScript framework for building user interfaces. VueJS uses the concept of components to design and build modular web applications. Vue is an approachable, versatile and performant framework that helps create a maintainable and testable web application.

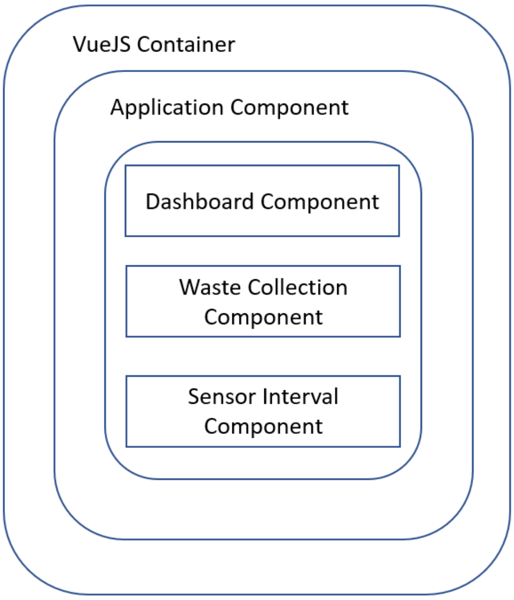

Figure 7 Web application components

In the design of the frontend, three components were created to provide the desired functionality of waste prioritization and increasing sensor intervals. The components are Dashboard Component, Waste Collection Component, and the Sensor Interval Component.

Dashboard Component: The component is made up of a Google Maps interface with a marker highlighting SGB points in St. Petersburg, Russia. The markers are clickable and redirect the user to the Waste Collection Component.

Waste Collection Component: This component provides the sensor input for the predictive engine when it comes to waste prioritization. The form fields provide input for temperature, humidity, fill-level and weight. An alert dialog is present as output from the predictive engine after the sensor inputs are submitted. A simulation of randomly generated sensor values every 5 seconds is sent to the predictive engine to provide a response. This is set on the component to mimic the behaviour in production.

Sensor Interval Component: This component provides the sensor input and sensor frequency information for the predictive engine when it comes to changing sensor intervals. The form fields provide input for temperature, humidity, fill-level and frequency of the observed sensor value. An alert dialog is present as output from the predictive engine after sensor inputs are submitted. A simulation with randomly generated sensor values every 5 seconds is sent to the predictive engine to provide a response. This is set on the component to mimic the behaviour in production.

In figure 8 presented, we depict sensor data collected at a sensor node and aggregated in a gateway to be sent to the cloud for processing and predictive analytics. Each sensor node represents sensor data coming from a SGB.

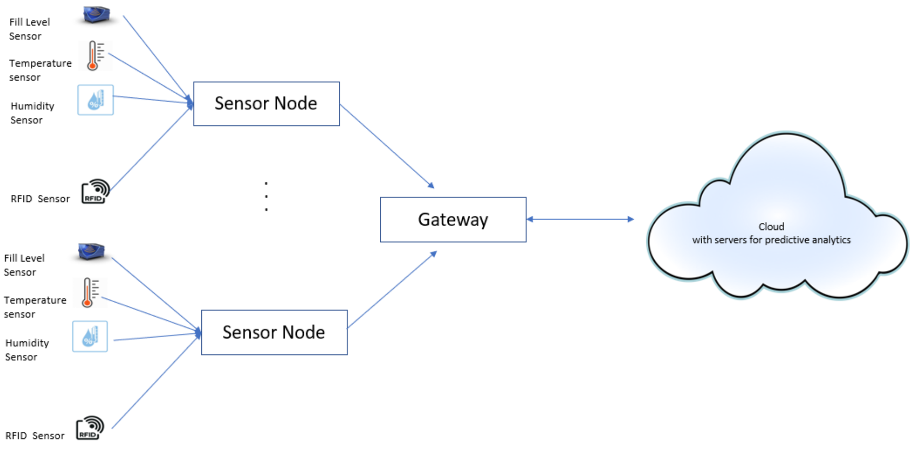

Figure 8 Sensor Architecture

Figure 9 describes the process of the decision support tool for changing sensor intervals. When sensor values are generated, and the frequency of the values determined, it is sent to the predictive engine which responds with a binary value. If the interval is to be changed, a notification is sent, the wait period triggered, and the process repeated. If the interval is to remain the same, a notification is sent, a wait period triggered, and the process repeated

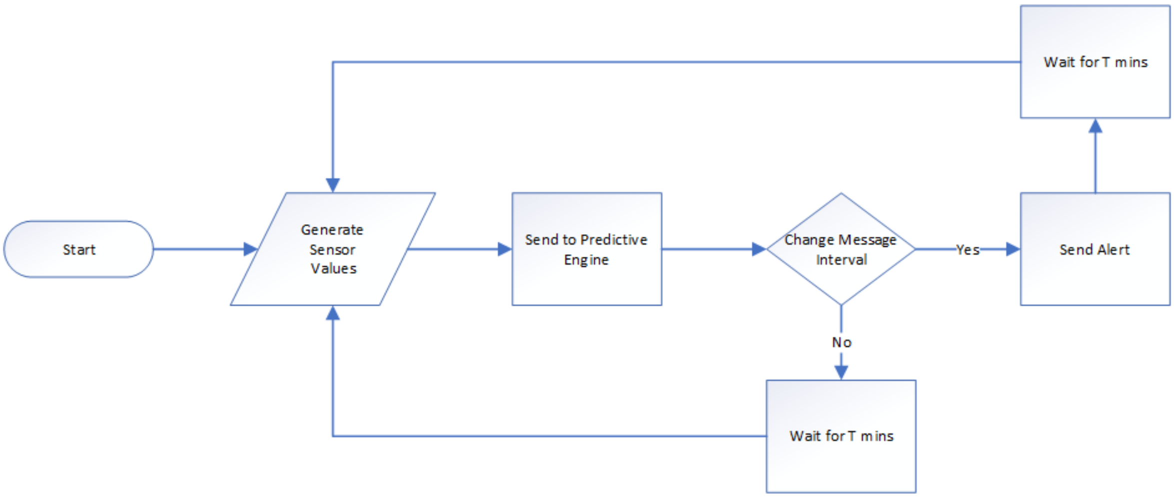

Figure 9 Flow Diagram for Sensor Interval

Figure 10 describes the process flow of the decision support tool based on decision trees. When sensor values are generated and sent to the pre-trained predictive engine, it responds with a binary value determining whether to collect waste or not. If waste is to be collected, an alert is sent and wait period triggered before sensor values are generated. If waste is not to be collected based on the engine's response, a wait period is triggered, and new sensor values are generated and the process repeated.

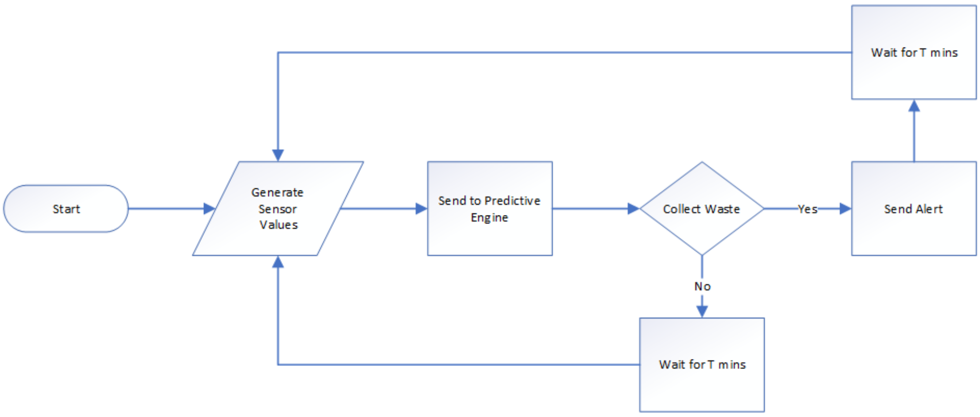

Figure 10 Flow Diagram for Waste Prediction

4.2 Web Service

The predictive engine is encapsulated with a REST based API to provide an interface for the frontend. Two endpoints are developed to interact with the frontend:

/waste: This REST endpoint hosts a pre-trained decision tree that predicts when a bin is full based on all sensor information and not just fill level. Priority is given to heavy waste and smelly waste to prevent bin destruction and citizen complaints.

/interval: This REST endpoint hosts a pre-trained decision tree that predicts the interval of the sensor updates based on the frequency of same sensor values. The decision tree predicts when to increase the interval of sensor updates or set the interval to a limit.

The message flow from the web application to the API backend follows the conventional request response approach in figure 11. The simulated sensors make requests to the API backend and the predictive engine responds as shown below

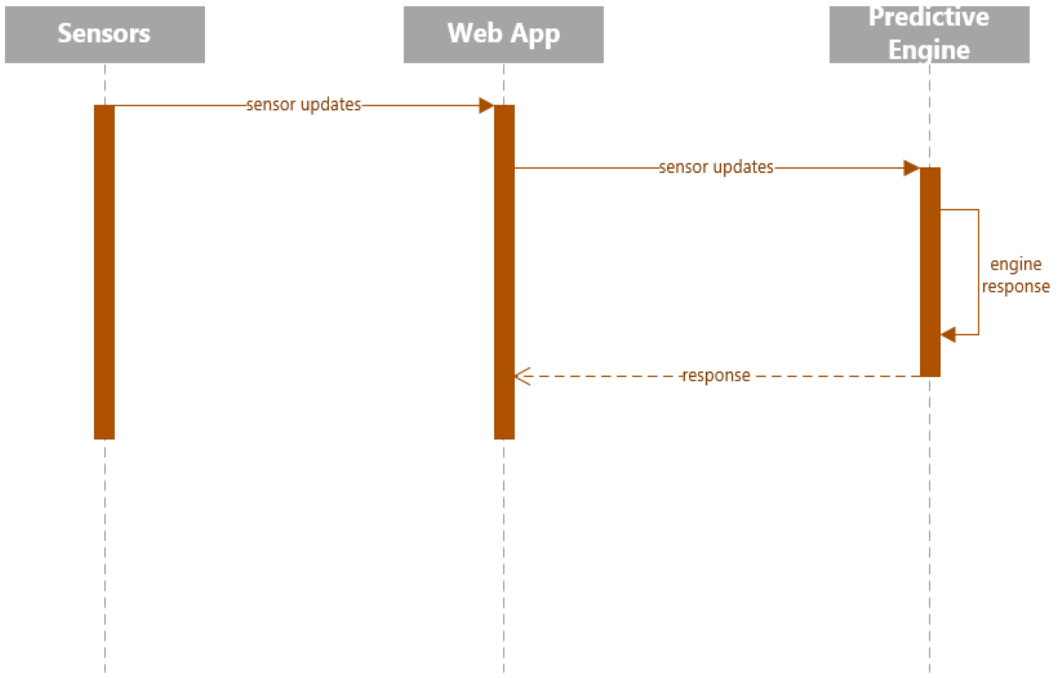

Figure 11 Sequence Diagram of Web application

4.3 Metrics

In this section we discuss three metrics for model evaluation, accuracy, precision and recall showcasing the relevance of our models. These metrics will be used in the next chapter to choose the best model for our predictive engine.

4.3.1 Accuracy

Accuracy is a key metric for classification models. Accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:

$$Accuracy = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

For binary classifications, it is calculated as follows:

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives

4.3.2 Precision

Precision is defined as the number of True Positives out of the number of True and False Positives. A low precision is an indication of a large number of False Positives.

For binary classifications, it is calculated as follows:

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

4.3.3 Recall

Recall is defined as the number of True Positives out of the number of True and False Negatives. A low recall indicates a large number of false negatives.

For binary classifications, it is calculated as follows:

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

4.3.4 Log Loss

Logarithm loss (log loss) is defined as a performance measure where the predicted input is a probability between 0 and 1. A lower log loss is preferred and machine learning models' goal is to minimize this value.

For binary classification, the log loss is defined as

$$-\frac{1}{N}\sum_{i=1}^{N}[y_i \log p_i + (1 - y_i)\log(1 - p_i)]$$

Where N = Number of samples

p_i = model probability

y_i = binary indicator for correctly classified label

5 RESULTS AND DISCUSSION

In this chapter, we discuss the results of comparing the 5 ML algorithms and how it reduces the number of messages sent by the SGB.

5.1 Comparison of Machine Learning Algorithms

Table 5.1 describes four performance metrics namely Accuracy, Precision, Recall and Log Loss to evaluate the chosen models. From the graph in figure 12, Decision Trees and Random Forests have the best accuracy score for this dataset with an accuracy score of 100%. K Nearest Neighbour had a lower accuracy score of 96.75%, followed by Gaussian Naive Bayes with a score of 86.60% and last being Support Vector Classifier with an accuracy of 55.55%.

Table 5. 1 Accuracy score and log loss of classifiers

Classifier	Accuracy(%)	Precision(%)	Recall(%)	Log Loss
K Nearest Neighbour	96.75	96.18	96.51	2.508466e-01
Support Vector	55.55	0.0	0.0	5.362928e-01
Decision Tree	100.00	100.00	100.00	9.992007e-16
Random Forest	100.00	100.00	100.00	6.321631e-04
Gaussian Naive Bayes	86.60	84.23	85.94	3.687910e-01

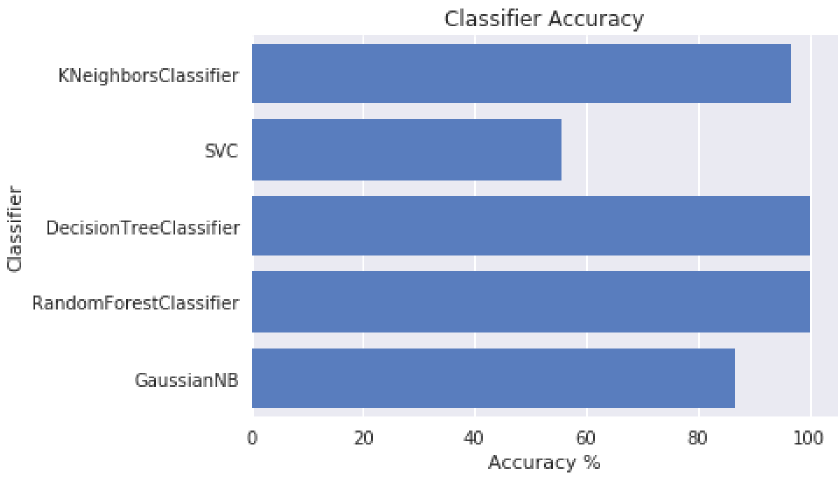

Figure 12 Accuracy of Machine learning classifiers on dataset

From table 5.1 and figure 13, Decision Tree and Random Forests have the best precision and recall score of 100% for the dataset. The high result of Decision Trees and Random Forests indicate that the proportion of data points that these models said were relevant were indeed relevant. Support Vector Classifier performed the least in terms of 0% in precision and recall respectively. A 0% precision and recall indicates that there are no true positives. Figure 13 and 14 show the visual representation of table 5.1 of precision and recall.

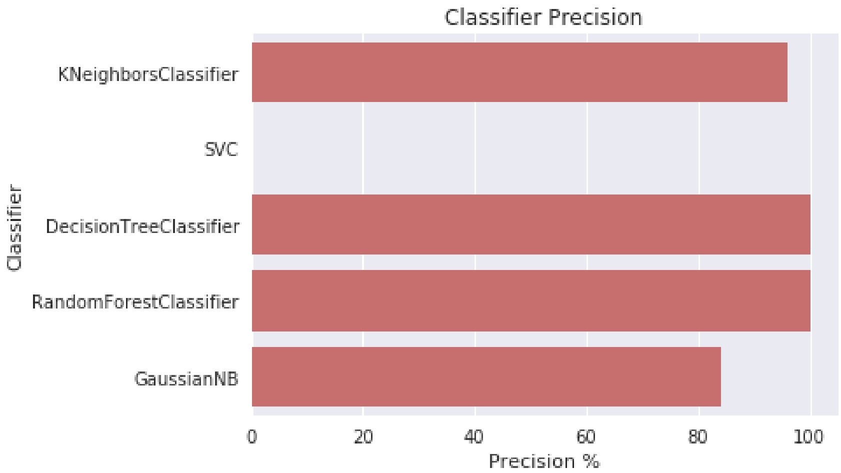

Figure 13 Precision of Machine learning classifiers on dataset

The high precision and recall of decision trees could indicate high variance or overfitting of the sample dataset. We utilized the train test split approach to avoid this scenario with 80% training data and 20% test data.

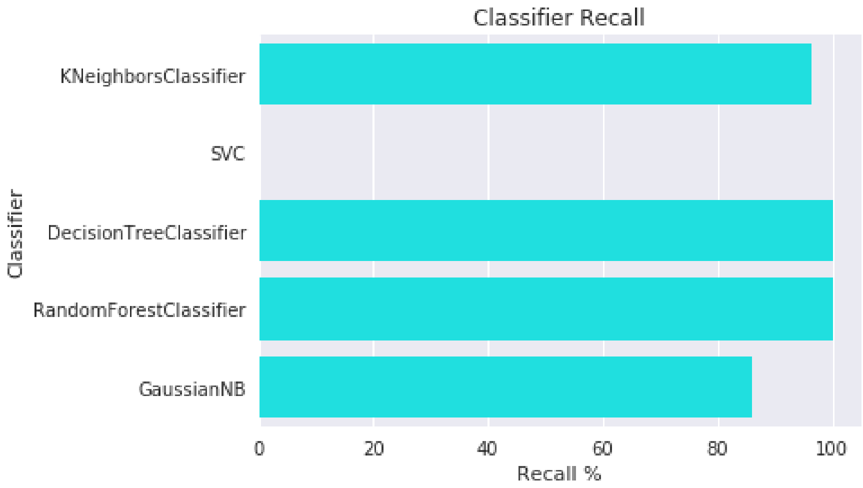

Figure 14 Recall of Machine learning classifiers on dataset

Logarithmic loss(Log loss) is a measure of the performance of a model where the prediction input is a probability between 0 and 1. The goal of every model is to minimize this value. The lower the log loss, the better. From table 5.1 and figure 15 , decision tree classifier has the lowest log loss, followed by random forest classifier, K Nearest Neighbour Classifier, Gaussian Naive Bayes classifier and the worst being the Support Vector Classifier.

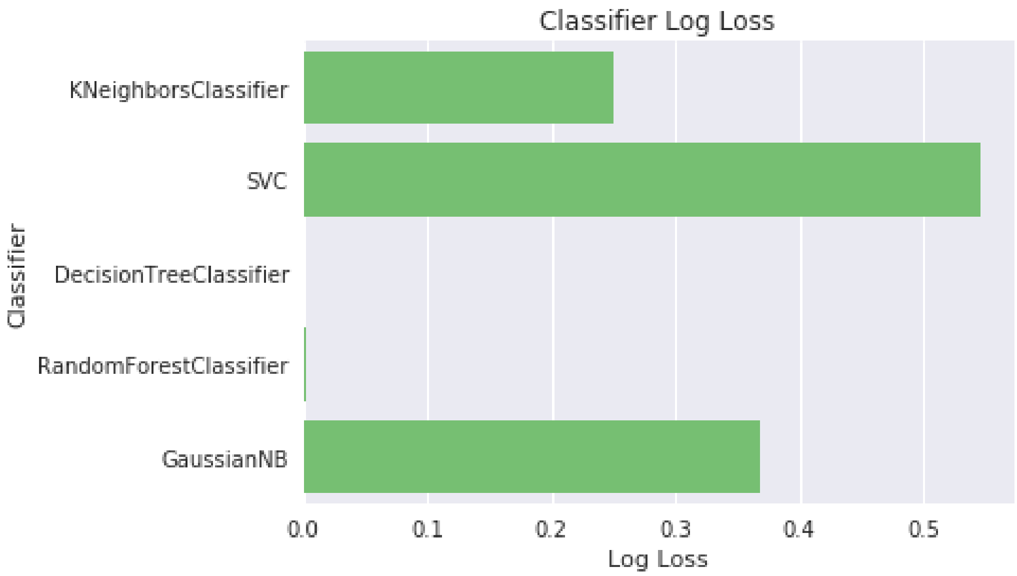

Figure 15 Log loss of machine learning classifiers on dataset

From the table below, Fill level and temperature have significantly higher feature importance to showcase and highlight their importance in determining when a bin needs to be collected.

Table 5. 2 Feature Importance

Feature	**Importance**
Fill_level	0.416359
Temperature	0.407947
Humidity	0.121646
Weight	0.054048

5.2 Number of Messages Reduction

Figure 16 represents the User Interface for Sensor Interval Information. Sensor data and frequency are generated randomly and sent to the predictive engine for processing. The engine responds via an alert dialog as to whether to increase the interval or leave interval unchanged. This occurs for each sensor update and frequency.

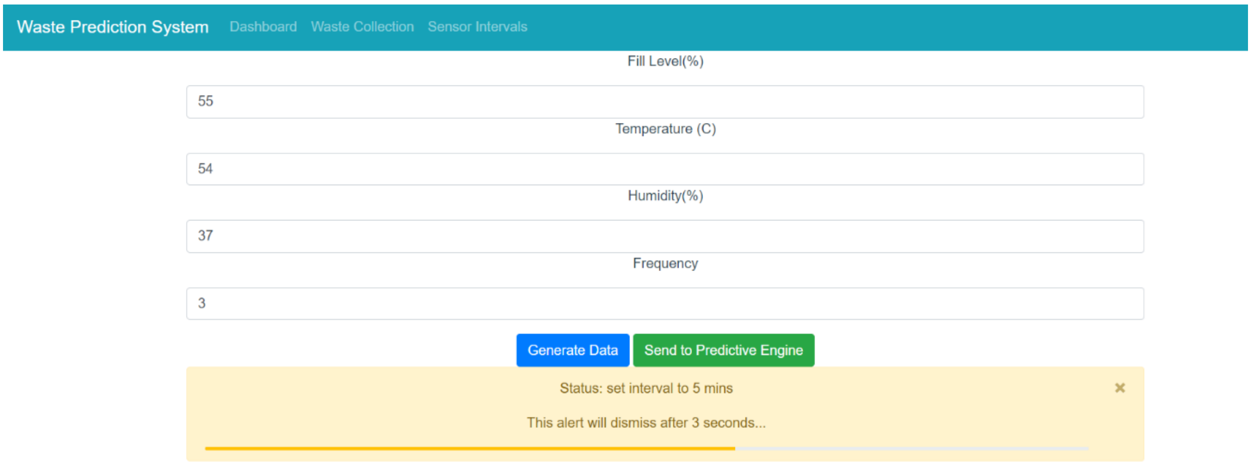

Figure 16 Sensor Interval Prediction Interface

Figure 17 represents the User Interface for Waste Collection Information. Sensor data are generated randomly and sent to the predictive engine for processing. The engine responds via an alert dialog as to when to collect waste or not.

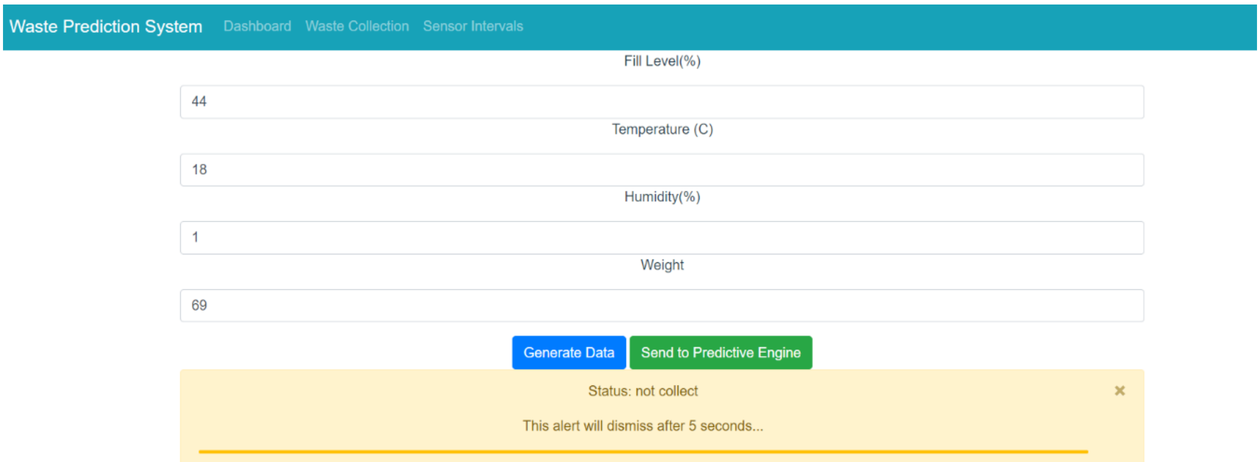

Figure 17 Waste Collection Interface

The graph in figure 18 is a graph of the message count before and after predictive modelling of the SGB when it comes to reducing the number of messaging sent by the SGB. Assuming that each message is sent by the bin every 5 mins.

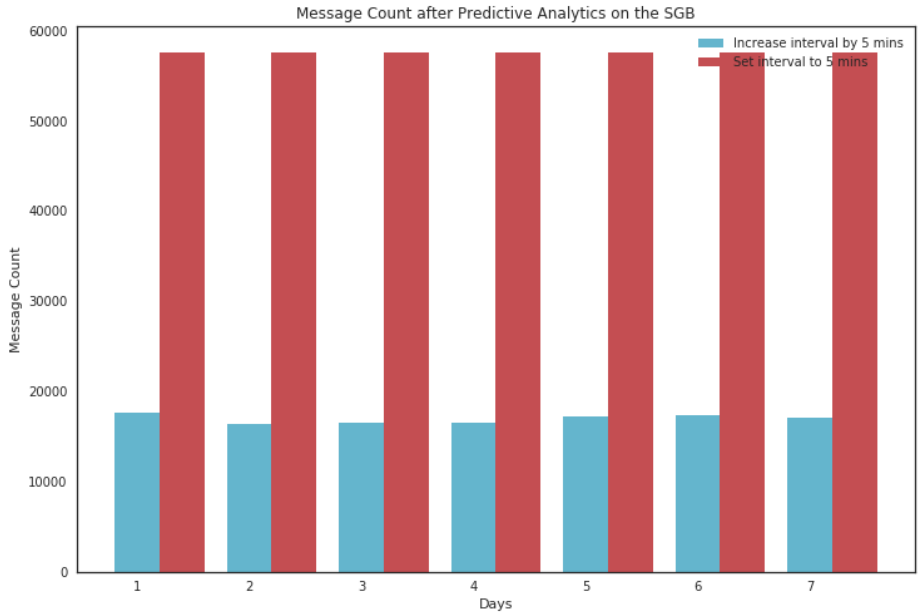

Figure 18 Graph of message count before and after prediction

For the use case of 200 bins, the number of datapoints needed for a full day is

$$Number\ of\ datapoints\ for\ 1\ bin$$
$$= \frac{\text{Number of hours in a day x Number of minutes in an hour}}{\text{5 mins}}$$
$$\frac{24 \text{ x } 60}{5}$$
$$288\ data\ points$$

$$Number\ of\ Datapoints\ for\ 200\ bins = Number\ of\ Data\ points\ for\ 1\ bin\ x\ 200$$
$$= 57600\ data\ points$$

Assuming the 57600 data points without prediction occurs, the table below highlights the significant reduction in the number of messages when intervals are increased by 5 minutes with the predictive engine.

The second column in table 5.3 represents the number of messages that are sent by the sensors every day with an interval of 5 minutes always. The third column represents the number of messages that are sent after increasing the interval between updates based on frequency of the same sensor values. As the frequency of the same sensor updates increases, the values can be predicted, and the interval increased to save energy at the sensor node. From the table, we see that Day 3 has the highest message count of 26106 after prediction. The percentage of the original message count is between 44% to 45% after increasing the interval.

Table 5. 3 Message count before and after prediction

Day	Message Count Before Prediction (5 min interval)	Message Count After Prediction (10 min interval)	% of Original Message Count
1	57600	25886	44.94
2	57600	25773	44.74
3	57600	26106	45.32
4	57600	25953	45.06
5	57600	25967	45.08
6	57600	26077	45.27
7	57600	25930	45.02

5.3 Sustainability

Waste management is everybody's responsibility in the society. Individuals and the society at large are consuming resources and products faster than the production, usage and disposal phases. This way of living is not sustainable and depletes resources for future generations. Predictive waste management helps ensure that resources such as energy, personnel, vehicles, technology resources are efficiently used to ensure better quality of service provision and reduced operational costs. A direct sustainability impact is by reducing the number of messages sent to the cloud, there are energy savings that make it inherently energy efficient.

Sustainability in waste management is addressed by [5] detailing how the move to a more sustainable society will require greater complexity in managing waste. The Ministry of Environment in New Zealand is used as a case study for this paper. The paper details visual representations of the interconnection between waste management components and the different sections of the Ministry. A parallel between the features of a complex adaptive system and a sustainable waste management system is made by emphasizing the dynamic nature of both systems. The definition and requirements of a sustainable waste management is stated as well as how the process to move to a sustainable waste management system is attained.

Sustainability is not only limited to environmental resources but also societal and individual well-being, economic growth and improved technological advancements. Sustainability dimensions can be classified as follows:

Individual: Focusing on the ability for human beings to thrive and experience dignity and fulfilment. Overflow of waste bins are reduced.

Social: mutually benefiting relationships between individuals in a society

Economic: the ability for add financial value to businesses

Technical: Ease of systems use, maintainability and its ability to evolve. Reduced energy uses because of reduction in number of messages sent periodically.

Environment: stewardship on resource usage to reduce the harm on the environment.

The argument can be made that from a sustainability point of view, municipalities should invest in educating citizens on producing less waste instead of investing in more efficient waste management systems. We believe a balance can be reached as the existence of such systems play a big role in good waste management practices because of future population explosion in cities.

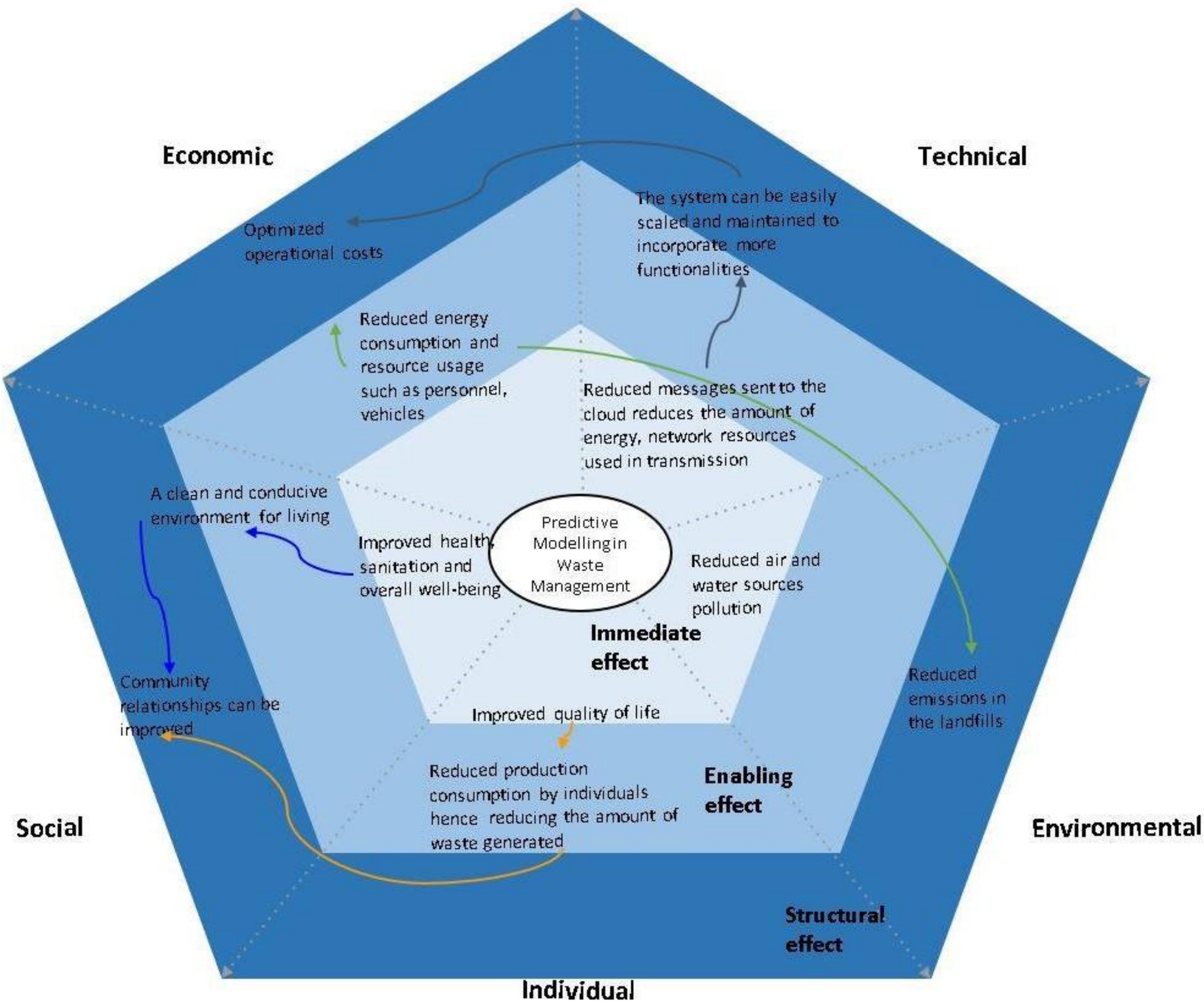

Figure 19 Sustainability Analysis Diagram

6 CONCLUSION

In this chapter, we discuss the conclusion and future work. The limitations of the thesis are also discussed. The objective of the thesis is to develop a predictive engine to predict when waste needs to be collected and to reduce the number of messages sent by the SGB.

6.1 Summary

Machine Learning Classification models provide an optimal way to improve waste management systems by predicting when a SGB needs to be collected and when the interval between sensor updates needs to be increased. The objective of the research was i) Investigate the state of the art in prediction in waste management identifying gaps ii) Determine the best predictive model for use in a waste management system iii) Determine how the proposed model can reduce the number of messages sent to the cloud by the SGB and finally iv) determine how the proposed approach supports sustainability. We investigated five machine learning models namely Random Forests, Decision Trees, Gaussian Naive Bayes, K Nearest Neighbours, Support Vector Classifiers and determined that decision trees were suitable for our use case based on accuracy score, precision score, recall score and logarithmic loss. We showed that by using a predictive model, we could predict when to increase the interval based on frequency of similar sensor updates. A percentage ranging from 44% to 45% of the original daily message count is now sent daily as shown in the results and discussion chapter. This reduction leads to an efficient use of energy by the SGB sensors as they must send fewer and with the energy efficient approach promotes sustainability directly.

6.2 Future Work

Our work did not incorporate the time of day in the prediction and could be done in the future work. Also, deep learning techniques were not explored and poses a clear gap in our research as it is the state of the art in machine learning. Regression approaches could be explored and compared with the classification approach.

REFERENCES

[1] J. Porras, A. Seffah, E. Rondeau, K. Andersson, and K. Alexandra, “PERCCOM: A Master Program in Pervasive Computing and COMmunications for Sustainable Development,” no. APRIL, 2016.

[2] “World population projected to reach 9.8 billion in 2050, and 11.2 billion in 2100 | UN DESA | United Nations Department of Economic and Social Affairs.” [Online]. Available: https://www.un.org/development/desa/en/news/population/world-population-prospects-2017.html. [Accessed: 04-Oct-2018].

[3] N. E. Johnson *et al.*, “Patterns of waste generation: A gradient boosting model for short-term waste prediction in New York City,” *Waste Manag.*, vol. 62, pp. 3–11, 2017.

[4] C. J. Baby, H. Singh, A. Srivastava, R. Dhawan, and P. Mahalakshmi, “Smart bin: An intelligent waste alert and prediction system using machine learning approach,” *Proc. 2017 Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2017*, vol. 2018–Janua, pp. 771–774, 2018.

[5] J. K. Seadon, “Sustainable waste management systems,” *J. Clean. Prod.*, vol. 18, no. 16–17, pp. 1639–1651, 2010.

[6] K. A. Kolekar, T. Hazra, and S. N. Chakrabarty, “A Review on Prediction of Municipal Solid Waste Generation Models,” *Procedia Environ. Sci.*, vol. 35, pp. 238–244, 2016.

[7] M. Kannangara, R. Dua, L. Ahmadi, and F. Bensebaa, “Modeling and prediction of regional municipal solid waste generation and diversion in Canada using machine learning approaches,” *Waste Manag.*, vol. 74, pp. 3–15, 2018.

[8] P. P. Ed and R. Goebel, *LNAI 7987 - Advances in Data Mining*, no. July. 2013.

[9] M. Mijac and R. Picek, “Smart City Services Driven By Iot : a Systematic Review,” *J. Econ. Soc. Dev.*, vol. 4, no. 2, pp. 40–51, 2017.

[10] P. Modak, D. C. Wilson, and C. Velis, *Waste Management: Global Status*. 2015.

[11] “A Brief Introduction to Waste Management.” [Online]. Available: https://www.gdrc.org/uem/waste/waste-intro.html. [Accessed: 30-Sep-2018].

[12] “Y.2060 : Overview of the Internet of things.” [Online]. Available: https://www.itu.int/rec/T-REC-Y.2060-201206-I. [Accessed: 30-Sep-2018].

[13] A. Zanella and N. Bui, “Internet of Things for Smart Cities,” *... Internet Things ...*, vol. 1, no. 1, pp. 22–32, 2014.

[14] “IERC-European Research Cluster on the Internet of Things.” [Online]. Available: http://www.internet-of-things-research.eu/. [Accessed: 30-Sep-2018].

[15] Y. Mehmood, F. Ahmad, I. Yaqoob, A. Adnane, M. Imran, and S. Guizani, “Internet-of-Things-Based Smart Cities: Recent Advances and Challenges,” *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 16–24, 2017.

[16] “How IoT is making waste management smarter.” [Online]. Available: https://www.reminetwork.com/articles/iot-waste-management-smarter/. [Accessed: 19-Sep-2018].

[17] B. Chowdhury and M. U. Chowdhury, “RFID-based real-time smart waste management system,” *2007 Australas. Telecommun. Networks Appl. Conf. ATNAC 2007*, pp. 175–180, 2008.

[18] L. A. Guerrero, G. Maas, and W. Hogland, “Solid waste management challenges for cities in developing countries,” *Waste Manag.*, vol. 33, no. 1, pp. 220–232, 2013.

[19] S. . Kumar *et al.*, “Challenges and opportunities associated with waste management in India,” *R. Soc. Open Sci.*, vol. 4, no. 3, pp. 702–715, 2017.

[20] G. K. Shyam, S. S. Manvi, and P. Bharti, “Smart waste management using Internet-of-Things (IoT),” *Proc. 2017 2nd Int. Conf. Comput. Commun. Technol. ICCCT 2017*, pp. 199–203, 2017.

[21] S. K. Nambiar and S. M. Idicula, “A multi-agent vehicle routing system for garbage collection,” *2013 5th Int. Conf. Adv. Comput. ICoAC 2013*, pp. 72–76, 2014.

[22] M. Faccio, A. Persona, and G. Zanin, “Waste collection multi objective model with real time traceability data,” *Waste Manag.*, vol. 31, no. 12, pp. 2391–2405, 2011.

[23] A. Tripathy and S. K. Rath, “Supervised Machine Learning: A Review of Classification Techniques,” *Int. J. Rough Sets Data Anal.*, vol. 4, no. 1, pp. 56–74, 2017.

[24] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, “Machine learning for internet of things data analysis: a survey,” *Digit. Commun. Networks*, vol. 4, no. 3, pp. 161–175, 2018.

[25] S. Balaji, “Waterfall vs v-model vs agile : A comparative study on SDLC,” *WATEERFALL Vs V-MODEL Vs Agil. A Comp. STUDY SDLC*, vol. 2, no. 1, pp.

26–30, 2012.

[26] "Vue.js." [Online]. Available: https://vuejs.org/. [Accessed: 04-Jun-2018].

APPENDIX I.

Predictive Engine

```
# coding: utf-8

# In[1]:


from sklearn.metrics import accuracy_score, log_loss, precision_score, recall_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC, LinearSVC, NuSVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier
from sklearn.naive_bayes import GaussianNB
from xgboost import XGBClassifier
import xgboost

classifiers = [
    KNeighborsClassifier(3),
    SVC(kernel="rbf", C=0.025, probability=True),
    DecisionTreeClassifier(),
    RandomForestClassifier(),
    GaussianNB()]


# In[23]:


# Logging for Visual Comparison
log_cols=["Classifier", "Accuracy", "Precision", "Recall", "Log Loss"]
log = pd.DataFrame(columns=log_cols)

for clf in classifiers:
    clf.fit(X_train, y_train)
    name = clf.__class__.__name__

    print("="*30)
    print(name)

    print('****Results****')
    train_predictions = clf.predict(X_test)
    acc = accuracy_score(y_test, train_predictions)
    print("Accuracy: {:.4%}".format(acc))
    precision =  precision_score(y_test, train_predictions)
    print("Precision: {:.4%}".format(precision))
    recall =  recall_score(y_test, train_predictions)
```

```
    print("Recall: {:.4%}".format(recall))


    train_predictions = clf.predict_proba(X_test)
    ll = log_loss(y_test, train_predictions)
    print("Log Loss: {}".format(ll))

    log_entry = pd.DataFrame([[name, acc*100, precision*100, recall*100, ll]], columns=log_cols)
    log = log.append(log_entry)

print("="*30)


# In[24]:


log


# In[26]:


sns.set_color_codes("muted")
sns.barplot(x='Accuracy', y='Classifier', data=log, color="b")

plt.xlabel('Accuracy %')
plt.title('Classifier Accuracy')
plt.savefig('comparison-accuracy.png',bbox_inches='tight')
plt.show()

sns.set_color_codes("muted")
sns.barplot(x='Precision', y='Classifier', data=log, color="r")

plt.xlabel('Precision %')
plt.title('Classifier Precision')
plt.savefig('comparison-precision.png',bbox_inches='tight')
plt.show()

sns.set_color_codes("muted")
sns.barplot(x='Recall', y='Classifier', data=log, color="cyan")

plt.xlabel('Recall %')
plt.title('Classifier Recall')
plt.savefig('comparison-recall.png',bbox_inches='tight')
plt.show()

sns.set_color_codes("muted")
```

```
sns.barplot(x='Log Loss', y='Classifier', data=log, color="g")

plt.xlabel('Log Loss')
plt.title('Classifier Log Loss')
plt.savefig('comparison-logloss.png',bbox_inches='tight')
plt.show()
clf = DecisionTreeClassifier()

# Fit with all the training set
clf.fit(X, y)


# In[17]:


importances = clf.feature_importances_
indices = np.argsort(importances)[::-1]
feature_names = X.columns

print("Feature ranking:")
for f in range(X.shape[1]):
    print("%s : (%f)" % (feature_names[f] , importances[indices[f]]))


# In[18]:


f, ax = plt.subplots(figsize=(8, 8))
plt.title("Feature ranking", fontsize = 12)
plt.bar(range(X.shape[1]), importances[indices],
    color="b",
    align="center")
plt.xticks(range(X.shape[1]), feature_names)
plt.xlim([-1, X.shape[1]])
plt.ylabel("importance", fontsize = 18)
plt.xlabel("index of the feature", fontsize = 18)
plt.savefig('feature-ranking.png',bbox_inches='tight')
```

APPENDIX II.

Frontend – Sensor Interval Information Interface Code

```
<template>
<b-container class="bv-example-row">

    <div class="md-2">
        <p>Fill Level(%)</p>
        <b-form-input v-model="fill_level"
                type="text"
                placeholder="Fill level(%)"></b-form-input>
         <p>Temperature (C)</p>
         <b-form-input v-model="temperature"
                type="text"
                placeholder="Temperature(C)"></b-form-input>
         <p>Humidity(%)</p>
         <b-form-input v-model="humidity"
                type="text"
                placeholder="Humidity(%)"></b-form-input>
         <p>Frequency</p>
         <b-form-input v-model="frequency"
                type="text"
                placeholder="Frequency"></b-form-input>
         <p></p>

        <b-button @click="generateData" variant="primary" size="lg_2">Generate Data</b-button>
        <b-button @click="sendData" variant="success" size="lg_2">Send to Predictive Engine</b-
button>
    </div>

        <div class="md-2">
            <b-alert :show="dismissCountDown"
             dismissible
             variant="warning"
             @dismissed="dismissCountDown=0"
             @dismiss-count-down="countDownChanged">
      <p>Status: {{ status }}</p>
      <p>This alert will dismiss after {{dismissCountDown}} seconds...</p>
      <b-progress variant="warning"
                :max="dismissSecs"
                :value="dismissCountDown"
                height="4px">
      </b-progress>
    </b-alert>
    </div>

</b-container>
```

```
</template>

<script>

import axios from 'axios';

  export default {
 data () {
  return {
    fill_level: 0,
    temperature: 0,
    humidity: 0,
    frequency: 0,
    status: 1,
    dismissSecs: 5,
    setIntervalId: 0,
    dismissCountDown: 0,
    showDismissibleAlert: false
  }
 },
 mounted() {

  this.generateData()

 },
 beforeDestroy() {
  clearInterval(this.setIntervalId)
 },
 methods: {
  countDownChanged (dismissCountDown) {
    this.dismissCountDown = dismissCountDown
  },
  generateData: function() {
      this.setIntervalId = setInterval(function() {
      this.fill_level = Math.floor((Math.random() * 100) + 1);
      this.temperature= Math.floor((Math.random() * 40) + 15);
      this.humidity = Math.floor((Math.random() * 60) + 1);
      this.frequency = Math.floor((Math.random() * 30) + 1);
      this.sendData()
      }.bind(this), 5000);


  },
  sendData () {

    axios
  .get('http://192.168.229.145:5000/interval?fill_level='
```

```
        +this.fill_level+'&temperature='
        +this.temperature+'&humidity='
        +this.humidity+'&frequency='
        +this.frequency)
    .then(response => {
        this.status = response.data
        })
      this.dismissCountDown = this.dismissSecs
    }
  }
}
</script>

<style>

</style>
```

APPENDIX III

```html
<template>
<b-container class="bv-example-row">

   <div class="md-2">
       <p>Fill Level(%)</p>
       <b-form-input v-model="fill_level"
               type="text"
               placeholder="Fill level(%)"></b-form-input>
        <p>Temperature (C)</p>
        <b-form-input v-model="temperature"
               type="text"
               placeholder="Temperature(C)"></b-form-input>
        <p>Humidity(%)</p>
        <b-form-input v-model="humidity"
               type="text"
               placeholder="Humidity(%)"></b-form-input>
        <p>Weight</p>
        <b-form-input v-model="weight"
               type="text"
               placeholder="Weight"></b-form-input>
        <p></p>

       <b-button @click="generateData" variant="primary" size="lg_2">Generate Data</b-button>
       <b-button @click="sendData" variant="success" size="lg_2">Send to Predictive Engine</b-
button>
   </div>

       <div class="md-2">
           <b-alert :show="dismissCountDown"
            dismissible
            variant="warning"
            @dismissed="dismissCountDown=0"
            @dismiss-count-down="countDownChanged">
     <p>Status: {{ status }}</p>
     <p>This alert will dismiss after {{dismissCountDown}} seconds...</p>
     <b-progress variant="warning"
               :max="dismissSecs"
               :value="dismissCountDown"
               height="4px">
     </b-progress>
   </b-alert>
   </div>

</b-container>

</template>
```

```
<script>

import axios from 'axios';

   export default {
  data () {
    return {
      fill_level: 0,
      temperature: 0,
      humidity: 0,
      weight: 0,
      status: 1,
      dismissSecs: 5,
      setIntervalId: 0,
      dismissCountDown: 0,
      showDismissibleAlert: false
    }
  },
  mounted() {

    this.generateData()

  },
  beforeDestroy() {
    clearInterval(this.setIntervalId)
  },
  methods: {
    countDownChanged (dismissCountDown) {
      this.dismissCountDown = dismissCountDown
    },
    generateData: function() {
        this.setIntervalId = setInterval(function() {
        this.fill_level = Math.floor((Math.random() * 100) + 1);
        this.temperature= Math.floor((Math.random() * 40) + 15);
        this.humidity = Math.floor((Math.random() * 60) + 1);
        this.weight = Math.floor((Math.random() * 200) + 1);
        this.sendData()
        }.bind(this), 5000);


    },
    sendData () {

      axios
    .get('http://192.168.229.145:5000/waste?fill_level='
        +this.fill_level+'&temperature='
```

```
        +this.temperature+'&humidity='
        +this.humidity+'&weight='
        +this.weight)
    .then(response => {
        this.status = response.data
        })
      this.dismissCountDown = this.dismissSecs
    }
  }
}
</script>

<style>

</style>
```