Smart Prediction and Monitoring of Waste Disposal System Using IoT and Cloud for IoT Based Smart Cities

Jacob John, et al. *[full author details at the end of the article]*

Accepted: 8 August 2021

Abstract

One of the prominent applications of Internet of Things (IoT) in this digital era is the development of smart cities. In IoT based smart cities, the smart objects (devices) are connected with each other via internet as a backbone. The sensed data by the smart objects are transmitted to the sink for further processing using multi hop communication. The smart cities use the analyzed data to improve their infrastructure, public utilities and they enhance their services by using the IoT technology for the betterment of livelihood of the common people. For IoT based smart cities, waste collection is a prominent issue for municipalities that aim to achieve a clean environment. With a boom in population in urban areas, an increasing amount of waste is generated. A major issue of waste management system is the poor process used in waste collection and segregation. Public bins begin to overflow for a long period before the process of cleaning starts, which is resulting in an accumulation of bacteria causing bad odors and spreading of diseases. In order to overcome this issue, in this paper an IoT based smart predication and monitoring of waste disposal system is proposed which utilizes off-the-shelf components that can be mounted to a bin of any size and measure fill levels. An Arduino microcontroller is employed in the proposed model to interface the infrared (IR), ultraviolet (UV), weight sensors, and a Global Positioning System (GPS) module is used to monitor the status of bins at predetermined intervals. The proposed system transmits the data using the cluster network to the master module which is connected to the backend via Wi-Fi. As data is collected, an intelligent neural network algorithm namely Long Short-Term Memory (LSTM) is used which will intelligently learn and predict the upcoming wastage from waste generation patterns. Moreover, the proposed system uses Firebase Cloud Messaging to notify the appropriate people when the bins were full and needed to be emptied. The Firebase Cloud Messaging (FCM) JavaScript Application Programming Interface (API) is used to send notification messages in web apps in browsers that provide service work support. Hence, the proposed system is useful to the society by providing facilities to the governments for enforcing stricter regulations for waste disposal. Additional features such as automated calibration of bin height, a dynamic web data dashboard as well as collation of data into a distributed real-time firebase database are also provided in the proposed system.

1 Introduction

As per reports by the United Nations, the world population by 2050 is expected to be 9.7 billion. By 2027, India is projected to surpass China as the world's most populated country. As population increases, so does consumption as hence waste Consumption patterns vary over the years, and the economic growth rate has been rapid. India generates roughly 62 million tons of municipal solid waste each year. Out of the 62 million, 30% is treated, and 70% is collected. Around 50% is dumped in landfills [1]. Per capita generation in our country varies from 170 to 620 g per person. This variability is mostly due to their location of whether urban or rural areas. In 2014, 51% of the municipal solid waste was organic, 32% is non-organic, and 17% is recyclable waste. E-waste of 1.975 million was generated in India in 2016.

Solid waste accumulation can cause pollution and ill-effects to health. Organic domestic waste can ferment and become a habitable place for microbes to grow. Exposure to hazardous waste causes diseases. This is a significant concern for the workers working with waste. Waste from industries is said to cause serious health risks and also cause water body contamination. Proper disposal of the hospital and medical waste is especially required. Landfills contain a variety of waste. This will hinder the decomposition of the biodegradable waste. Waste segregation is essential as it will enable the treatment of waste as per the kind of waste it is. Surat, Gujarat was adversely affected by a plague in 1994 due to the city's inefficiency to practice proper solid waste management.

A nationwide effort was made in 2014 with the introduction of Swachh Bharat mission, a five-year-long program. Before this, several cities had started their own initiatives. Surat took up various practices like involving citizens, door to door collection, hotel-kitchen waste management, etc. after the plague. Solid Waste Management Rules were implemented in 2016. These were regulations on segregation being mandatory, the inclusion of waste pickers in the municipalities' waste management process, a fee charged for bulk generators of waste, etc. A web-based application was launched in 2016 called as the Ministry of Environment, Forest and Climate Change (MoEFCC) initiative. This application is able to track and monitor the waste management in India [1] effectively. Moreover in Japan, a small town called Kamikatsu has introduced and initiated the implementation of a new and effective zero-waste policy from the year 2003 which aims at eradicating the waste by the year 2020 and achieved it upto 81% recycling rate without turning to landfills [2].

Pollution is also an environmental issue affecting huge numbers all over the world. Major forms of pollution are air, light, noise, plastic, soil, thermal water, etc. In India alone, air pollution is a severe concern causing the early deaths of 2 million Indians every year. Meanwhile, in China, air pollution has been the cause of 350 to 400 thousand people a year.

A review of numerous papers has been done, and there are IoT based smart solutions existing from the literature [3–7]. However, most of the existing systems lacks in some areas such as the range of the devices, accuracy in value due to the hindrance of weather and other factors, etc. Population increase and urbanization have led to drastic changes over the years with respect to the economy, industrialization, lifestyles, etc. Waste management is one of the numerous issues that have come up that need to be attended to at the earliest. Hazardous waste causes pollution and has health impacts in the long run. Waste management has many factors. There are numerous kinds of hazardous and non-hazardous waste. Bins overflow at times, and, on the other hand, bins are sometimes

emptied when they are not completely filled. This issue can be tackled by changing the static routes and schedules to be more optimal.

While options like composting and treatment of waste in plants are available, the country also needs standards and regulations to be implemented for efficient waste management. Consumption of paper and plastic, its recycling future growth prediction has been described in this paper [8]. Current waste disposal practices in India have been elucidated, and the future land requirement for waste disposal was projected. The paper also shows the growth of global warming and air pollution. Waste segregation can be tackled at the municipality level. Aerobic composting can be more implemented. Anaerobic digestion is said to have a great future in efficient waste management.

From the literature [9], the various authors have proposed the solution for the waste management in smart cities. The goal intended was to find the strengths weaknesses and develop an overall best solution. Most of the existing approaches were lack in accuracy in the information collected by the systems. This lack accuracy is results in change in weather conditions. Users were also susceptible to unauthorized access. There was a limitation of range as the devices used in the systems and these devices are capable of only short-range communication. Motivated from these observations, in this paper a novel Smart Prediction and Monitoring of Waste Disposal system using IoT and Cloud for IoT based Smart Cities has been proposed in this paper. The major contributions of this paper are:

1) An IoT based smart predication and monitoring system has been proposed in this paper for effective waste disposal which uses off-the-shelf components that can be mounted to a bin of any size and with measure fill levels.
2) The proposed system can transmit the data using the cluster network to master module which is connected to the backend via Wi-Fi. Moreover, we use an Arduino microcontroller to interface the IR, UV, weight sensors and a GPS module is used to monitor the status of bins at predetermined intervals.
3) LSTM is used in this work since it is able to learn and predict the upcoming wastage from waste generation patterns more intelligently.
4) The proposed system utilizes Firebase Cloud Messaging (FCM) to notify the appropriate people when the bins were full and needed to be emptied. Additionally, The FCM JavaScript API has been used to send notification messages in web apps in browsers that provide service work support.
5) Additional features such as automated calibration of bin height, a dynamic web data dashboard as well as collation of data into a distributed real-time firebase database are also provided in the proposed system.
6) Finally, two new algorithms namely Sensors (1) 'Data Collection Algorithm Running on the Arduino' and (2) 'Algorithm for Firebase integration running on the Arduino in collaboration with the ESP8266' have been proposed and implemented in this work.

The rest of this paper is organized as follows: Sect. 2 provides a survey of related work in the areas of IoT and sensor based waste management systems and classification using deep learning techniques. Section 3 describes the architecture, experimental set up and the algorithms proposed in this work for developing the smart and intelligent waste management system. Section 4 explains the implementation details of this work, results obtained from this work and the suitable discussions made about the results with comparative analysis. Section 5 gives the conclusions on this work and then it suggests some future works that can be carried out by extending the proposed system.

2 Related Works

In the past, many researchers have proposed different approaches for designing IoT based waste management systems for making smart cities. Among them, Recycle.io [10] is an IoT enabled framework that utilizes an array of sensors and edge computing to detect real-time waste disposal violations and thus, enabling municipalities to enforce stricter regulations on the disposal of waste. Furthermore, this framework also aims to alleviate the problem of expensive and time-consuming segregation of waste. Recycle.io consists of smart recycling bins (SRB) and Smart Organic Bins (SOB), each attached with a Raspberry Pi, a camera, an ultrasonic sensor, and an infrared sensor. The cameras capture images of materials that are disposed of in each of the respective bins. Using the Raspberry Pi as an edge device, the image is processed to determine whether there are any violations at the source site. For example, an organic item being disposed in a Smart Recycle Bin (SRB). The framework is integrated with Microsoft Azure's Custom Vision Service. This is an Artificial Intelligence tool that is assimilated from a set of labeled images and creates a machine learning model. A classifier unit trained on Compact Fluorescent Lamp (CFL), eggshells, cardboard, and Styrofoam, identifies the object in the image. Furthermore, the Custom Vision Service presents the added benefit of incorporating active learning, i.e., the ability to allow a feedback loop to help improve the classifier. In addition to this, a server less architecture, along with Azure IoT Hub, provides a higher level of scalability. Azure Functions are used to store and replicated data into an Structured Query Language (SQL) database. While blob storage is used to store violations, Azure functions are also used for data aggregation and summarization from multiple bins. This summarized data is used to form a real-time data dashboard for monitoring these smart bins.

Due to a server less aspect, no infrastructure needs to be maintained by the developers. Furthermore, costs are reduced as functions are only billed on a "pay-as-you-go" basis. The dynamic provision also eradicates the need to purchase additional computing power or to partition a cloud server for an existing instance. However, since all products are being powered by Microsoft services, Recycle.io runs at risk of suffering from vendor lock-in. In addition to this, a visible camera on smart bins could introduce a privacy issue among some members of the community. It is causing them to either be more self-conscious, uncomfortable, or flustered at the idea of having their waste disposals being monitored. The SRBs and SOBs could also benefit from solar charging as a primary power source and use batteries as an auxiliary power source and thereby making them more environmentally friendly. In the past, Navghane et al. [11] proposed an IoT Based smart garbage and waste collection bin to tackle the problem of overloaded public waste bins. Such situations create a breeding ground for rats and insects while also attracting stray dogs and cats. Furthermore, they also create a foul smell when the waste lies unattained for prolonged periods of time. In addition to this, this IoT implementation aims to reduce human intervention (checking bins on a routine basis) while also eliminating the need for a middle-man: the contractor. The message is sent directly to the cleaning vehicle.

The smart dustbin is equipped with an LPC2148 ARM microcontroller. This microcontroller is interfaced with a TSOP 1738 IR Sensor, a weight sensor, and a Wi-Fi module that supports Wi-Fi Direct (P2P) and the standard 802.11b/g/n protocol. The weight sensor works using the principle of piezo-resistivity and is also RoHS compliant. It is used to detect the weight of the bin to check if a given threshold is reached. Similarly, the IR sensor is also used to monitor the bin against a threshold to check if the garbage level exceeds a set amount. Three IR sensors form a level detector that is used to check the level

of waste in the bin at three levels—0%, 50%, and 90%. When both the level detector and weight sensor are activated, the micro-controller transmits the details via a Wi-Fi module. At the receiver's end, an HTML web page is used to present the status of the dustbin to the agency. Ideally, a mobile device with a web browser is used to access the web page.

Along with the reduce need for human intervention and help a municipality attain Smart IoT characteristics, the dustbins assist garbage collection agencies in keeping the city clean. If the data and the Hyper Text Markup Language (HTML) pages are made publicly available, it will urge the agencies to take action. Hereby, improving reliability and productivity as the monitoring would be automated. Similar to the majority of the implementations, there are no mechanisms for zero fault tolerance nor redundancy. Furthermore, if one of the three sensors fail, the entire implementation is likely to act abnormally. In addition to this, the system also lacks scalability, for every new dustbin added, a new web page has to be created and monitored separately. The bins also require a constant Wi-Fi connection, which can be a challenge in rural areas. A distributed peer-to-peer network would work better in this case.

Baby et al. [12] presented a smart bin that uses an intelligent waste alert system to notify the respected authorities in the event of a filled-up dustbin. Furthermore, this implementation is coupled with a prediction system that utilizes machine learning to forecast waste production in a particular region. This implementation focuses on reducing the time consuming and expensive process of physically checking each waste bin individually. Furthermore, it eliminates for garbage collectors to take trips to every waste bin as they only visit bins when they're "critically filled." However, a future enhancement could be the use of maps API along with the shortest path algorithm to provide garbage collectors an optimum path to take for visiting full waste bins.

This system monitors waste using infrared and ultrasonic sensors that are interfaced with a Raspberry Pi microcontroller. When a desired level of waste is reached, the Raspberry Pi notifies authorities about the waste bin via Short Message Service (SMS) alerts and e-mails. This is done using the Simple Mail Transfer Protocol (SMTP) service to send the mail. An If This Then That (IFTTT) web service is used to send mobile and Google calendar alerts. This is an efficient and novel approach with the advent of mobile computing. Authorities are more likely to respond to text messages, emails, and calendar alerts. However, emails and SMSs tend to be ignored or missed when sent continuously large amounts. This happens when several bins fill up around the same time. A solution to this is to aggregate the data and send a single email/SMS with all the bins that require attention.

An Arduino module, on the other hand, is used in conjunction with the Raspberry Pi to monitor the level of trash. The Arduino uses an Ethernet shield to send data to the cloud. This data is then analyzed using Microsoft Azure and Power BI. Linear regression is utilized to predict future amounts of waste. However, a neural network would be more suited for such a prediction due to its improved accuracy and reduced bias-variance tradeoff. Furthermore, employing two microcontrollers is unnecessary and increases the costs of the implementation. Thereon, an Arduino alone would be sufficient for monitoring and sending data to the cloud.

Taimur et al. [13] used a novel approach that combines the existing sensory methods along with back-end data analytics for efficient waste collection. Using off-the-shelf components such as the Raspberry Pi and an HC-SR04 standard ultrasonic range sensor, the depth of the garbage bins is monitored. This data is transmitted over the Internet using the Raspberry Pi's inbuilt WIFI capabilities. Furthermore, the monitored statistics are encrypted in the triple-DES format prior to transmission. This helps protect the data when the information is compromised. It should be noted that additional work is needed

in order to explore the consequences of cyber-attack on waste collection data. In order to increase power efficiency, the Wi-Fi module is only activated when data is required to be sent. However, future work to increase power efficiency could be to use solar energy and a low powered monitoring mode. This, coupled with transmitted data as pre specified intervals, would again reduce the overall power usage. Furthermore, a distributed peer-to-peer network or a cluster network can be used rather than relying on each device connecting to a Wi-Fi router. This implementation also used Python and Google maps API to map out real-time routes using bin coordinates. An android based mobile platform is used as a front end to present a final route to drivers. This system also dynamically updates the final route is updated on the basis of a collection list, an auxiliary list, and fuel economy. A 46% increase in fuel efficiency, along with an 18% reduction in collection time, was observed over a ten-day trial period.

Mirchandani et al. [14] have proposed Radio-frequency identification (RFID) tags to track the wastes based on an online web-based system. It also checks for the fullness of the waste-bins and thus updates the status of each dustbin through the municipal servers. The municipal servers are notified when the dustbins are full and would subsequently calculate the shortest route for the trucks to take to reach the dustbin. Through this, the current waste collection system would be optimized, thus nullifying the hazards of waste accumulation. It would also see that the spillage of the trash is checked, thus ensuring that the bins don't overflow and produce foul odors and spread bacteria. These RFID's can locate the bins and collect information about the amount of trash accumulated.

This information is sent through two sensors, the level sensor, and the toxicity sensor. This information is sent to a decision-making system, which would then determine whether the trash from the bin must be collected. These sensors would also indicate the level of trash in the bins to notify the user about the current status of the bins so that the user would through trash in it accordingly. The user will just need to scan their RFIDs with the bins to get this information from the sensors. This is a very innovative approach for proper and efficient waste management as it not only optimizes the collection of the waste but also it helps in reducing the concomitant cost involved in its management.

Mukherjee et al. [15] proposed a 'Smart Bin' which would harness the energy, and the rest is integrated into the electrical power grid. Thus these 'Smart Bins' are converted into Green energy sources. This has been feasible due to the technological achievements in the field of servers and sensors like PIR, Ultrasonic Sensor, etc. However, using these sensors is not advisable as they require a lot of power, and thus, equipping the bins with solar panels and piezo crystals are much more electricity efficient. The solar panels will gather energy during the daytime and power the crystals which harness the energy of the people walking by. Thus, setting up a network of 'Smart Bins' harnesses massive amounts of energy and it is integrated into the electrical power grid.

Poddar et al. [16] also proposed Smart Bins that can provide a cleaner and better collection and management of the bins. To make such a bin, passive infrared (PIR) sensors, Ultrasound sensors, temperature sensors, and proximity sensors are used. Also, in addition to these, components like Liquid Crystal Display (LCD), Real-Time Clock, Servo Motor have been used. Complete automation of the system is done with the help of the Arduino Uno board, which acts as the Central Processing Unit (CPU) of the system, controlling and coordinating all of the components of the system. Internet connectivity is provided with the help of a small device called the Ethernet Shield/Wi-Fi shield. The Lid of the Bin closes after a certain threshold of fill level, say 90%, has been achieved, and the lid won't open. This would ensure that the bins don't overflow and would thus maintain the cleanliness of the environment.

Shyam et al. [17] proposed a system of sensors that work in a collaborative fashion. The main agenda of their system is that the waste levels would be measured using the sensors; the data obtained will be sent through the Internet to a server for storage and processing mechanisms. Through this, live monitoring of the waste bins is done, and thus the selection of the bins becomes easier. Also, the routes are being calculated and sent to the workers so that they received the best route each and every day. Once a starting point has been reached, the data is analyzed to get the rate at which each bin gets filled so as to predict in advance which bins to collect from which place. To conclude, this is an intelligent waste collection system as it is able to predict which bins are to be collected based on the data analysis of the history of information sent by the sensors of the grid of Smart Bins.

The main smart bin in the proposed method by Saha et al. [18] is solar-powered and consists of sensors to detect fill levels and transmit the data to a cloud server. Users have access to and are notified about information regarding the fill level of the bins and are even suggested optimal routes for waste collection. This smart solution can be applied to various kinds of containers and overall helps in reducing fuel consumption, time, and use of trucks. A low-cost smart bin is proposed by Devi et al. [19]. It consists of a Node MCU wi-fi module whose chip collects data from sensors to the cloud. An ultrasonic sensor is used to check the waste level of the bin. This proposal also has environmental monitoring, which can be enabled with the use of several extra sensors. Quality, temperature, and heat can be monitored this way. The implementation of this system was successful. The status view showed the percentage of how full the bins were and also gave the value of surrounding environmental factors. When the bins are full, respective authorities are notified via email, SMS, call, etc., till the bins are emptied. The low cost makes it affordable to utilizing on a large scale, and it has more uses than efficient waste management.

Kumar et al. [20] developed a new garbage monitoring and an effective clearance system using IoT. The paper proposes a method of detecting garbage levels by attaching sensors to the dustbins. The dustbin levels are monitored by these sensors, and once the dustbin gets full or the weight of waste exceeds the threshold value assigned, the sensors send out a message 'dustbin filled' with the help of a Global System for Mobile Communications (GSM) module. Hence the authority in this case, a garbage truck driver is notified about the overflowing dustbin so that he/she can take effective action for the garbage disposal. Two LED lights are attached to the dustbin—one green, to show that the dustbin is still not completely filled, and one red, to display the condition of an overflowing dustbin. Moreover, the GSM module sends the message to an android application. With such notification, the latitude and longitude of the dustbin location are also mentioned, which the truck driver can click to open the maps application and drive to the destination. Other information like weight, volume, date, and time are mentioned, and all this is available on a database server too. The pros of this system are that it has given an effective method of garbage clearance, hence ensuring environment cleanliness, the information on the database servers provide a free access to the municipal authorities too, about the condition of dustbins in their locality, and there is also a feature in their android application to help a user find the nearest dustbin. The areas where the paper could improve on are it is a prototype and has not been implemented at a product level, so there is no surety if it is feasible in the real world, the implementation can be made durable by making it cheaper and compact, and there is no technology used to ensure that the wrong type of waste is being thrown in a dustbin meant for only a particular type of waste.

Keerthana et al. [21] proposed a solution to monitoring garbage levels and its disposal with a minimal number of trips made to dustbins, thereby bringing down the overall expenditure. It is implemented by using a data-sharing model by a cloud connection

between trash cans, truck drivers, and corporations. Two threshold levels are set, which will send out a message of trash overflowing. The trash cans lock up when the maximum levels are reached. For garbage and level detection, the IR sensor, fill level sensor, and weight sensors are attached to the dustbins. GSM module is used for communication among authorities. The pros are a locking system on dustbins to ensure more waste is not thrown, and two threshold levels are fixed. The con is that there is no detection for type of waste.

Kolhatkar et al. [22] introduced a new concept of a moving dustbin. The dustbin has ultrasonic sensors to measure its levels and proximity sensors to detect obstacles in front of it. A person can stop the dustbin by placing their hand near it. The proximity sensor senses it, and the dustbin stops for the user to dispose of waste. The dustbin follows a path in a locality depending on the path preloaded or on the concept of obstacle detector robot or line follower. All these are connected to an Arduino board. A wi-fi module is also connected to transmit data of dustbin levels to a server. It sends an email to concerned authorities for the dustbin's clearance. The LCD display gives status of the dustbin. Pros are that food wastage is reduced by 33%, the dustbin moves through a neighborhood, and efficient waste disposal is ensured. Cons are, this implementation requires battery power, waste disposal is not automatically taken care of and cannot distinguish between types of waste- whether dry or wet garbage.

Kumar et al. [23] proposed a novel method to monitor the garbage cleaning process from a dustbin remotely. Firstly, ultrasonic sensors connected to Arduino sense the fill level of dustbins. Then, the authorities are alerted about the instant cleaning of dustbin through email. The truck driver empties the dustbin. Each dustbin has an RFID tag attached, which provides an automatic update to servers that the cleaning process has been done efficiently. Notifications are sent using the wi-fi module connected to the Arduino. The pros of this system are that the implementation helps efficient smart waste disposal by monitoring an often-neglected part of the whole procedure, the garbage cleaning part by the truck driver. The con is it cannot sense the type of waste disposed of in the dustbin.

Hong et al. [24] proposed a battery-based smart garbage bin (SGB) system in order to reduce the amount of food waste through efficient management of waste. This system resulted in a 33% reduction in food waste.

The SGB system presents the added benefit of being user-friendly. SGBs incorporate an RFID system to bill customers. When a customer scans their RFID card against the bin and disposes of their waste, the balance of the RFID card is displayed on an LCD screen using previous waste data. Furthermore, rather than billing customers in real-time for each disposal, the transaction takes place when no other residents are waiting to use the SGB. The payment data is sent in an encrypted manner to a router (upon receiving a request message from the router) followed by an automated credit card deduction happening in the back end, thereby, mitigating delays produced during real-time transactions.

This system also employs a wireless mesh network (WMN), routers, and Header Smart Garbage bins (HSGBs). The WMN allows for service continuity and secure communication reliability. This, coupled with routers, allows for service provisioning. Furthermore, the HSGB is responsible for collecting data, analyzing, and managing other SGBs within its region. If the HSGB fails, it passes on authority to another SGB within the region and thus allowing for fault tolerance. The use of an energy-efficient communication technique presented in the paper allows for power saving. However, despite this implementation, battery life is still poor. Furthermore, due to the monolithic and complicated architecture, maintenance costs are high. Routers, LCD screens, RFID card systems, and databases also incur significant costs despite the system's efficiency. In addition to this, a renewable

source of energy, such as solar power, can be used. Improper segregation and disposal of waste can have detrimental effects in the long run.

Kavya et al. [25] aimed at tackling this issue by segregating waste into three types—dry, metal, and wet—before proper and efficient disposal of the waste. Segregation is done by placing the waste onto a conveyor belt and utilizing sensors to identify the type of waste. An inductive sensor detects metal, air blower helps identify light articles, and the moisture sensor identifies the wet waste. The metal waste is demagnetized before disposal. Accordingly, it is separated into different bins. Wireless sensors continuously monitor the bin to know its fill level. When the bin is full, the trash management department is notified, and an optimized collection schedule can be made manually by the collection team of the department.

Jain et al. [26] also proposed a smart dustbin for efficient waste management. While the fill level is checked using an ultrasonic sensor, a force sensor has been used additionally to check the weight of the bins. When a specific value is crossed, the location of the bin is sent to the server, which communicates to the android mobile the area of the location of the bin. The common usage of smartphones aids in this being an accessible system. Many other recent works on IoT based intelligent systems are also available in the literature [27–29]. However, the use of deep learning in waste management is not explored well in the literature. Therefore, a new smart and intelligent waste management system that uses IoT based sensors for data collection and a deep learning based prediction model is proposed in this paper for implementing an effective waste management system.

3 Proposed System

The main aim of the proposed work is to measure the fill level of a smart dustbin using fill-level sensors, humidity and temperature using the respective sensors and with the help of the data generated from this measurement, propose a time series that would aid in the collection of the garbage from the smart dustbin. The data, consisting of the parameters mentioned above, is collected for three days and then sorted chronologically in the form of a time-series graph. This time data is collected in the form of discrete data points that are regularly placed. This data is backed up into a cloud database. To perform a time series prediction, Long Short Term Memory is used. To supplement this, the humidity and temperature sensors are able to measure their respective values, which above a certain threshold, closes the lid of the dustbin until it is emptied by an appropriate authority.

3.1 Long Short-Term Memory

Long short-term memory (LSTM) [30], as proposed by Hochreiter and Schmidhuber [31], is a type of Recurrent Neural Network (RNN) [32]. RNN is an artificial neural network architecture in deep learning. RNNs maintain states of variables at each memory block and used a combination of recurrent connections and standard inputs at every block in order to calculate the output activation. Three types of gates—forget gate layer, input gate layer, and output gate layer are used in order to change state and perform functions on the information flowing through the network. LSTM uses feedback connections, which makes use of entire sequences of data as compared to the use of single data points made by the standard feedforward neural networks. The significant difference of RNN with the help of LSTM from the classic or vanilla RNN [33] is in the nullification of the vanishing of the gradients

or explosion (conversion to infinity), which are back-propagated during the training process due to the computation involving finite-precision numbers. The general formula of the RNN is given in Eq. (1) as follows:

$$hidden^{(t)} = f\left(hidden^{(t-1)}, current\_input^{(t)}; \theta\right) \tag{1}$$

Equation (1) shows that the current hidden state hidden (t) is a function f of the previous hidden state hidden(t − 1) and the current input current_input (t). θ is another parameter of the function f. The $hidden^{(t-1)}$ layer of the RNN constitutes of a lossy summary of the past sequences until layer t. For LSTM equations, variables are given below.

$n_t \in R^d$: input vector

$f_t \in R^h$: forget gates activation vector

$u_t \in R^h$: input/update gates activation vector

$p_t \in R^h$: output gates activation vector

$s_t \in R^h$: hidden state vector/output vector

$e_t \in R^h$: cell state vector

$W \in R^{h \times d}, U \in R^h$: weight matrices

$b \in R^h$: bias matrices

The subscripts d and h represent the number of input and hidden units, respectively. The activation functions are given below.

σ_g : sigmoid function

σ_c : hyperbolic tangent function

σ_h : $\sigma_h(n) = n$

The primary functions are given in 2–6.

$$f_t = \sigma_g\left(W_f n_t + U_f s_{t-1} + b_f\right) \tag{2}$$

$$u_t = \sigma_g\left(W_i n_t + U_i s_{t-1} + b_i\right) \tag{3}$$

$$p_t = \sigma_g\left(W_o n_t + U_o s_{t-1} + b_o\right) \tag{4}$$

$$e_t = f_t \cdot e_{t-1} + u_t \circ \sigma_c\left(W_c n_t + U_c s_{t-1} + b_c\right) \tag{5}$$

$$s_t = p_t \cdot \sigma_h\left(e_t\right) \tag{6}$$

This implementation utilizes a Long Short-Term Memory (LSTM) network in order to perform the time series predictive [34].

Figure 1 shows the overall structure of the proposed system for smart waste management. It consists of the user interface module for client interaction, hardware module consisting of the Aurduino and sensors, Hardware and software interface module, LSTM training module, Rule formation module, database, Decision making and forecasting module, rule base and the testing module. The interaction among the modules as shown in the

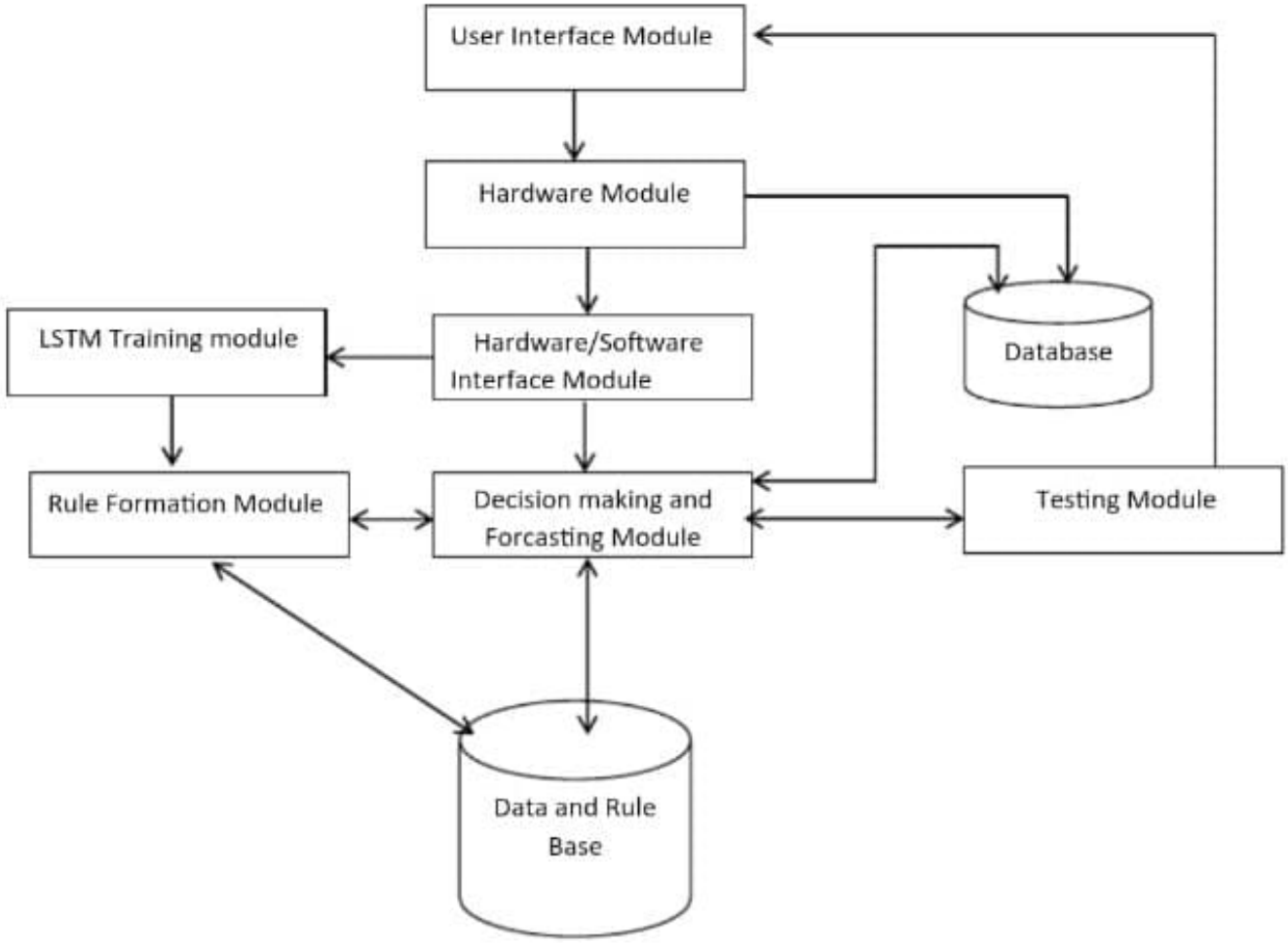


Fig. 1 Overall structure of the proposed system

diagram are responsible for data collection, classification and forecasting that are used in the filling process.

The hardware layer of the IoT enabled dustbin is represented by the block diagram in Fig. 2. The Arduino microcontroller is the central component that runs the entire system of reading data from the environment. It lays inside the dustbin and collects data through a series of sensors while communicating information to the user through actuators like LED flashes. One ultrasonic sensor connected to the digital pins 5 and 6 is measuring the ambient distance, which is the distance at which personnel may be standing near the dustbin. The other ultrasonic sensor, attached to the inside of the dustbin lid, and connected to pins 8 and 9, measures the fill level of the trash getting collected in the dustbin. It is crucial to keep a check that the waste does not spill out of the dustbin. Through the Arduino programming, the data sensed by the fill-level ultrasonic sensor is reflected by the LED indication of 5 LEDs connected from 0 to 4 digital pins. One LED separately connected to digital pin 10 serves the simple purpose of displaying the status on or off of the smart dustbin.

Ultrasonic sensor HC-SR04 has been utilized for two features of the system. The first purpose of this sensor is to measure the distance between itself and a target object. It uses the non-contact technology of waves. The sensor can be used for a range of 2 cm to 400 cm. In this implementation, one of these sensors is used as a fill-level sensor, while another one has been used as an ambient distance detector.

The second is the fill-level sensor placed under the lid of the bin so it can correctly measure the height up to which the bin has been filled. The ambient distance sensor has been placed on the front side of the bin. It detects if there is an object(hand) at a near distance to the bin and, if yes, opens the lid so that the trash can be deposited in it. The Accelerometer ADXL-335 usede in this work can detect the toppling of the dustbin depending

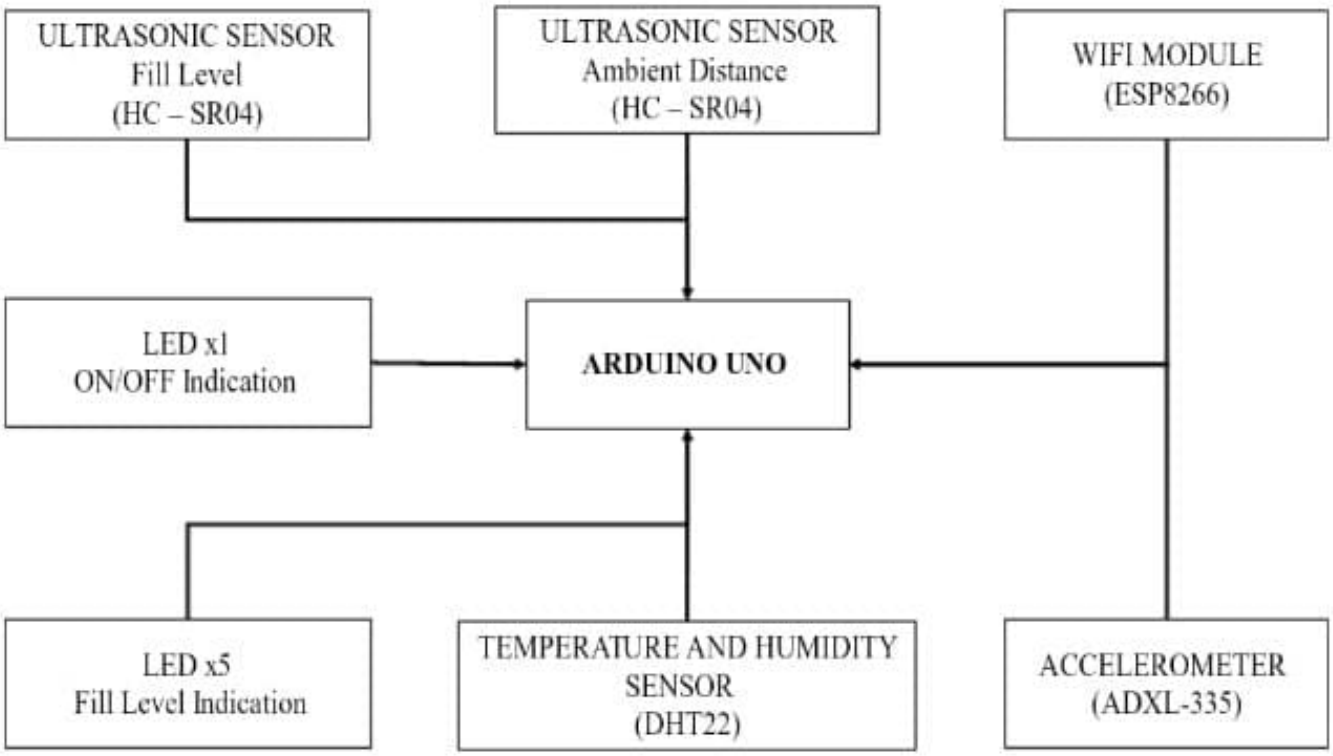


Fig. 2 Block diagram containing the Arduino based hardware module

on the capacitance changes sensed by the three-axis model. ADXL connects to the Arduino through analog pins 1 to 3. DHT22 senses the temperature and humidity values inside the bin and is connected to pin 11. Hence, it is necessary to prevent flames and the production of harmful gases like carbon monoxide. The accelerometer is used for detecting the stability of the bin. The algorithm (reference to the algorithm) shows the conditions used for checking the movement along the x-, y- and z-axis. In case the bin topples, it will close the bin and prevent the waste from falling out. After this, it ends the program.

The ESP8266 Wifi module sends over the data via TCP/IP protocol to the website built over firebase and JSON. The website displays the information collected over real-time in a user-friendly design implemented by bootstrap, among other things. ESP8266 is easy to interface with the cloud-based backend for transferring sensor data over the internet for quick visualizations.

The WiFi module is being used to transmit data from a cluster network in the system to the master module in the backend. It is used to send semi-structured JSON data to a firebase module. It also uses HTTPS and a secure API key to send data. Two advantages of this module are that it is low cost and standalone. At times, with the accumulation of waste in a particular area, there is the possibility of foul smell or even fire. The temperature and humidity sensor can help prevent this by monitoring the temperature and humidity of the environment of the trash. The sensor used for this purpose in this system is DHT22, which has a temperature measuring range of −40 °C to + 125 °C and humidity measuring range of 0% to 100%.

3.2 Proposed System Architecture

The smart dustbin system is built on a four-layered architecture, as displayed in Fig. 3. The main layers being the Database, Client Website, Hardware, and the Software layer. Firebase, a cloud-based database, especially for implementing IoT technology, has been used. Firebase is preferred as there is no need to establish middleware between

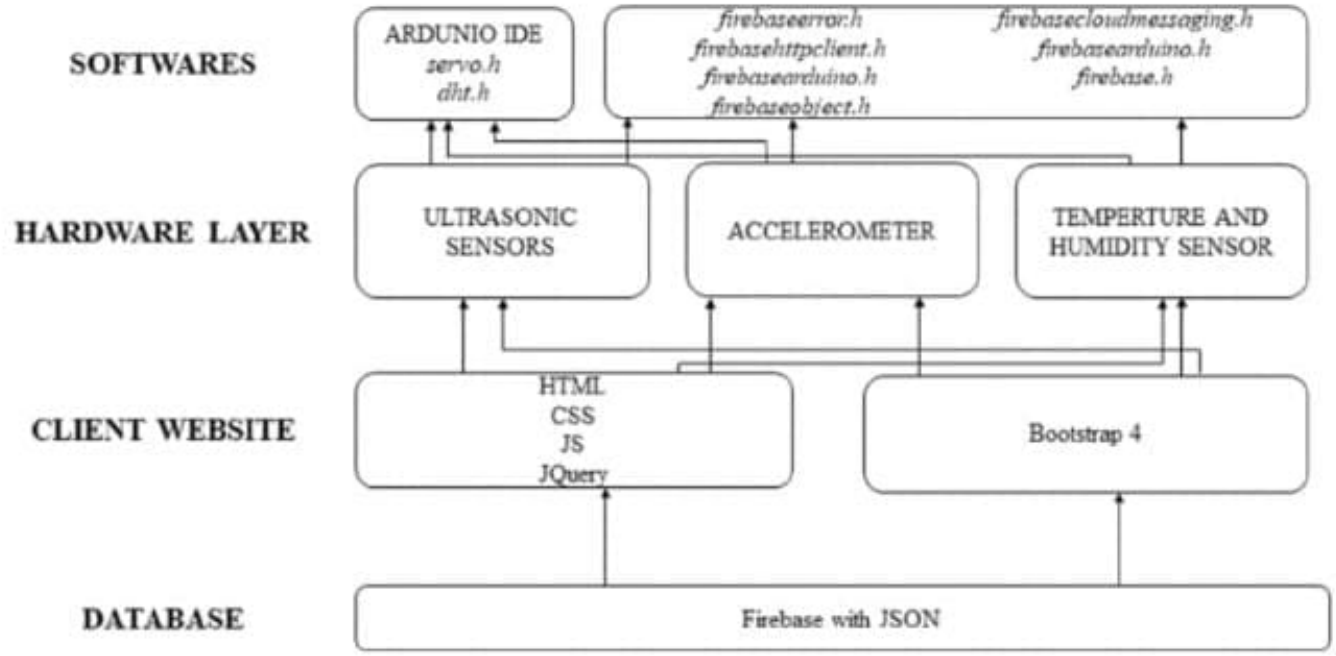


Fig. 3 Proposed system architecture

the website and service, and the data query code can be done in the client app. Firebase also uses JSON objects as data structures for the database, which is useful in real-time databases.

The client-side website is developed using HTML, CSS, Javascript, and Bootstrap to create interactive timeline graphs and pie charts for neat data visualizations. The website displays real-time readings of temperature and humidity inside the bin, toppled status of the bin, and the level of dustbin filled with waste. Through this data on the website, workers can be signaled when to clear the bins. The hardware layer of the bin comprises the intricate connections of the Arduino and the multiple sensors. The components used are two ultrasonic sensors, six LEDs, DHT22, ADXL335, and the Wifi module ESP8266. Some are sensors used to collect information about the environment, and the LED act as actuators to display important information like fill- level indications.

The final layer to integrate the last three layers is the software layer. One part of it is the Arduino IDE to activate the functionalities of the microcontroller and the sensors. Libraries Servo.h and DHT.h support the working of servo motor and DHT22. Servo motor lifts up and down the lid of the bin. Other libraries like firebase.h, firebaseCloudMessaging.h, and FirebaseHttpClient.h are necessary to transfer data over HTTP protocol and cloud server. Firebase libraries help in the background working of data retrieval from sensors and storage in the cloud database designed behind the client website.

The smart dustbin has both hardware and software aspects in terms of implementation. An Arduino Uno board served as the microcontroller for the various sensors comprising the smart dustbin. Digital pins of the Arduino board are reserved for the ambient ultrasonic sensor, fill level ultrasonic sensor, DHT22 sensor, and the 6 LEDs. 5 LEDs are used as an LED bar graph to signal the fill level status. Sixth LED shows the status of the smart dustbin sensors being on or off. Three analog pins of the Arduino are needed for the ADXL335 3-axis accelerometer, which is fixed on a steady base. A power of 5 V is supplied to the sensors using a breadboard connection, along with the grounding.

3.3 Hardware Algorithms of the Proposed System

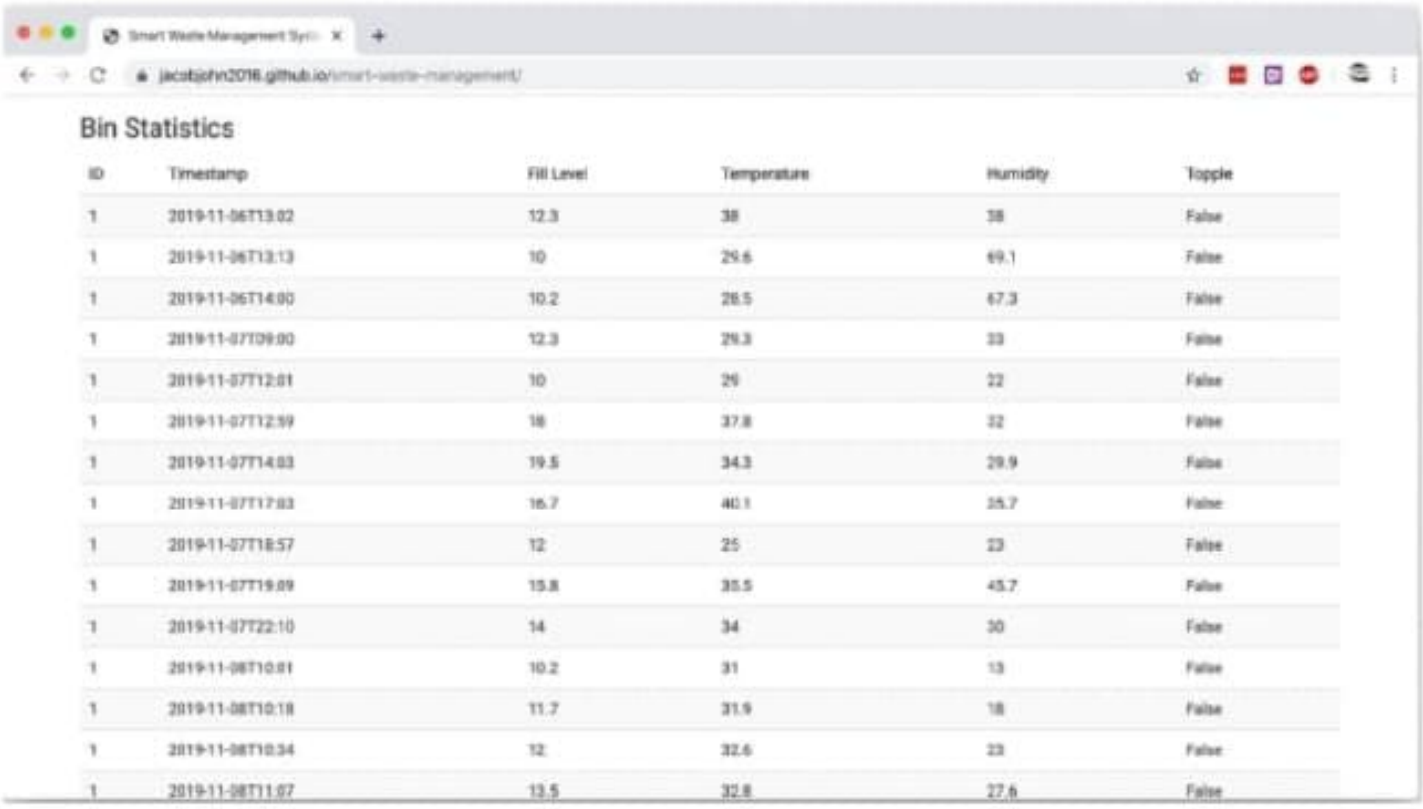
According to Arduino coding implementation in algorithm 1, when a person is standing at a distance less than 20 cm near the smart dustbin, the open() function enables the automatic opening of the lid. Whereas, the close() function runs the servo motor and closes the bin. The smart bin also provides the added benefit of being able to adapt to a bin of any size. The ultrasonic sensor used for checking the fill level in the bin is also used for calibrating the Arduino. Fill-level is measured by comparing it to the initial bin-height in loop(). The fill-level value is mapped to the LEDs using map().

When the bin surpasses a certain threshold for the accelerometer for a certain period of time, the bin will automatically seal shut by issuing the close() command and exit() the program. The management personnel will be hence notified on the dashboard that the dustbin has toppled and receive a browser notification. Similarly, when the temperature and humidity exceed a certain threshold, the bin will seal shut. This is to prevent any excess odors from escaping the bin. The municipality can then safely dispose of any waste from the bin. Furthermore, a high temperature would also imply that the bin might be ignited, which is an extreme safety hazard. In this case, a high priority browser notification is sent multiple times to the management to indicate immediate intimation. The ultrasonic sensor works on the principle of transmitting and receiving an echo signal in order the calculate distance. Upon setup, five sets of distances will be calculated and averaged to calculate the distance to the base of the bin. Taking the average also helps improve the precision of the height of the bin. This height would be used for calculating the percentage the dustbin is filled. Furthermore, the final percentage will be mapped onto an LED array that will indicate to an outsider, what percentage of the bin is full. The accelerometer detects the topple of the dustbin by keeping a check on the amount of time the accelerometer has sensed movement. It is checked by keeping a reference time, ckTime, which is 2000 ms. computeHeatIndex() is an efficient function to determine heat generated in the bin, based on data readings collected from the temperature and humidity sensor.

Algorithm 1 - Sensors Data Collection Algorithm Running on the Arduino

```
FUNCTION measure_ambi()  // Function to measure the distance of the bin from man
BEGIN
   SET dist = (duration/2)/29.1  // Distance is measured
RETURN
END

FUNCTION measure_fill()
 BEGIN
   SET duration_fill = echo_pin. (HIGH)  // Computation of filling duration
   RETURN duration_fill
 END

FUNCTION OPEN(DEVICE, d)  // d is the distance
BEGIN
 IF ( d < 20) THEN
        CALL OPEN(DEVICE);
        ATTACH (servo_pin);  // Attached the servo pin
        CALL DELAY(delay,1);   // DELAY for 1 microsecond
       WRITE(servo_pin,0);       // WRITE 0 to servo_pin
       CALL DELAY(delay, 3000);     //DELAY for 3000 microseconds
       WRITE(servo_pin, 150);          // WRITE 150 to servo_pin
        CALL DELAY(delay, 1000);   // DELAY for 1000 microseconds
        DETACH servo_pin;
       CALL CLOSE(DEVICE);
ENDIF
RETURN
END

FUNCTION CLOSE(DEVICE)
BEGIN
   ATTACH (servo_pin);  // Attached the servo pin
   CALL DELAY(delay,1);   // DELAY for 1 microsecond
   WRITE(servo_pin, 150);          // WRITE 150 to servo_pin
 CALL DELAY(delay, 1000);   // DELAY for 1000 microseconds
 DETACH servo_pin;
CHECK(DEVICE_STATUS);  // Status is checked
IF(DEVICE_STATUS==OPEN) THEN
    CLOSE();
ENDIF
RETURN
END

Function calibrate_adxl()
BEGIN
  for i=0 to samples
  BEGIN
      READ analog signal to x_pin;   // reading the analog signal and stored in x
      READ analog signal to y_pin;   // reading the analog signal and stored in y
      READ analog signal to z_pin;  // reading the analog signal and stored in z
     xsample /= samples;  // new x sample
     ysample /= samples;  // new y sample
     zsample /= samples;  // new z sample
  END
  RETURN
  END

FUNCTION setup()
BEGIN
  SET arduino to transmit 9600 bits per second;
  CALL DELAY(delay, 5000);   // DELAY for 5000 microseconds
  CALL  calibrate_adxl()
  SET tapple = false;
  ATTACH (servo_pin);  // Attached the servo pin
  SET trig_pin_ambi as OUTPUT;
  SET echo_pin_ambi as INPUT;
  SET trig_pin_fill as OUTPUT;
  SET echo_pin_fill as INPUT;
```

```
        WRITE(servo_pin,0);        // WRITE 0 to servo_pin
        CALL DELAY(delay, 100);    // DELAY for 100 microseconds
        DETACH servo_pin;
        for thisLed=0 to ledCount-1
            SET ledPins[thisLed] as OUTPUT;
        for i in 0 to 2
        BEGIN
            cali_height[i] = call measure_fill();
            CALL DELAY(delay, 1000);    // DELAY for 1000 microseconds
            bin_height = (cali_height[0]+ cali_height[1]+cali_height[2])/3;
            dht.begin();
        END
    RETURN
    END

    FUNCTION loop()
    BEGIN
        for i in 0 to 2
        BEGIN
            aver_ambi[i] = measure_ambient_temperature();
            CALL DELAY(delay, 10);    // DELAY for 10 microseconds
            dist_ambi = (aver_ambi[0] + aver_ambi[1] + aver_ambi[2]) / 3;
            open(dist_ambi);
            int fill_level = measure_fill();
            int ledLevel = map(fill_level, 0, bin_height, 0, ledCount);
            for thisLed=0 to ledCount
            BEGIN
              if (thisLed < ledLevel) then
                  WRITE HIGH to ledPins[thisLed];
              else
                  WRITE HIGH to ledPins[thisLed];
              endif
              value1 = analogRead(xpin);
              value2 = analogRead(ypin);
      value3 = analogRead(zpin);
      xValue = xsample - value1;
      yValue = ysample - value2;
      zValue = zsample - value3;
      if (xValue < minVal or xValue > maxVal or yValue < minVal or yValue > maxVal
          or zValue < minVal or zValue > maxVal) then
       if (topple == false) then
         start=millis();
         topple=true;
       endif
      else if ( topple == true) then
      if(millis()>=start+ckTime) then
            PRINT 'dustbin has fallen';
      endif
            close();
            exit(1);
      hum = dht.readHumidity();
      temp = dht.readTemperature();
      hic = dht.computeHeatIndex(temp, hum, false);
      if (hum and temp > threshold) then
        close();
      endif
      if(temp > higher_threshold) then
            PRINT 'Immediate supervision necessary';
      endif
      exit(1);
      END
    END
    RETURN
END
```



Bin Statistics

ID	Timestamp	Fill Level	Temperature	Humidity	Topple
1	2019-11-06T13:02	12.3	38	38	False
1	2019-11-06T13:13	10	29.6	69.1	False
1	2019-11-06T14:00	10.2	28.5	67.3	False
1	2019-11-07T09:00	12.3	29.3	33	False
1	2019-11-07T12:01	10	29	22	False
1	2019-11-07T12:59	18	37.8	32	False
1	2019-11-07T14:03	19.5	34.3	29.9	False
1	2019-11-07T17:03	16.7	40.1	35.7	False
1	2019-11-07T18:57	12	25	23	False
1	2019-11-07T19:09	15.8	35.5	45.7	False
1	2019-11-07T22:10	14	34	30	False
1	2019-11-08T10:01	10.2	31	13	False
1	2019-11-08T10:18	11.7	31.9	18	False
1	2019-11-08T10:34	12	32.6	23	False
1	2019-11-08T11:07	13.5	32.8	27.6	False

Fig. 4 A dynamically produced table updated in real-time using data extracted from Google's Firebase Realtime FireStore sourced from the smart bin

Fig. 5 A time-series graph of the data generated from the smart bin constructed using Javascript libraries from ApexCharts.js. Top–Bottom: Garbage fill level in cms, Temperature in degrees Celsius, Humidity in percentage

The readings from the sensors are communicated to the website via ESP8266 interfacing. Tale 2 displays this algorithm, which uses Firebase. The website displays readings collected by the sensors and updates them every 15 min, keeping it real-time. Using bootstrap and advanced javascript functionalities, line graphs are efficiently displaying the changes in data over time. For example, over the day, the degrees of temperature and humidity calculated in the bin or the amount of waste generated based on the fill-level status can be comprehended by the user- friendly graphs. Further predictions on this data collection have also been undertaken based on LSTM models for Time Series forecasting.

The firebase algorithm is for retrieving the information sensed by the various sensors. In setup(), the Wifi connection status is displayed. In loop(), the firebase is started, and the data sensed by sensors like temperature, humidity, and fill level is read when the difference in the current and previous timestamp is more than the interval defined or when the dustbin is toppled. Then, the JSON object is updated, and the new data is pushed to the firebase database created initially. This is how the flow of data sensed from hardware to the website is incorporated with ease. Algorithm 2 gives the Firebase integration running on the Arduino in collaboration with the ESP8266.

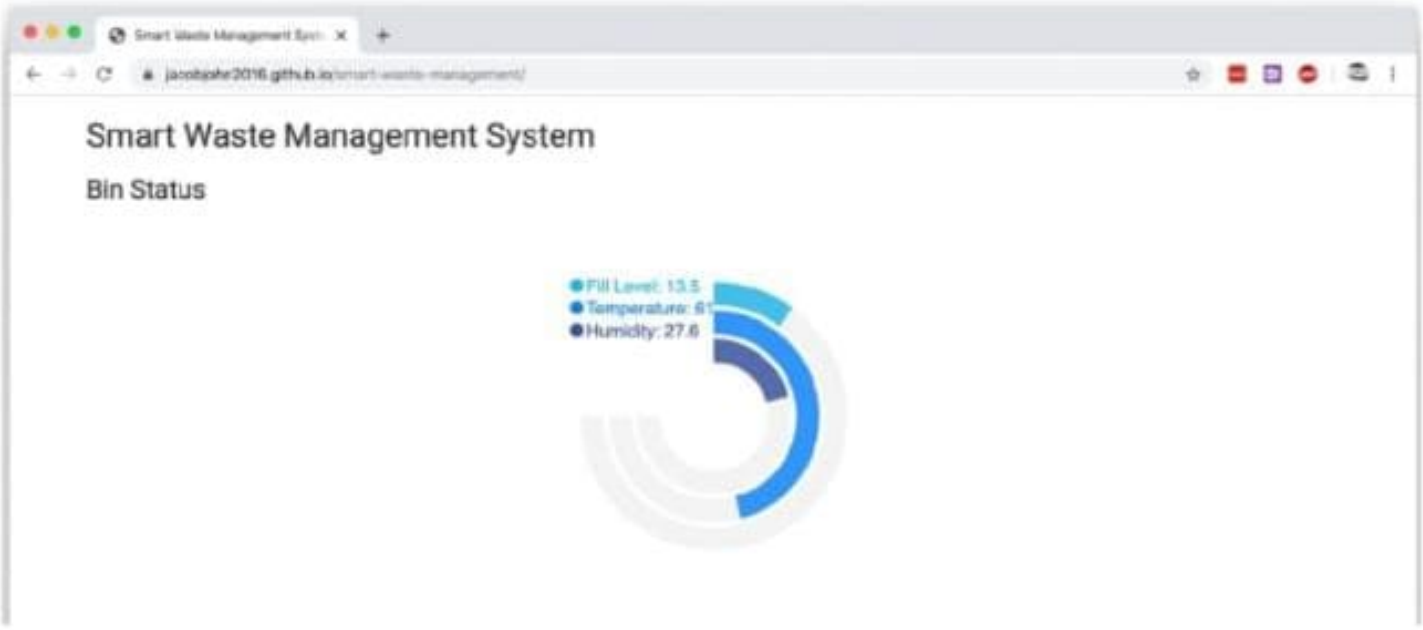


Fig. 6 Realtime status of the bin using the most recent snapshot of the data obtained from the smart bin sensors

Table 1 A comparison table of the various different smart bin implementations available with their feature set along with our proposed smart bin system

	Automated segregation	Cloud storage	Temperature and/or humidity sensors	Optimized collection routes	Prediction of garbage levels	Notification/ alert system	Automated opening/closing	Realtime statistics
Recycle.io [10]	✓	✓				✓		✓
[11]						✓		
[12]		✓			✓	✓		✓
[13]		✓		✓	✓	✓		
[14]		✓		✓		✓	✓	✓
[16]		✓	✓			✓	✓	✓
[17]		✓		✓	✓	✓		✓
[19]		✓	✓			✓		✓
[20]		✓				✓		✓
Internet of bins [21]		✓		✓		✓		✓
[22]		✓				✓	✓	✓
[23]						✓		✓
[25]	✓		✓			✓		✓
[26]		✓					✓	✓
Proposed System		✓	✓		✓	✓	✓	✓

Algorithm 2 - Algorithm for Firebase integration running on the Arduino in collaboration with the ESP8266

```
FUNCTION setup()
BEGIN
  digitalWrite(BUILTIN_LED, HIGH);
  dht.begin();
  Print CONNECTING_TO;
  PRINT (wifi⟶SSID() + ", " + wifi⟶psk());
  wifi->connect();
  while (WiFi.status() != WL_CONNECTED) BEGIN // when wi-fi is connected
      delay(500);
  PRINT lp++;
  if ( lp >= 20) // When more than 20 lines are printed
        break;
 END
 CLEAR BUFFER;
 PRINT CONNECTED TO;
 PRINT IP ADDRESS;
 Start Firebase;
RETURN
END
FUNCTION Loop()
BEGIN
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= Interval or readTopple() is TRUE) then
  BEGIN
      previousMillis = currentMillis;
      readHumidity(); //  Reading humidity
      readTemperature();  // Reading Temperature
      readFillLevel();      //Reading Fill level
      if (lp >= lp_time) then
      BEGIN
              lp = 0;
            JsonObject &root = jsonBuffer.createObject();
              Update JsonObject &root;
              Firebase.push("/sensor/dht", root);
      END
  END
RETURN
END
```

4 Implementation, Results and Discussions

Data was collected using the system for three days at different time intervals. Figure 4 shows the tabulation of the data collected sorted chronologically. It shows the bin id, timestamp, fill level, humidity, temperature, and topple status. Figure 5 shows the time-series graphs.

The fill level graph helps us understand at what time of the day the trash is usually reaching its maximum fill level when it is emptied, and if it is being emptied when not necessary. The temperature and humidity values help us to prevent fires. Figure 6 shows the real-time status of the bin using the data last collected by the system. Table 1 gives the various different smart bin implementations available with their feature set along with our proposed smart bin system.

The firebase generated graphs were also used to notify the appropriate people when the bins were full and needed to be emptied. This was done using firebase cloud messaging (FCM). FCM JavaScript API was used to send notification messages in web apps in browsers that provide service work support.

Figure 6 gives the Realtime status of the bin using the most recent snapshot of the data obtained from the smart bin sensors. Another essential feature of this smart waste management system is its ability to predict waste generated over a series of intervals. Since the bin collects data every 15 min over the entire day, it contributes to the creation of a dataset with a vast volume, thereby making it highly suitable for the creation of a neural network model. One of the essential characteristics of the data set is that the data points are in a time series. This means that data is realized at regularly spaced success intervals, which results in the formation of a discrete-time dataset. Heights of ocean tides, number of passengers that travel by airline, and sales of a particular product are some examples of datasets. Traditional regression algorithms such as logistic and linear regression tend to fall short when the data involves long term dependencies or lacks a particularly linear pattern. This implementation only performs a time series prediction of the fill level of the dataset as temperature and humidity are very complex variables to perform predictions on. They are confounded by several natural phenomena that need to be accounted for while performing a prediction. Thus, requiring a significant number of resources and time. Furthermore, weather agencies already have highly developed models for predicting weather patterns.

Figure 7 shows the original data set obtained. The total dataset contains over 160 data points, which should be sufficient for creating a regression model. Each sudden, sharp trough that inflects at 0 cm is when the fill had to be emptied out and fills back up to 30 cm. This implementation assumes that the bin is to be emptied immediately after reaching the complete fill level for the sake of data analysis. A Long Short-Term Memory (LSTM) network can be used as a deep learning model to make this prediction. It can handle sequence dependencies introduced by the complexities of time series data. The Keras deep learning, developed by Chollet [35] and available for free on Tensorflow, provides us with a powerful LSTM network package. Furthermore, LSTMs also handle long term sequence dependencies making them highly suitable for data that relies on context. They also overcome the vanishing gradient problem that is typically experienced by DNNs. In addition to this, LSTMs are also a form of Deep Neural Networks, which works in harmony with the extensive data set produced by the smart waste system. The time series prediction of bin fill level can be treated as a regression problem instead—predicting the bin level in the next 15 min based on the current level of waste. A function is used to convert the single column data into two column data in the dataset. The first column contains data recorded currently at

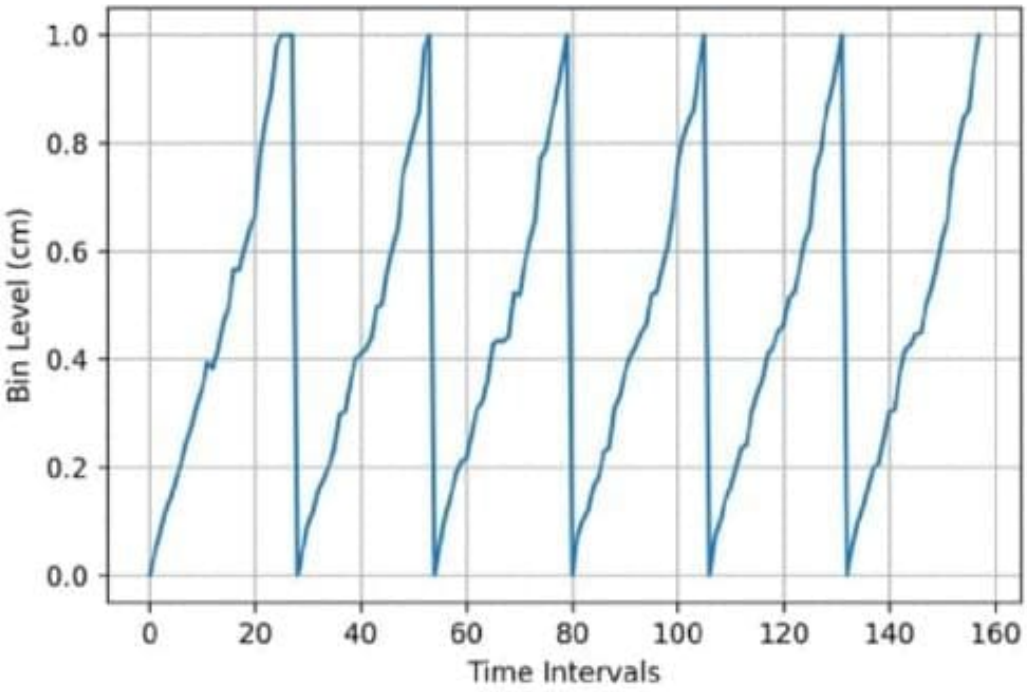


Fig. 7 Graph showing the bin level in cms over 160 data points collected

time t, while the second column contains data recorded in the next 15 min or t+15, which is to be predicted by the LSTM. The data was then normalized to a scale of 0 to 1 to allow it to work with the tanh or sigmoid activation function.

The architecture of the network as displayed in Fig. 8 includes one visible layer, which is the input layer, followed by four hidden layers, which are the memory blocks and an output layer with the default sigmoid activation function. This is the first of the six models that were trained and compared.

The second model, as displayed in Fig. 9, uses five hidden layers instead, along with a batch size of 2 (LSTM-2). The other five models used a batch size of 1. This is to ensure that we find the best hyper parameters for the LSTM models. Comparisons were made on the basis of three evaluation metrics—RMSE, MAE, and R2 errors.

The third model, LSTM-WM, is an LSTM regression model constructed using the window method is shown in Fig. 10. Rather than using just the current fill level and the next fill level, the LSTM can work with a larger window of data collected during the prior time intervals as well. A window size of four was used in this case, with t−30, t−15, t and t+ 15 in order to make the time series prediction. The first three dimensions are fed into the input layer and the last one is used as the output. Similar to the first LSTM, a single neuron with four hidden layers is used in this work. However, this LSTM neuron accepts only an input of 1×3 at a time.

The fourth model, labeled as LSTM-TS, is an LSTM for regression with time step is shown in Fig. 11. Similar to the window method, a t−30, t−15 and t is passed as a look back. However, the time step method takes a dimension of 3×1 as input. Increasing the dimensions of the train and test set allows the LSTM's input to utilize more time steps. Unlike the window method, which uses the observations as separate input features, this method uses them as time steps of one input feature.

The fifth model in Figure 12 uses memory between batches, hence the name LSTM-MBS. The LSTM-MBS uses the same backbone architecture as the LSTM-WM. However, by making the model "stateful," the algorithm doesn't reset the states in the network after training each batch nor while making predictions. Now, the model can build a state over

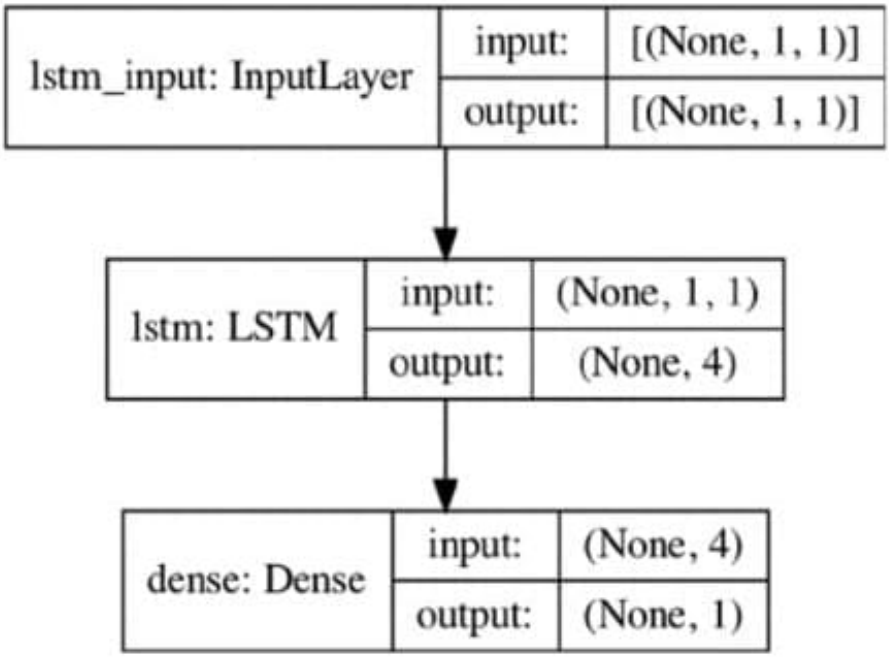


Fig. 8 The LSTM architecture with one LSTM neuron with four hidden layers

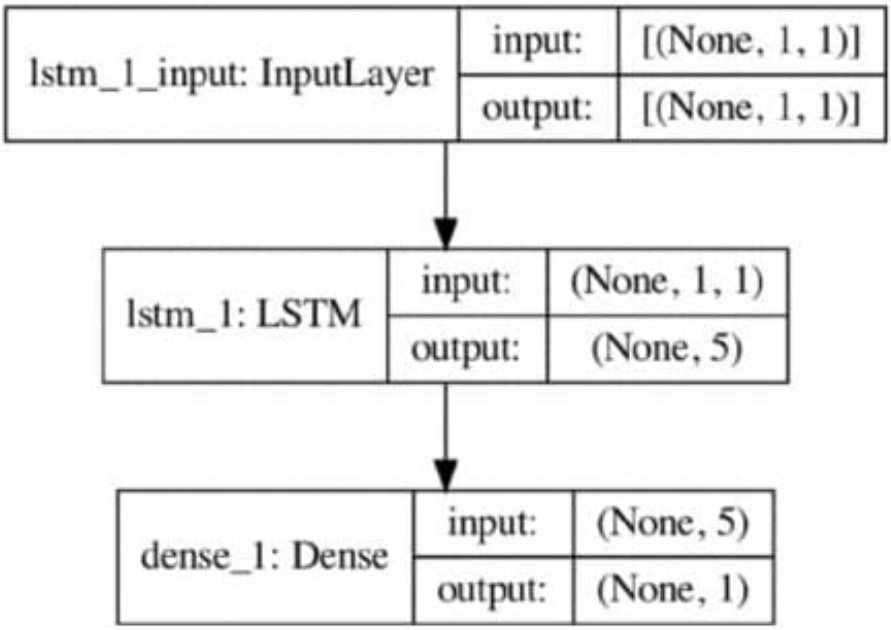


Fig. 9 LSTM 2 with 5 hidden layers in the LSTM neuron

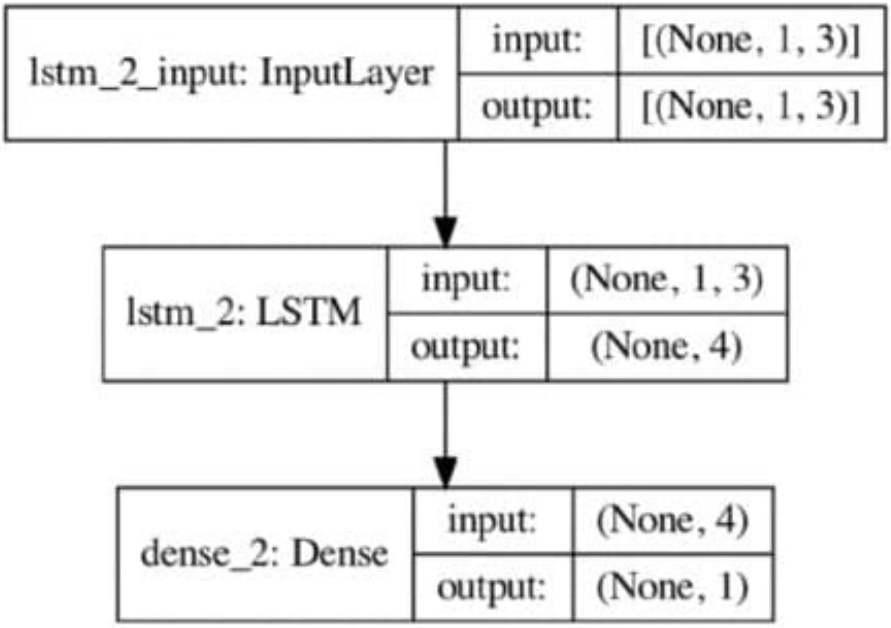


Fig. 10 The LSTM Window Method using a 1×3 dimensional input layer

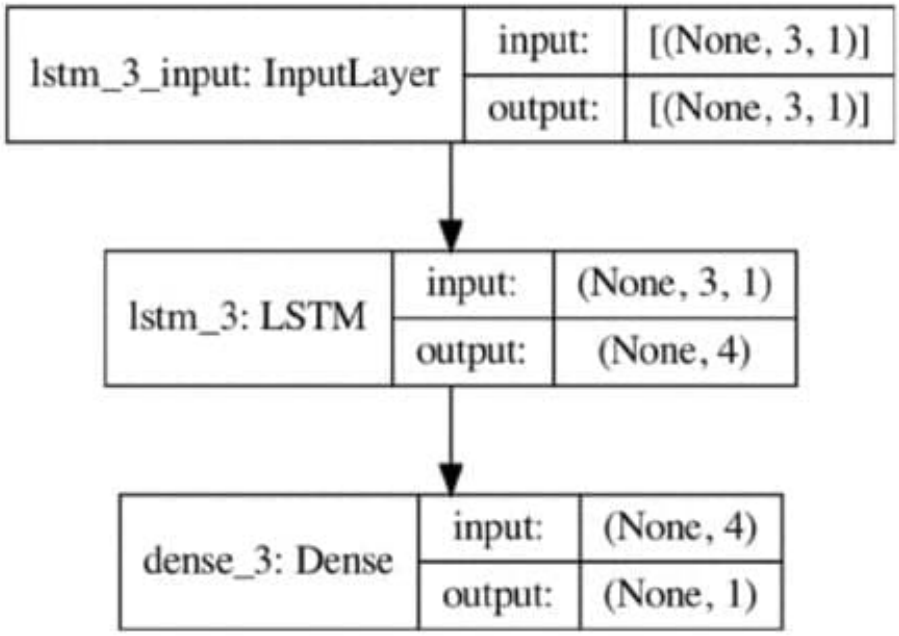


Fig. 11 The LSTM Time Step method using a 3×1 dimensional input layer

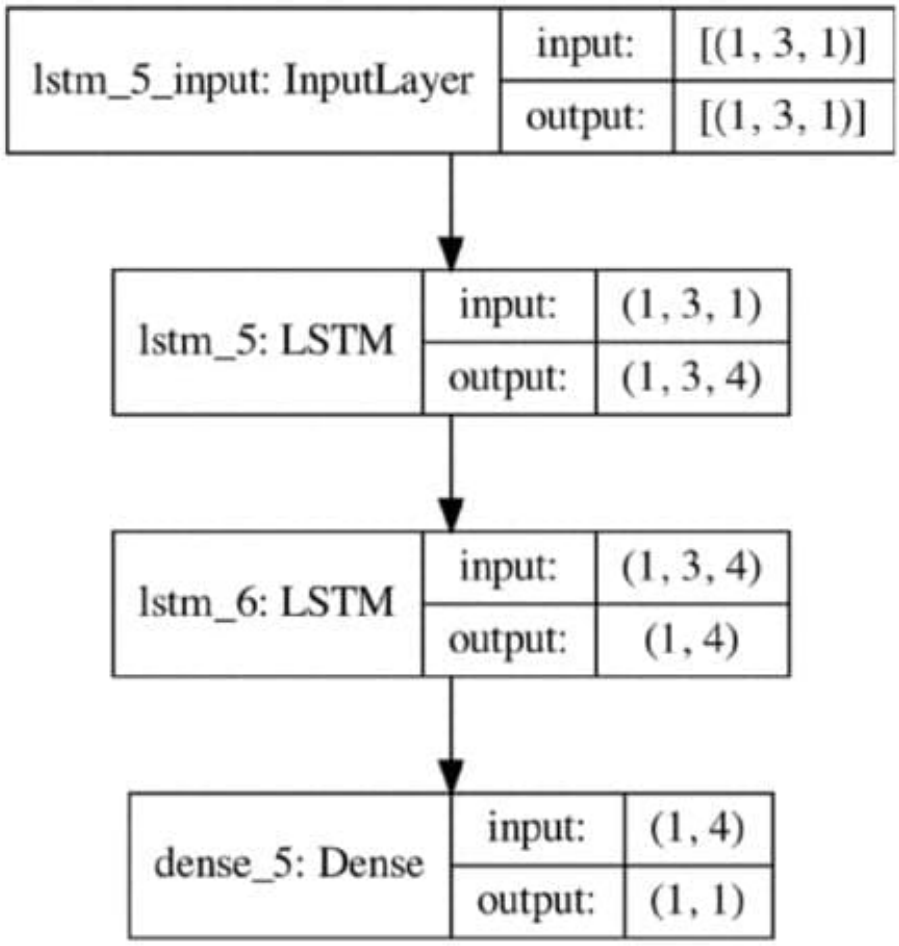


Fig. 12 The stacked LSTM model using two LSTM neurons

the entire training vector. Furthermore, the state will also be maintained while evaluating the testing set.

The S-LSTM model also uses the same principle as the LSTM-WM in order to maintain states between batches. However, in addition to this, it stacks another four layers of hidden LSTM neurons on top of the existing one to make the neural network deeper, thereby assisting in training the network. However, in smaller datasets, a deeper model may penalize accuracy as the weights would take longer to converge, despite using the Adam optimizer.

The networks were trained for 200 epochs with a batch size of 1 for the five models and a batch size of 2 for the second model. Figure 13 displays a graph that compares the Root mean squared error (RMSE) or loss over 200 epochs among the five models.

As graph displays, the S-LSTM performs the best over the 200 epochs due to its lowest RMSE loss as compared to the other algorithms is shown in Fig. 19. Although the S-LSTM starts with a significantly low RMSE score, the model takes a longer time to train.

This can be observed in Fig. 13 as the line for the S-LSTM takes of the longest time to plateau, similar to the LSTM-MBS. Henceforth indicating that although resetting the states make results in longer training time for such a dataset, it yields better results. Another essential thing to note is that the standard LSTM and LSTM-2 take the shortest time to train, and the loss plateaus in a few epochs. However, they perform significantly worse than other LSTM models. Stacking up LSTMs is also likely to increase training time as the network is much deeper thereby requiring more weights to be learnt.

Table 2 provides further evidence for S-LSTM significantly better performance. The stacked LSTM has a better RMSE, MAE, and R2 score in both training and testing, by a significant margin. Figures 14, 15, 16, 17, 18, and 19 show the graphs with the result of the predictions in the training set as orange and the test set as green. It should also be noted that unlike traditional machine learning algorithms and ANNs, the LSTMs didn't memorize the dataset. This is clearly reflected by how most algorithms were unable to predict the peak of both the training and testing set. Furthermore, none of the six models overlay the original dataset; every figure consists of at least some form of irregularity.

The stacked LSTM is henceforth a much better fit for the dataset and in the prediction of fill levels. However, this is assuming that the fill levels constantly cycle at a repeated rate. An extension of this paper would be to collect waste levels over the years for a particular location. Thereby controlling more variables and understanding how the waste is generated over a yearly basis. This, however, couldn't be performed in this paper due to obvious time constraints.

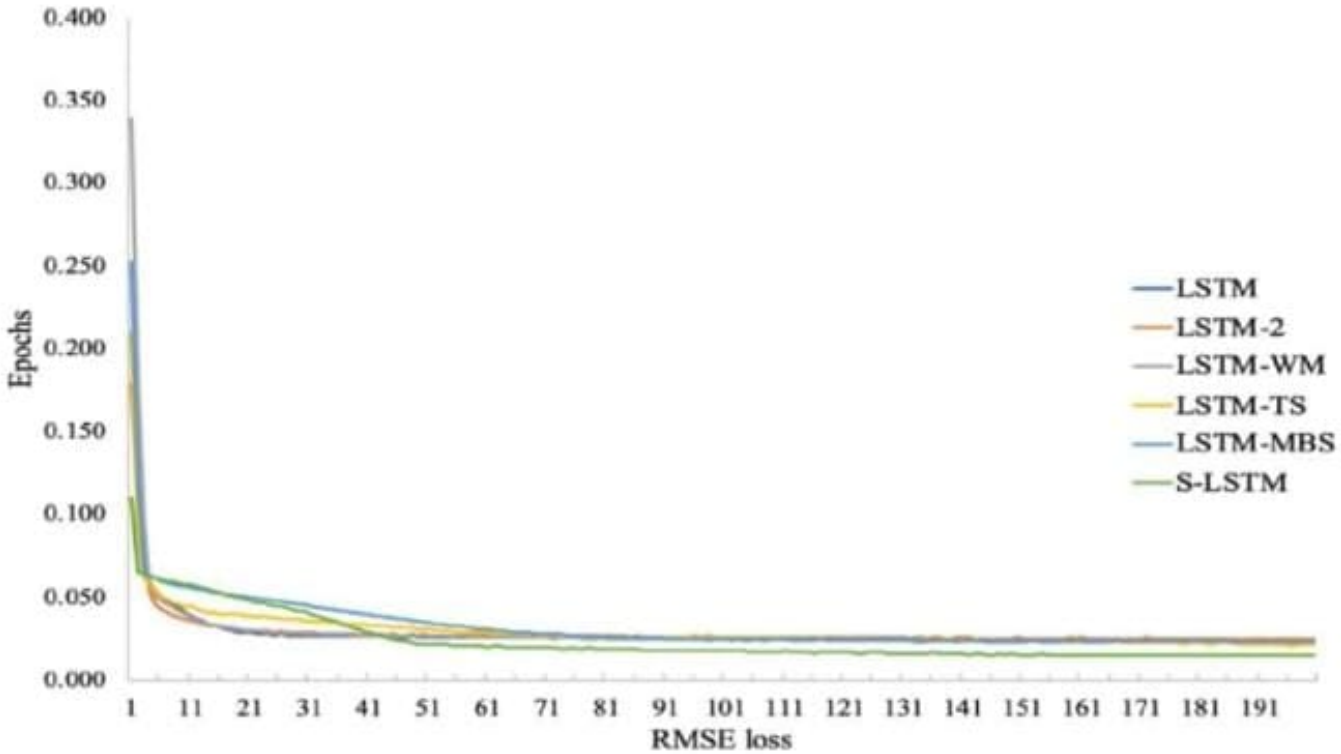


Fig. 13 Graph comparing the RMSE losses among LSTM (with 4 neurons), LSTM-2 (batch size of 2), LSTM-WM (using Window Method), LSTM-TS (with Time-Steps), LSTM-MBS (with Memory Between Batches), S-LSTM (Stacked LSTM with Memory Between Batches) over 200 epochs

Table 2 Comparisons between the evaluation metrics received from each model

	RMSE		MAE		R2	
	Train score	Test score	Train score	Test score	Train score	Test score
LSTM	4.71	5.28	2.07	2.18	0.69	0.58
LSTM-2	4.80	5.38	2.08	2.23	0.68	0.57
LSTM-WM	4.40	3.76	1.93	1.61	0.72	0.78
LSTM-TS	4.31	3.63	1.88	1.48	0.73	0.79
LSTM-MBS	4.60	3.87	1.96	1.64	0.70	0.76
S-LSTM	**3.48**	**2.31**	**1.34**	**0.92**	**0.92**	**0.83**

RMSE is Root Mean Squared Error, MAE is Mean Squared Error, and R2 is R-Squared error. In bold are the best evaluation metrics obtained from each category

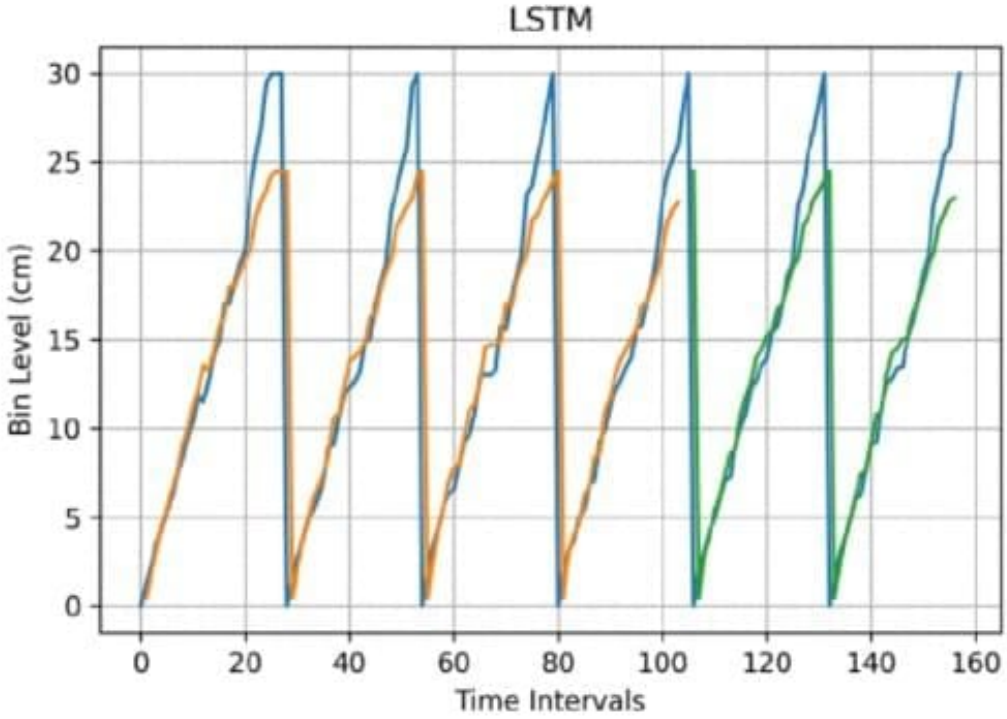


Fig. 14 The predictions obtained from all models with the training set predictions in orange and testing set predictions in green. From top–bottom: LSTM (with 4 neurons)

5 Conclusions and Future Works

In this work a new smart and intelligent waste management system has been proposed using deep learning techniques and IoT. The smart features of the dustbin include the automatic lid opening system that works based on sensing of the personnel in front and also using the amount of waste generated depending on the level of dustbin filled. The temperature and humidity readings are helpful in calculating heat and carbon monoxide levels produced. If waste lies in the dustbin for long periods of time, it starts to give off a stench,

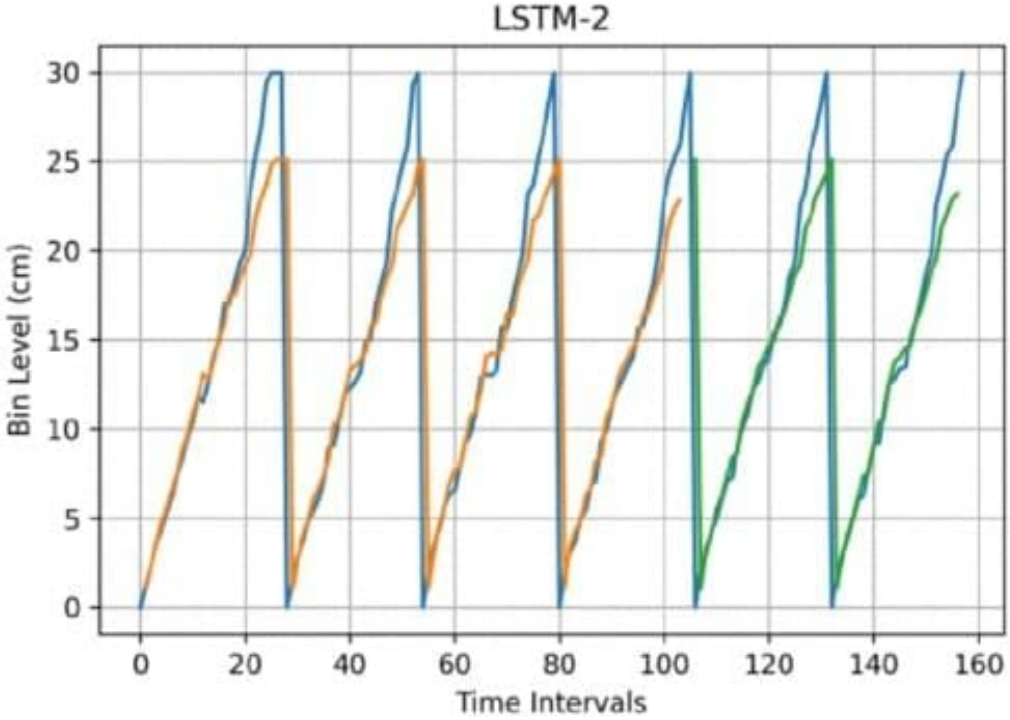


Fig. 15 The predictions obtained from all models with the training set predictions in orange and testing set predictions in green with LSTM-2 (batch size of 2)

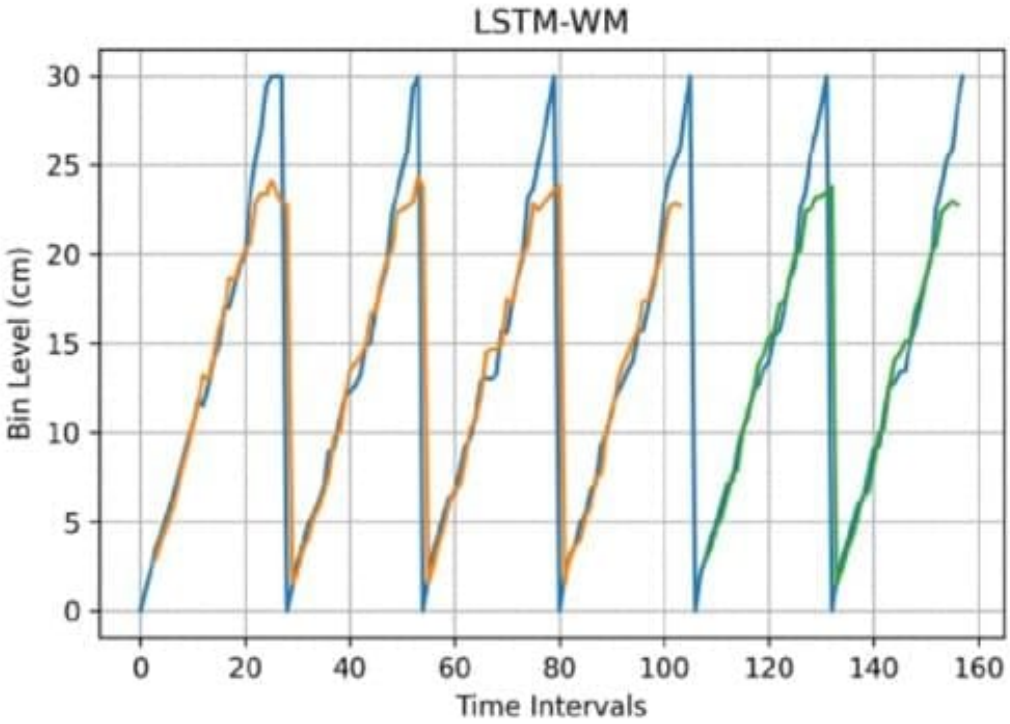


Fig. 16 The predictions obtained from all models with the training set predictions in orange and testing set predictions in green with LSTM-WM (using Window Method)

which is unhygienic for the area. Since there is no direct sensor for measuring bad odor, measuring the levels of carbon monoxide has been used in this work. Once a threshold of temperature and humidity value is reached, the dustbin shutters close until the clean-up process is done by municipal workers. The readings also help in detecting flames. The smart dustbin can also inform the authorities if a dustbin has toppled, spilling waste and making an area dirty. The predictions are carried out in this work using time series on the data collected by these sensors and this is an added smart feature. It can effectively predict the future scenarios of the dustbin, for example, how often the dustbin is to be cleared over the day, or how much waste is generated in different geographical areas. This feature can

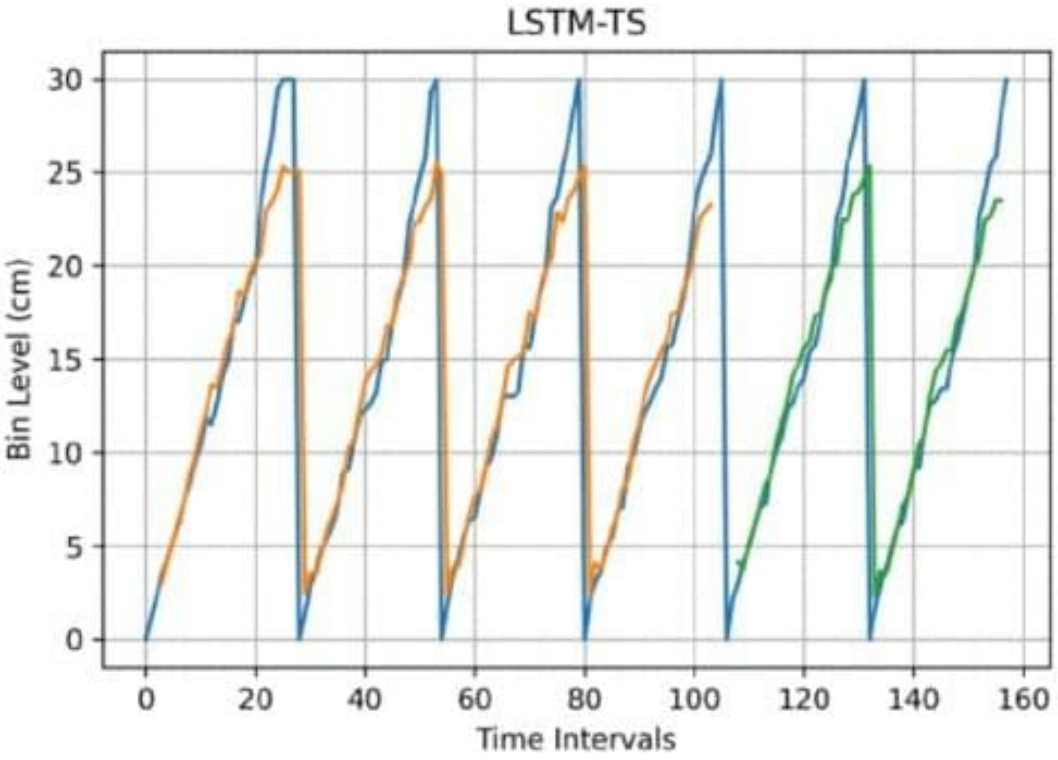


Fig. 17 The predictions obtained from all models with the training set predictions in orange and testing set predictions in green with LSTM-TS (with Time-Steps)

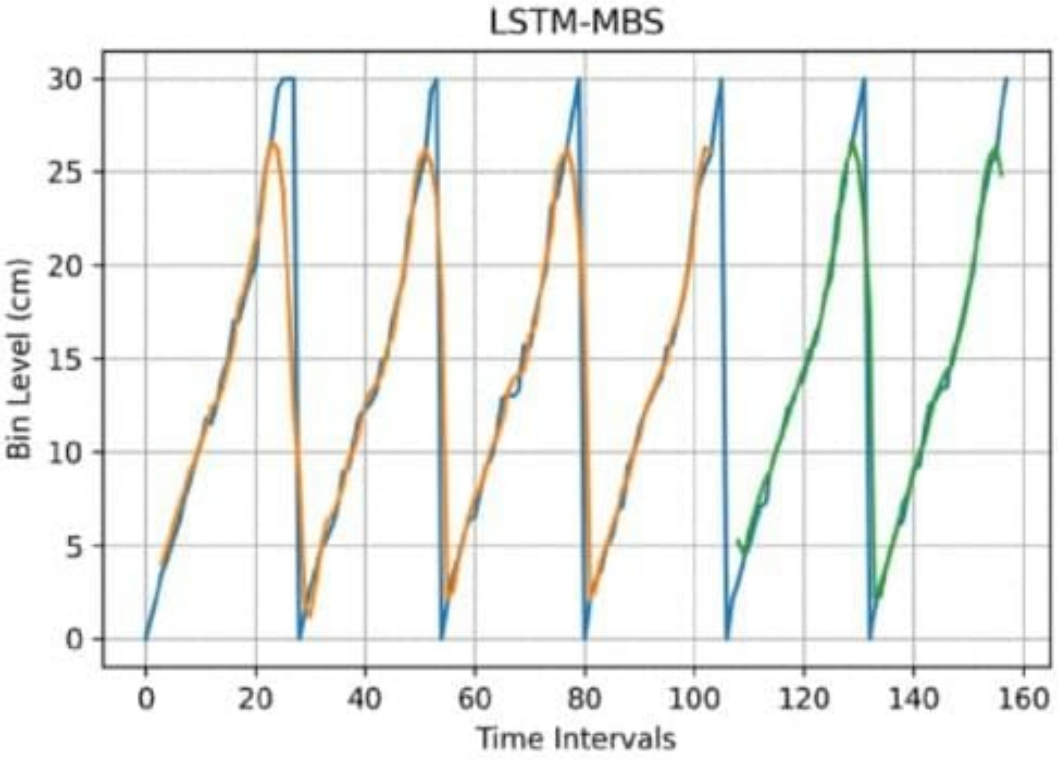


Fig. 18 The predictions obtained from all models with the training set predictions in orange and testing set predictions in green with LSTM-MBS (with Memory Between Batches)

be of much use to municipalities. Thus, there are various features incorporated in our smart dustbin using IoT. AS a future work, the proposed system can be improved to make it more aesthetic and consumer-friendly. The current implementation is highly scalable due to its unique additional features and online accessibility of the collated data. However, bin packing algorithms such as next fit, first fit and best fit algorithms can be added to this work for optimizing the number of bins.

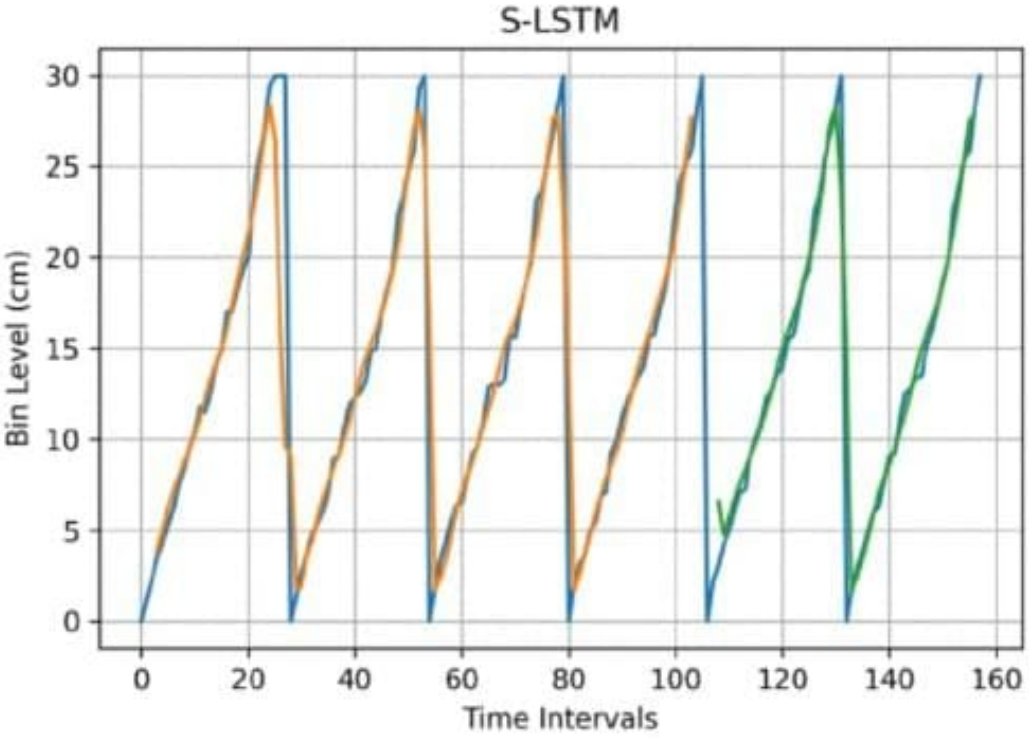


Fig. 19 The predictions obtained from all models with the training set predictions in orange and testing set predictions in green. For S-LSTM (Stacked LSTM with Memory Between Batches)

References

1. Reddy, V. K., & Ram, A. S. (2019). Waste management initiatives and activities in India for society's welfare. *International Journal of Scientific and Technology Research, 8*, 2995–2998.
2. Wray, S. (2020). *Small things add up: The Japanese town leading the zero-waste charge*. Smart Cities World. Retrieved May 5, 2021, from https://www.smartcitiesworld.net/special-reports/special-reports/small-things-add-up-the-japanese-town-leading-the-zero-waste-charge
3. Tahsien, S. M., Karimipour, H., & Spachos, P. (2020). Machine learning based solutions for security of Internet of Things (IoT): A survey. *Journal of Network and Computer Applications, 161*, 1–18.
4. Alaa, M., Zaidan, A. A., Zaidan, B. B., Talal, M., & Kiah, M. L. M. (2017). A review of smart home applications based on Internet of Things. *Journal of Network Computer Application, 97*, 48–65.
5. Singh, A., Payal, A., & Bharti, S. (2019). A walkthrough of the emerging IoT paradigm: Visualizing inside functionalities, key features, and open issues. *Journal of Network and Computer Application, 143*, 111–151.
6. Asghari, P., Rahmani, A. M., & Javadi, H. H. S. (2019). Service composition approaches in IoT: A systematic review. *Journal of Network and Computer Application, 120*, 61–77.
7. Alaba, F. A., Othman, M., Hashem, I. A. T., & Alotaibi, F. (2017). Internet of Things security: A survey. *Journal of Network and Computer Applications, 88*, 10–28.
8. Gupta, S., Mohan, K., Prasad, R., Gupta, S., & Kansal, A. (1998). Solid waste management in India: Options and opportunities. *Resources, Conservation and Recycling, 24*(2), 137–154.
9. Mdukaza, S., Isong, B., Dladlu, N., & Abu-Mahfouz, A. M. (2018). Analysis of IoT-enabled solutions in smart waste management. In *IECON 2018-44th annual conference of the IEEE industrial electronics society* (pp. 4639–4644). IEEE.
10. Al-Masri, E., Diabate, I., Jain, R., Lam, M. H., & Nathala, S. R. (2018). Recycle.io: An IoT-enabled framework for urban waste management. In *2018 IEEE international conference on big data (big data)* (pp. 5285–5287). IEEE.
11. Navghane, S. S., Killedar, M. S., & Rohokale, V. M. (2016). IoT based smart garbage and waste collection bin. *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE), 5*(5), 1576–1578.
12. Baby, C. J., Singh, H., Srivastava, A., Dhawan, R., & Mahalakshmi, P. (2017). Smart bin: An intelligent waste alert and prediction system using machine learning approach. In *2017 international conference on wireless communications, signal processing and networking (WiSPNET)* (pp. 771–774). IEEE.

13. Bakhshi, T., & Ahmed, M. (2018). IoT-enabled smart city waste management using machine learning analytics. In *2018 2nd international conference on energy conservation and efficiency (ICECE)* (pp. 66–71). IEEE.
14. Mirchandani, S., Wadhwa, S., Wadhwa, P., & Joseph, R. (2017). IoT enabled dustbins. In *2017 international conference on big data, IoT and data science (BID)* (pp. 73–76). IEEE.
15. Mukherjee, S., Bhattacharyya, B., & Banerjee, N. (2017). Harnessing green energy for smart dustbin. In *2017 IEEE international conference on smart technologies and management for computing, communication, controls, energy and materials (ICSTM)* (pp. 484–490). IEEE.
16. Poddar, H., Paul, R., Mukherjee, S., & Bhattacharyya, B. (2017). Design of smart bin for smarter cities. In *2017 innovations in power and advanced computing technologies (i-PACT)* (pp. 1–6). IEEE.
17. Shyam, G. K., Manvi, S. S., & Bharti, P. (2017). Smart waste management using Internet-of-Things (IoT). In *2017 2nd international conference on computing and communications technologies (ICCCT)* (pp. 199–203). IEEE.
18. Saha, H. N., Auddy, S., Pal, S., Kumar, S., Pandey, S., Singh, R., Singh, A. K., Banerjee, S., Ghosh, D., & Saha, S. (2017). Waste management using Internet of Things (IoT). In *2017 8th annual industrial automation and electromechanical engineering conference (IEMECON)* (pp. 359–363). IEEE.
19. Devi, P., Ravindra, W. S., & Prakash, S. S. (2018). An IoT enabled smart waste management system in concern with Indian smart cities. In *2018 2nd international conference on trends in electronics and informatics (ICOEI)* (pp. 64–69). IEEE.
20. Kumar, S. V., Kumaran, T. S., Kumar, A. K., & Mathapati, M. (2017). Smart garbage monitoring and clearance system using internet of things. In *2017 IEEE international conference on smart technologies and management for computing, communication, controls, energy and materials (ICSTM)* (pp. 184–189). IEEE.
21. Keerthana, B., Raghavendran, S. M., Kalyani, S., Suja, P., & Kalaiselvi, V. K. G. (2017). Internet of bins: Trash management in India. In *2017 2nd international conference on computing and communications technologies (ICCCT)* (pp. 248–251). IEEE.
22. Kolhatkar, C., Joshi, B., Choudhari, P., & Bhuva, D. (2018). Smart E-dustbin. In *2018 international conference on smart city and emerging technology (ICSCET)* (pp. 1–3). IEEE.
23. Kumar, N. S., Vuayalakshmi, B., Prarthana, R. J., & Shankar, A. (2016). IOT based smart garbage alert system using Arduino UNO. In *2016 IEEE region 10 conference (TENCON)* (pp. 1028–1034). IEEE.
24. Hong, I., Park, S., Lee, B., Lee, J., Jeong, D., & Park, S. (2014). IoT-based smart garbage system for efficient food waste management. *The Scientific World Journal, 5*, 1–3.
25. Kavya, M., Sahana, P., Shruthi, G., Sunitha, M. C., & Jyothi, A. P. (2017). Sensor based smart dustbin for waste segregation and status alert. *International Journal of Advance Research and Innovative Ideas in Education, 2*(5), 98–103.
26. Jain, A., & Bagherwal, R. (2017). Design and implementation of a smart solid waste monitoring and collection system based on Internet of Things. In *2017 8th international conference on computing, communication and networking technologies (ICCCNT)* (pp. 1–5). IEEE.
27. Thangaramya, K., Kulothungan, K., Logambigai, R., Selvi, M., Ganapathy, S., & Kannan, A. (2019). Energy aware cluster and neuro-fuzzy based routing algorithm for wireless sensor networks in IoT. *Computer Networks, 151*, 211–223.
28. Munirathinam, T., Ganapathy, S., & Kannan, A. (2020). Cloud and IoT based privacy preserved e-Healthcare system using secured storage algorithm and deep learning. *Journal of Intelligent & Fuzzy Systems, 39*(3), 3011–3023.
29. Selvi, M., Gayathri, A., Kumar, S. V. N. S., & Kannan, A. (2020). Energy efficient and secured MQTT protocol using IoT. *International Journal of Innovative Technology and Exploring Engineering, 9*(4), 11–14.
30. Sak, H., Senior, A. W., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling.
31. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
32. Zhang, J. S., & Xiao, X. C. (2000). Predicting chaotic time series using recurrent neural network. *Chinese Physics Letters, 17*(2), 88.
33. Sherstinsky, A. (2018). *Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network*. arXiv preprint arXiv:1808.03314.
34. Gers, F. A., Eck, D., & Schmidhuber, J. (2002). Applying LSTM to time series predictable through time-window approaches. In *Neural nets WIRN Vietri-01* (pp. 193–200). Springer.
35. Gulli, A., & Pal, S. (2017). *Deep learning with Keras*. Packt Publishing Ltd.