


RESEARCH ARTICLE

Scheduling and Predictive Maintenance for Smart Toilet

AMAR LOKMAN, R. KANESARAJ RAMASAMY, (Senior Member, IEEE), AND CHOO-YEE TING 

Faculty of Computing and Informatics, Multimedia University, Cyberjaya 63000, Malaysia

Corresponding author: R. Kanesaraj Ramasamy (r.kanesaraj@mmu.edu.my)


This work was supported by TM Research and Development from Telekom Malaysia, Malaysia.

ABSTRACT Modern society needs bathrooms. Poor sanitation is caused by worn-out appliances and expensive cleaning. The technique also requires an inexpensive, dependable sensor. This study had three goals. Creating an IoT administration platform is the main goal. Literature evaluations assess the merits and downsides of existing systems. Second, we suggest predictive maintenance to assist predict bathroom equipment breakdowns. Finally, a scheduling algorithm was used to determine how many janitors to hire. We'll measure the model's effectiveness and make future recommendations. Infrared, temperature and humidity sensors create an IoT bathroom. Sensors have been studied to understand how to adapt them to the hygienic and private toilet environment. Sensor accuracy and cost-effectiveness could be enhanced with more development and testing. The Auto-Regressive Integrated Moving Average (ARIMA) model accurately predicts time series lags, making it a good candidate for predictive maintenance. Long Short-Term Memory (LSTM) is good in time series predictions, therefore it's fair to compare the two. We use the ARIMA model to handle Remaining Useful Life (RUL) prediction techniques by altering Moving Average (MA) and Auto-Regressive (AR). A genetic algorithm is used to create a janitorial cleaning schedule. The genetic algorithm was proposed to schedule cleaning workers. This approach improves the genetic algorithm by studying soft and hard scheduling restrictions. The Greedy algorithm is used to compare. Experimental evaluations reveal that the suggested model ARIGA meets both goals.

INDEX TERMS Scheduling, predictive maintenance, IoT.

I. INTRODUCTION

Increasing global modernization has led to the widespread adoption of advanced technology. IoT is an example (IoT). IoT integrates technological and social domains to solve daily problems [1]. Home automation, smart cities, health-care, smart business, and monitoring systems will use IoT. The simplified IoT view is a real-world network integrated with sensors and drivers that extends the existing Internet dominated by computers and mobile devices. The smart toilet could drive the IoT services and hardware business. The IoT system for smart toilets optimizes resources using cloud-based and sensory data management systems [2]. People still complain about restroom problems. The Star reports that visitors like Malaysia but are disgusted by our public restrooms [3]. According to ioi-analytics.com,

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaolong Li .

“Smart Homes,” “Wearables,” “Smart Cities,” and “Others” employ IoT to increase performance. IoT solutions in Smart Cities reduce traffic congestion, reduce environmental impact, and make cities safer. IoT techniques prioritize energy and material conservation. This study aims to cut waste and costs.

Maintenance using IoT. A maintenance system ensures a building or corporation's resources are in good condition to prevent an unwanted event. Most structures are managed by a computerized system (CMMS). CMMS just shows when anything happened. The author says the printing system's maintenance system is vital to its dependability, availability, and maintainability. This system won't disrupt manufacturing [4]. Predictive maintenance (PdM) predicts equipment failures and takes preventative steps. Each piece of equipment has a predetermined lifetime ranging from hours to decades. For example, the toilet has a flush, pipe, sink, and lamp. Smart toilet lifecycle information. Everything will be monitored

using IoT. With IoT, predictive maintenance sensors may collect data and predict problems.

Connectivity, data volume, innovative gadgets, inventory reduction, adaption, and regulated production generated Industry 4.0. Industry 4 can modify and make data available, and it can enable human or robot action [5]. PdM utilizes historical data, models, and domain expertise. Using statistical or machine learning models, they can detect trends, patterns, and correlations to anticipate impending problems and optimize maintenance to prevent mainly unavailability [6]. PdM and techniques to increase production capacities have led to the words intelligent industry and intelligent manufacturing [7]. Corrective, preventative, predictive, and prescriptive maintenance occurs. Remedial maintenance is performed when a malfunction or warning signal appears. Preventive maintenance schedules are followed. PdM predicts failure based on time and knowledge, preventing downtime.

IoT is helpful for Building Management Systems (BMS). BMS controls and monitors a building's mechanical and electrical system, including lighting, ventilation, electricity, fire, air quality, and comfort management [8], [9]. According to Mega Jadi, cleaning prices depend on square footage and type of service. 1 cleaning per week for 4 weeks costs RM380 in the Klang Valley, 2 cleanings per week for 4 weeks costs RM720, 3 cleanings per week for 4 weeks cost RM1020, 4 cleanings per week for 4 weeks cost RM1280, and 5 cleanings per week for 4 weeks costs RM1500. The charges are based on a single cleaning in a 2500-square-foot office. Every office cleaning takes 2-4 hours. Larger offices may require more labor, which will affect pricing. A building's use pattern-based scheduling system can save cleaning costs. Each toilet has a different use pattern based on data from the smart toilet system, allowing it to display high- and low-use floors. Thus, cleaning staffing can be based on toilet usage.

CPM and PERT have been the de facto standards for scheduling projects during the previous few decades. CPM and PERT always assume a limitless resource supply. Scientists are using genetic algorithms to construct meta-heuristic algorithms. Adapting the implementation technique for resource-constrained building project scheduling won't address the scheduling challenge. Prefabricated buildings can only use project scheduling for repetitive operations, unlike traditional buildings. This research aims to improve janitor scheduling by enhancing an algorithm that uses separate parameters for hard and soft constraints.

Due to privacy and wetness, IoT systems don't work well in toilets or bathrooms. IoT development in healthcare has concentrated on wearable and in-body sensor devices such as digital pills, smart beds, smart food, and smart band-aid [10]. Because of soaking and undressing, these wearable devices aren't ideal for the bathroom. The Author noted that restrooms are unsafe for the elderly [11]. Over 235,000 people visit the ER annually due to bathroom injuries, according to the CDC.

The toilets aren't clean enough to satisfy users. According to New Straits Times (2019), cleaners report seeing dirt in toilets 20 minutes after washing and moist floors. Oddly, most users are adults. 61% of Malaysia's 10,257 public toilets are dirty, according to reports. Only 350 people (3.4%) got five stars, while 1,086 (10%) got no stars. Most public restrooms are in bad condition, according to a Local Government Department audit. Every piece of equipment has a short- or long-term lifespan. Flush, pipe, sink, and lamp are utilized in toilets. According to the author, a three-stage industry life cycle is the most crucial [12].

High-rise cleaning is expensive. The author says airline crew scheduling includes cabin and cockpit crew [13]. The planned schedule would align both crews. This function checks task scheduling, not use patterns. In the Klang Valley, commercial cleaning is commonly done in four-week packages: RM380 for one cleaning per week for four weeks (RM95 for each session), etc. The charges are based on a single cleaning in a 2500-square-foot office. Every office cleaning takes 2-4 hours. Larger offices may require more labor, which will affect pricing. The author compared scheduling methods to determine workload importance but concluded that different articles lack evidence to deliver satisfactory findings [14].

Research objectives assist determine direction and project goals. This study has three objectives. First, create an Internet of Things (IoT) smart toilet management system with sensors to meet user privacy requirements. Second, use ARIMA and LSTM to reduce prediction error and offer accurate predictions. Apply resource-efficient scheduling for the smart toilet using the evolutionary algorithm and the constraint parameter.

The design of Internet of Things systems, predictive maintenance, and scheduling are the focus of this study. The datasets were collected by the Internet of Things system at Multimedia University's computing and informatics faculty and vaccine center. These are medium data. The inquiry will evaluate the sensor data and the prediction's accuracy. To increase precision, sensor data should avoid epidemic conditions. The scheduling of janitors is based on many goals, including flexible employment scheduling.

II. LITERATURE REVIEW

Linked things are a key part of the actual world thanks to the IoT. Smart home systems require a whole system, which incorporates cloud-based data analysis and storage, end-user apps, middleware, and device connection [15]. The author identifies the components of any IoT as client devices, a cloud backend with a database, analytic apps, and user applications [16]. Figure 1 shows how the IoT is created utilizing a waterfall technique requiring knowledge, design, and creation. This comprehensive view of IoT components gives us options when comparing systems. First, several groups analyzed the system. These organizations outlined the system's needs, goals, and policies. Next, the project flow

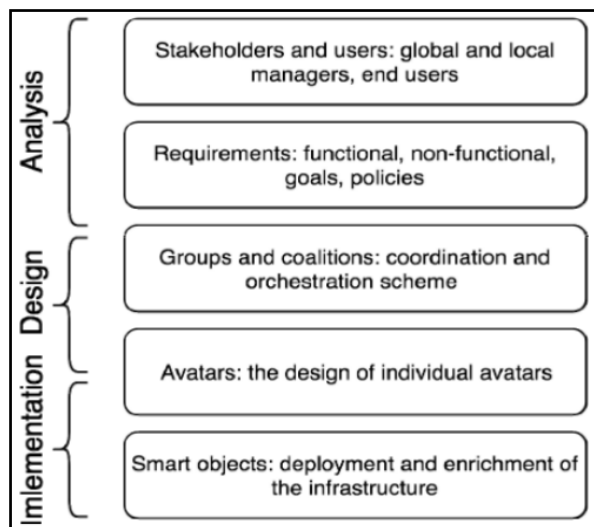


FIGURE 1. An IoT system concept.

and personnel participating in each component are designed. Next, implement the system.

New technology can monitor and forecast disease, improve output, decrease costs, and notify management. IoT monitoring technologies outperform traditional approaches, according to researches. The author suggested monitoring vital indicators with IoT [17]. Case studies monitored football players’ pulse rates during games. The proposed system monitored participants’ heart rhythms to predict injuries and untimely death. Another article proposed an IoT-based agriculture monitoring system [18]. The approach monitored citrus soil moisture and nutrients to determine fertilization and irrigation. Case studies showed that the approach improved citrus production, reduced labor costs, and reduced chemical fertilizer pollution. The next article proposed centralized gas level and leakage detection in hazardous conditions [19]. Wireless sensors captured the data. Remote servers offered the management of environmental sensor data via a user interface. The proposed technology would alert of crucial incidents. Real-time building site monitoring was suggested using a wireless sensor network and information modeling [20]. Wireless sensor nodes sent dangerous gas levels and ambient parameters like temperature and humidity to a remote server. The system warned of abnormalities. A case study showed that a proposed solution improved workplace safety and helped management make real-time choices. Figure 2 displays the gas area IoT system architecture.

Current studies can present real-time IoT sensor data to monitor environmental variables in a specific area. Smart buildings and healthcare require IoT-based sensors. Many studies have demonstrated that Internet of Things sensors can considerably improve system performance. The author advised monitoring smart facilities with IoT sensors [21]. Before building, simulations tested the system’s functionality. The study found that various IoT-based sensors

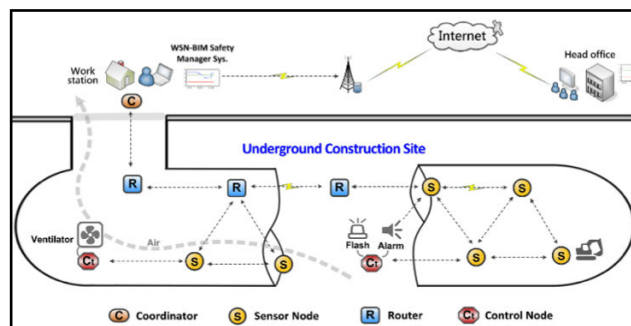


FIGURE 2. The idea behind the system for detecting dangerous gases.

increase smart building monitoring. The proposed technique should improve energy efficiency and green smart buildings.

The author demonstrated an IoT-based radon gas sensor [22]. The technology could warn of unsafe radon gas levels. The developed methodology could alert and begin pre-programmed steps when radon gas reached a predefined level. Another author proposed a portable indoor tracking system with sensors for carbon dioxide, carbon monoxide, chlorine, sulfur dioxide, nitrogen dioxide, humidity, and temperature [23]. Raspberry pi gateways processed sensor data. Unlike temperature and humidity, this suggested system tracked six gases during the test. Smart toilets in IoT systems have ultrasonic and infrared sensors that identify humans and their distance. The author showed that the smart toilet system can replace expensive Internet of Things (IoT) sensors.

An IoT-based healthcare system was created to detect and stop chikungunya [24]. Medical history, geography, and weather data determined status. The findings showed that the suggested technology can identify sick people and alert nearby governments and clinics to prevent outbreaks. A “smart house” extends beyond technology and convenience. According to the article proposed automated light control and keyless access save electricity [25]. Water flow sensors and smart meters reduce water use by linking to IP cameras and motion sensors. The author says smart house hacking is a drawback. Figure 3 shows an advanced urinal flushing solution [26]. Geological departments must manage water. Most public restrooms flush automatically. This device conserves water. This reduces cross-infection when flushing. This method can only track bowl cleaning frequency. The smart toilet system should cover the bowl and environment.

The toilet is one of the most dangerous places, especially for seniors, according to data [27]. Older people are more likely to fall and slip in the restroom, which can lead to serious health issues. Despite redesigning the shower, tub, floor, and toilet, bathroom injuries have remained high. Public restrooms need work. Public bathrooms are available nationwide, but their poor conditions deter many from using them. Due to neglect, public restrooms spread disease. Public restrooms can be used by diseased persons without proper

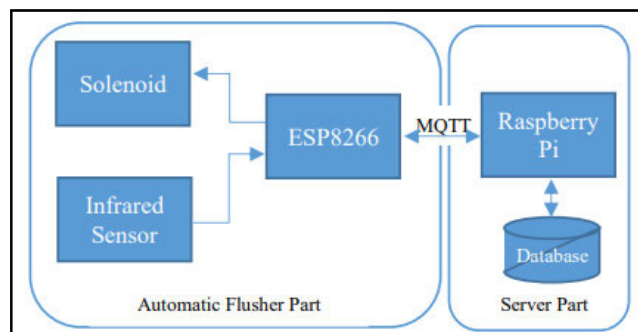


FIGURE 3. Overview system for smart urinal flusher.

sanitization. Use it at your own risk of infecting others. Thus, dirty public restrooms propagate sickness. The author argues that disease prevention is better than treatment [28].

The author says health monitoring with technology is rare [29]. They demonstrated long-term health monitoring software and hardware. Easy-to-use system. A calorimetric assay analyses motion and pressure sensors and automatically display the results. Urine analysis strips could assess urine flow rate. Deep learning and computer vision monitored health. Another author wants to remove public microorganisms [30]. Smart technology in public restrooms prevents bacteria transmission. Scanners were used to compare toilet and seated photos. It indicates cleanliness. If it's dirty, they'll tell you to flush more. Workers are informed of workplace risks. Always smell the toilet. The next author says a country's economy depends on its cleanliness. IoT-based toilet cleaning was their idea [31]. Sensors notify managers to clean filthy toilets. Odor and turbidity sensors can assess toilet cleanliness. Another article sought to raise awareness of personal hygiene in daily life [32]. They used IoT to clean the restrooms. Ammonia odor sensors detected restroom activity. The next article sought to emphasize how public bathrooms prevent disease and improve health [33]. They begged parliament to mandate public bathrooms in municipal governments. Clean public restrooms were also stressed to prevent illness spread.

Maintaining a building's maintenance system prevents issues. According to the article, a printing system's dependability, availability, and maintenance depend on its maintenance program [4]. This maintenance system won't affect output. It predicts and prevents equipment failure. Equipment has a lifecycle, whether short or long. Bathrooms have a flush, pipe, sink, and light. Smart toilet sensors. IoT should monitor this. Internet-enabled predictive maintenance sensors can detect faults before they occur.

Physical, knowledge and data-driven prediction model approaches exist. Long-term operating and maintenance experiences form the knowledge model. Rules, facts, or examples may represent these experiences [34]. It may be difficult to acquire correct data for knowledge-based models to anticipate. Data-driven methods have grown in popularity. It uses massive technological system data. It can be statistical,

stochastic, or machine learning [35]. Data-driven models must manage uncertainty. Physics-based models enable computational simulations of degradation, although many physics events remain unexplained. Despite environmental influences like temperature, pressure, and others, the result is accurate and complete [36].

Machine learning algorithms train predictive maintenance data. Algorithms vary. This project involves time series analysis methods. Use regression analysis for maintenance data. This method describes multiple factors. Linear regression can predict aero-material consumption [37], and student psychomotor domain [38], and generalize multivariate regression models utilizing fMRI data [39]. No application uses time series analysis. Time-series analysis forecasts equipment and process breakdowns.

This study provided a simple predictive maintenance method for industrial machinery [40]. Arduino devices automate, network, and collect data. This method implements threshold alerts for two automatic polishing and sanding equipment operating parameters. These parameters consider both data sets to determine the minimum value needed to trigger an alert. Predictive maintenance seeks to identify early warning signs and analyze historical data.

The author developed predictive genetic ion implanter maintenance using various classifiers [41]. The model predicts unexpected breakage and unexploited resource lifespan. They used kNN and SVM algorithms. Both algorithms classify. This model simply counts tool lifetime and usage. No notice is expected if unanticipated circumstances arise. Figure 4 illustrates predictive maintenance utilizing multiple classifiers.

The author suggested an autoregressive integrated moving average (ARIMA) model predict slitting machine sensor failures and quality problems [42]. This technique could not forecast the remaining useful life (RUL) in data-specific settings, according to the author. RUL is crucial since it can keep the system running during repairs. This model predicts production cycle parameters, which are used to categorize data in a supervised model. Logarithmic transformation reduces time series data variation.

The author used a hybrid prediction model to evaluate IoT sensor performance [43]. This model tested IoT sensors. Using eight datasets from an accelerometer, gyroscope, temperature, and humidity sensors, we predicted this experiment's findings. IG is calculated from all dataset attributes. Apache Storm determines if the process is normal or aberrant. The author ignores sensor time series data. However, it cannot predict its future.

The Long Short-Term Memory (LSTM) and XGBoost algorithms considered long-term characteristics and COVID-19 pandemic data [44]. Despite being more accurate than the XGBoost model, the LSTM model has a very small sample size. The LSTM model's MAE, RMSE, and MAPE are reduced as a result. LSTM-based and noise-layered convolutional networks were presented [45]. Noise cannot be accurately managed, however, RMSE can be improved

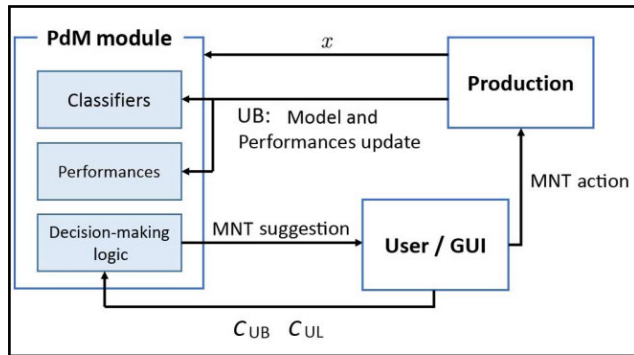


FIGURE 4. Multiple classifiers for PdM overview.

for various mobile window lengths. Next, the author suggests using the Auto Regressive Integrated Moving Average (ARIMA) model to estimate the power connector system's remaining useful life [46]. When unexpected changes occur, performance may fall, but prediction accuracy is increased. The following author suggests utilizing the ARIMA (1,1,0) model with the Nonlinear Auto Regressive (NAR) model to anticipate COVID-19 instances, which average 1500 each day [47]. ARIMA's p , d , and q values take longer to calculate than NAR's, but it's more accurate and consistent with historical patterns. The author recommends logistic regression models that use temperature, stress, and strain sensors to predict jet engine blade failure [48]. Although it can enhance accuracy by 87% in real-world applications, the approach fails when used to vast amounts of data. Logistic regression, extreme gradient boost, and random forest were suggested to forecast machine downtime [49]. The receiver's operational characteristics show its superiority, even though the procedure is slower due to the huge number of variables to modify.

Operating System (OS) job scheduling allocates system resources to many tasks. The system chooses CPU-waiting jobs from queues and sets their time limits. This timetable ensures that all chores are performed on time.

Parallel machine job scheduling involves assigning tasks to available machines and determining their processing order. Moore's algorithm is the task scheduling standard. It maximizes late tasks at one site to earn revenue. The author adapted this approach for m machines [50]. The janitor cleans each floor's toilets using scheduling. Office and academic buildings have distinct human presence patterns. The smart toilet system is used in corporate buildings and universities. Smart houses schedule appliances to optimize energy use and load balancing [51]. The author decides between summer weekdays, summer weekends, winter weekdays, and winter weekends. They solve grid utility, demand react aggregator, and customer MOPs using improved enhanced differential evolution (iEDE).

Due to their efficiency in solving complex issues, meta-heuristics have grown in favor. We use Ant Colony Optimization (ACO), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and League Championship

Algorithm (LCA) to study several scheduling strategies. ACO techniques can solve discrete optimization issues that need elucidating pathways to targets. It has solved the traveling salesman problem, multidimensional knapsack problem, job shop scheduling problem, quadratic assignment problem, grid and cloud task scheduling, and many other challenges. ACO solves problems by mapping the ant system to the issue. The author used adaptive pheromone values to schedule grid jobs [52]. Evaporation is monitored and cannot drop below zero.

PSO uses local and global search strategies to balance investigating new areas and developing old ones. PSO's appeal is due to its ease and affordability in a range of applications. PSO scheduling research has suggested many options. The author suggested combining the PSO with Gravitational Emulation Local Search (GELS) algorithms for grid computing autonomous job scheduling [53]. This maximized efficiency. By preventing local optima, the local search algorithm GELS improves PSO results. GELS compares PSO findings to find the optimum solution instead of randomly scanning the space. PSO-GELS reduces makespan by 29.2% compared to Simulated Annealing (SA) in a scenario with 5000 jobs and 30 resources. In this paradigm, the PSO cannot validate multiple-task scheduling. The author presented the League Championship Method, a novel global optimization meta-heuristic algorithm (LCA) [54]. It follows sports association competitions between teams. LCA has been used to solve several optimization problems, including the traveling salesman problem, reactive power dispatch difficulty, workshop scheduling issue, electromagnetic device optimization challenge, and cloud job scheduling issue. This method helped to optimize cloud schedules [55]. The authors wanted to reduce job completion time in an Infrastructure as a Service (IaaS) cloud environment. It outperforms FCFS, LJF, and BEF, according to the findings (BEF).

Research has employed numerous representations for GA scheduling solutions. Direct, permutation-based, and tree representations are the most commonly used representations. Chromosomes (ch) are n -dimensional vectors that specify the resource on which job I am planning. Chromosomal maps represent this. The use of direct representation was used in the article [56], [57]. A two-dimensional vector represents a chromosome in permutation-based representation. One dimension represents resources, while the other indicates job order.

III. METHOD

This section covers three main topics: system design, software and hardware requirements for constructing and testing algorithms, and algorithm analysis. We will also briefly explain the algorithm evaluation technique.

A. THE ARCHITECTURE

Figure 5 shows the system design. This research project's system separated sensing and analytical modules. The sensing module was the main tool for monitoring and analyzing test

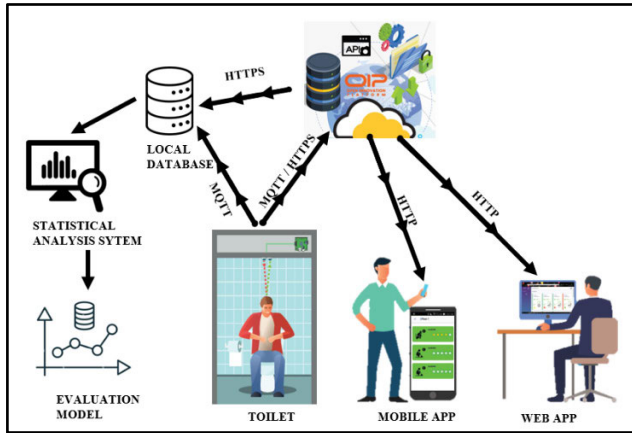


FIGURE 5. An Overview of the Design of the System Architecture.

platform operations. MQTT and HTTP can move data. Data is saved to a local database for filtering, statistical analysis, and evaluative modeling. Telekom Malaysia’s Open Innovation Platform (OIP) stores more data. Telekom Malaysia Research and Development intends to maximize microservices, APIs, the Internet of Things (IoT), and smart services with OIP. OIP simplifies device connection, data storage, processing, analysis, and data protection, making it easier to develop an end-to-end Internet of Things solution.

Figure 6 shows each operation’s architecture. Hardware implementation will start this project. The restroom will use the sensor. Humidity, temperature, and infrared sensors are offered. System microcomputers are Raspberry Pi and Arduino TTGO. Wireless connectivity, integrated sensors, online storage, and a strong CPU are needed for the control kernel to work. In many trials, Raspberry Pi-based micro-controllers have met system control kernel requirements. Simply replacing a Raspberry Pi’s memory card changes its setup. Small businesses may benefit from using Raspberry Pi for real-world applications due to its inexpensive cost. Troubleshooting and data verification after hardware installation. This will verify data accuracy and identify potential issues during system operation. Then, the data will be entered into the database and verified. Predictions and timetables will use the data. Pre-processes sensor data patterns. After that, we decided to exploit the degrading feature to develop an accurate RUL forecasting model at diagnosis. It will pre-process sensor and fitness calculation data for scheduling. Selection, crossover, mutation, and others make up the scheduling algorithm. Data analysis will decide model performance.

B. SOFTWARE ENVIRONMENT

This system requires three software programs to work. First, Anaconda Navigator 1.18 is used. This software uses Python to create machine-learning applications. It runs Spyder, Jupyter Notebook, and RStudio, among others. Programming the board requires the Arduino IDE, version 1.8.19. Open innovation platform will save telemetry data (OIP). OIP

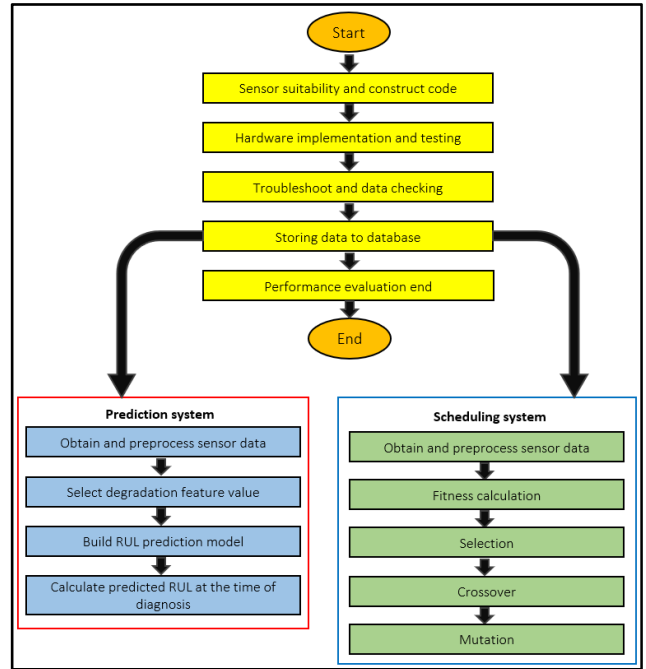


FIGURE 6. Architectural details of the process.

TABLE 1. Software description in tabular form.

Software	Specification
Operating System	Microsoft Windows 11 (64-bit)
Machine learning tool	Anaconda Navigator v1.18
Programming language	Python and C programming
Database platform	Open Innovation Platform

manages data, services, and IoT devices. Table 1 lists the system’s software requirements.

C. HARDWARE ENVIRONMENT

Raspberry Pi and ESP32 boards helped build the sensor module. A strong CPU was needed to pre-process data for analysis in the proposed system’s control kernel, which required Wi-Fi connectivity, the ability to integrate many sensors, and online data storage. After reviewing the many prototyping boards available, we selected the finest components for these needs. The Raspberry Pi and TTGO microcontrollers can meet the requirements as control kernels, according to studies. Thus, the project’s control kernel was the Raspberry Pi and TTGO OS. Raspberry Pi, a single-board computer (SOC), runs Linux and may be upgraded by swapping the memory card. This expedites upgrades. Like a computer, it can multi-task. Raspberry Pi applications can leverage networking, data transport, databases, and web servers. Secure Shell allows remote access. The Raspberry Pi cannot do ADC. The Raspberry Pi with TTGO combined control kernel for real-world applications is affordable for small and micro-businesses. Depending on the information needed, different sensors are used. It sends sensor data from wired or wireless networks to

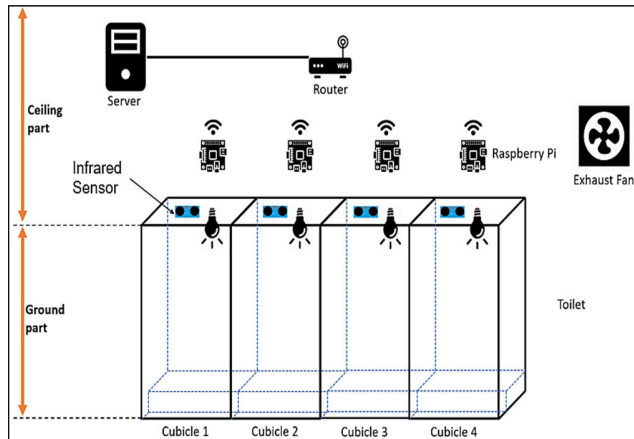


FIGURE 7. Workflow hardware implementation system.

a cloud database for cloud analysis. Figure 7 shows how to construct the hardware system, and Figure 8 shows a toilet floor plan.

Figure 7 shows three sensors with wire cables linked to the Raspberry Pi or TTGO. The database receives data from both microcontrollers over Wi-Fi. Ethernet connects the router and server. Figure 8 shows the Faculty of Computing and Informatics toilet floor plan (FCI). Despite the enclosure's infrared sensor, a light bulb signal is. The sensor is inside the housing to avoid confusion. The Raspberry Pi, router, and exhaust fan are in the overhead cabinet above the ceiling level.

Microcontrollers are single-chip computers having processor core, memory, and programmable input/output peripherals. For embedded systems. Automated devices employ microcontrollers. Many smart gadgets and control systems exist. The Raspberry Pi Foundation in the UK developed this credit-card-sized single-board computer. Raspberry Pi offers several programming opportunities. This study used a Raspberry Pi 4 Model B. General-purpose input-output (GPIO) is one of Raspberry Pi's most useful capabilities. Board I/O pins input and output signals. Quad-core ARM BCM2711 processors use four gigabytes of RAM. An ethernet jack and four USB ports link keyboards, mice, cameras, and more. This device runs Linux OS. Linux users enjoy a huge selection of libraries and apps. Python, C++, Java, and many others are supported. I2C, UART, SPI, wireless, and wired LAN interfaces are available.

TTGO ESP32 provides a complete development environment. It has over 30 I/O pins, Wi-Fi, Bluetooth Low Energy, and a microcontroller. Low-cost, low-power microcontrollers are hard to find. The board can be powered by either a single-cell lithium-polymer battery or a 5V USB connection, and the voltage of its working signal can range anywhere from 2.2 V to 3.0 V. TTGO ESP32 I/O pins can only take 3.3 volts. TTGO ESP32 microcontroller programming using Arduino software integrated development environment. Arduino IDE programs can be uploaded to the board through USB. Arduino's Sketches language is just C and C++.

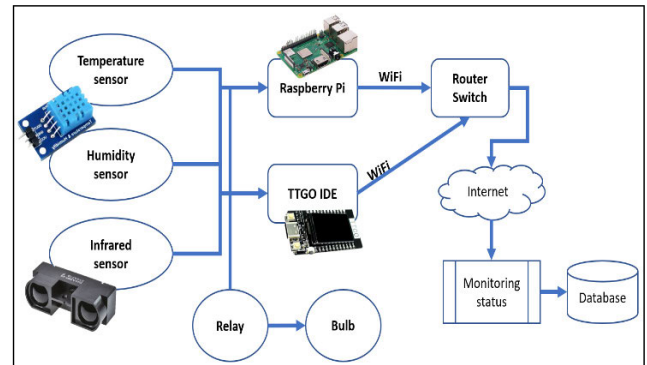


FIGURE 8. Floor plan for toilet implementation.

This study included temperature, humidity, and infrared radiation sensors. Sharp, the industry leader, makes most infrared detectors and rangers. Sharp Infrared Detectors and Rangers, available in many combinations, can measure distances precisely. Infrared distance sensors generate an analog signal proportional to the sensor's distance from the thing being measured. The datasheet states that the SHARP GP2Y0A710K0F output voltage ranges from 2.5 V at 100 cm to 1.4 V at 500 cm. Distance detection allows this range. Table 2 lists Sharp sensor specs. The ultrasonic sensor is superior to the infrared sensor [58]. The author details the monitoring mechanism. Because a human's body isn't flat on the area being scanned, ultrasonic sensors are better at identifying them. Increased detection distance reduces infrared sensor precision.

D. CODE IMPLEMENTATION AND EXECUTION

The Raspberry Pi's standard Python editor, IDLE (Integrated Development and Learning Environment), helped us program the gadget. An interpreter lets you test each instruction to validate the code. This site uses Python 3.5.3. Figure 9 depicts the method used to count restroom users. The system operates from 7 am until 10 pm. We chose this time because it matches the school's kids' and teachers' working hours. If the time limit is surpassed, the system won't run. A user can be "enter," "stay," "leave," or "nobody" depending on whether motion detection detects movement.

The program's initial iteration will check the pre-distance. If the distance is less than the limit, motion detection 1 will not be done. After this, motion detection 2 requires maintaining stationary for a certain time before moving on. To trigger motion detection level 3, the final and most essential stage, the distance must be much larger than the permissible margin of error. Moving on to the fourth and last motion detection, which will result in no movement. The word "nobody" indicates a distance reading error. After that, the gain will be calculated from scratch for the next iteration. The database will contain all available information. If there is a difficulty with the internet connection, it will try to reconnect without pausing or interrupting the program. The data will be

TABLE 2. Sharp infrared sensor specifications.

Type	Specification
Operating voltage	4.5 to 5.5 V
Operating current	30 mA
Measuring range	100 to 550 cm
Output type	Analog
Dimension	58 x 17.6 x 22.5 mm
Mounting hole	4.2 mm

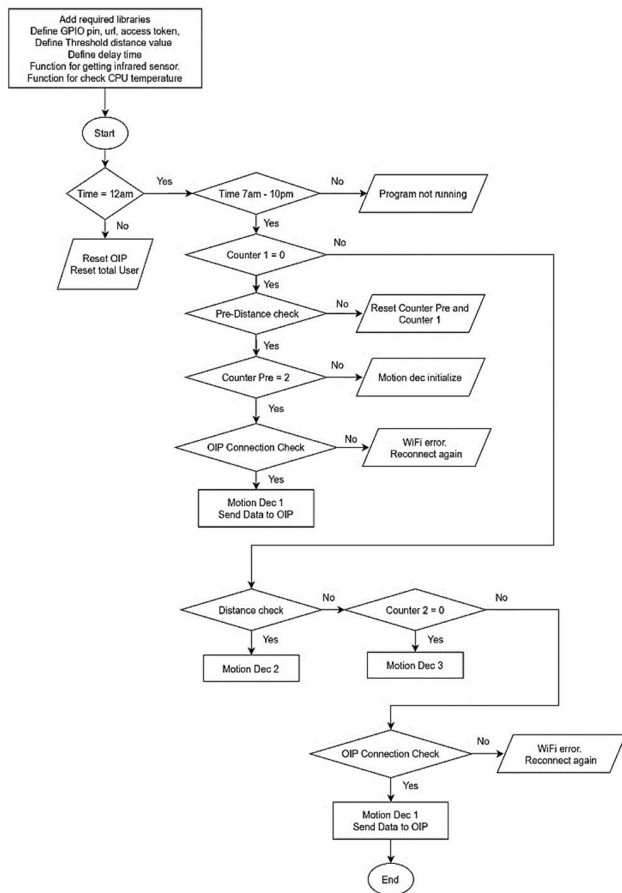


FIGURE 9. Programming Flowcharts for the Raspberry Pi.

re-entered into the database once the internet connection is stable again.

Different toilets have different bathroom detection distances. Males and females are measured by average human height. 50–200 cm. Most people defecate sitting or squatting. Defecating at 35 degrees is best. Each cubicle, squatting toilet, and bowl toilet has infrared sensors on top. Figure 10 shows that a sensor will point straight toward the cubicle’s occupant. The sensor angle adjusts to x_1 and x_2 .

The dataset was collected from the Faculty of Computing and Informatics (FCI) at Multimedia University (MMU) and the vaccination center at the Dewan Tun Canselor MMU. The team completed the data capture layer task for the proposed work. Sensors collected contextual data including

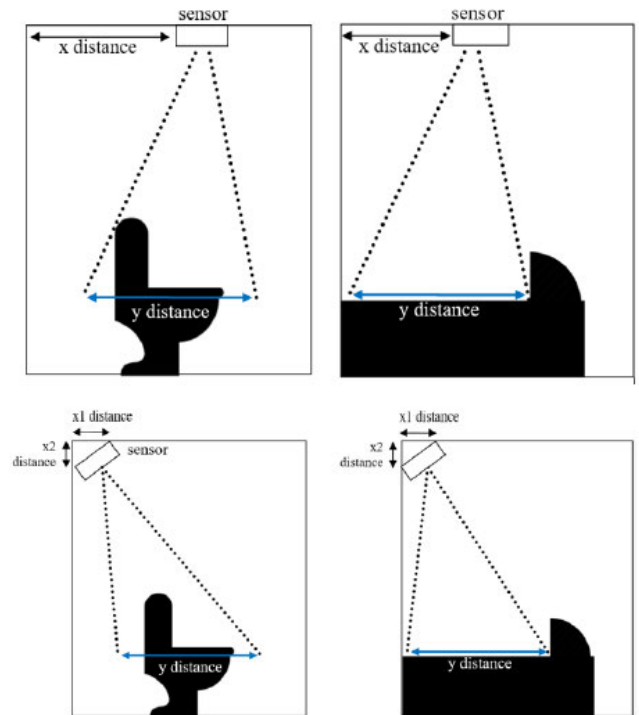


FIGURE 10. Inside the cubicle, where the sensor is located.

motion detection, distance, temperature, humidity, total user, ram, and so on. Passive Infrared (PIR) sensors were used to determine user occupancy in motion detection states 1–4. The experiment collected data from September 2019 through December 2021. Each floor of the FCI building included seven toilet facilities, three male and four female.

E. DATA ACQUISITION LAYER

Line graphs better showed user statistics. The x-axis shows the date, while the y-axis shows the number of individuals in each cubicle. Since university buildings are busy with classes, the total number of persons using them increases during the semester but decreases between semesters. From this vantage point, we can see which cubicles visitors use most for their toilet type (squatting or sitting). Sitting toilets have more users than squatting toilets, and most of them won’t utilize the last cubicle. Figure 11 shows that cubicle 3 has fewer individuals than cubicles 1 and 2.

Pre-processing removed data outliers. Data noise was assumed since data recording is vulnerable to numerous external factors. Data recording prompted these hypotheses. Like outliers, this incident had several possible causes, such as sensor malfunction, measurement error, etc. Moving averages, loess, lowess, Rloess, RLowess, and Savitsky-Golay smoothing filters help eliminate outliers. Use Savitsky-Golay. In this study, the moving average filter was employed to make enormous volumes of data easier to deal with, as previous researchers have done [59].

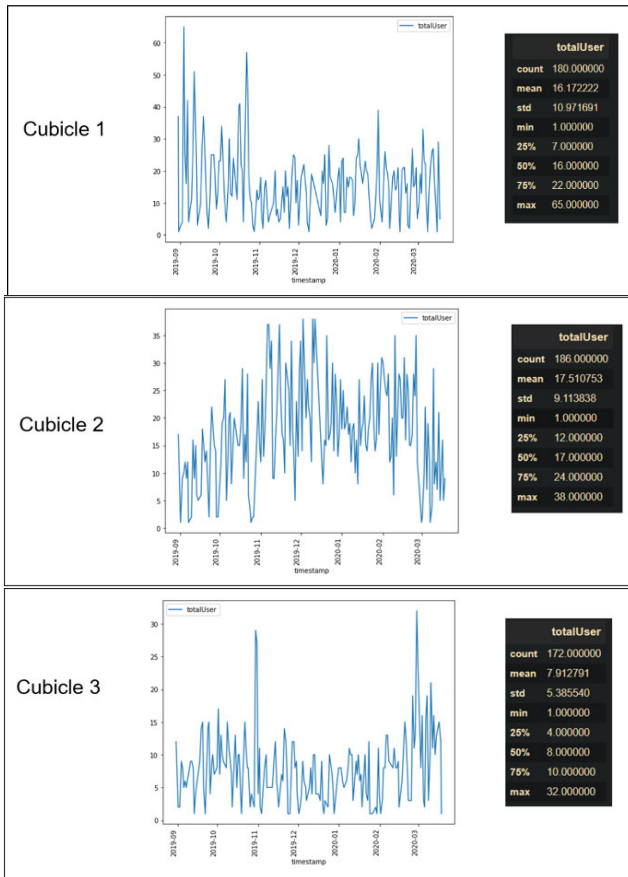


FIGURE 11. A line diagram for IoT data.

Inconsistent data and large samples require normalization. This reduced the information needed to make a prognosis and schedule. The machine learning models used normalized data with values in the same range. This ensures that weights and biases converge gradually. Normalizing whole sample data for machine learning algorithm models enhanced predictions and training.

Below are instructions for implementing this method. I'll explain the suggested method's operation after this. ARIGA combines ARIMA and Genetic Algorithm for predictive maintenance and scheduling. The algorithm will improve outcomes by combining both.

F. PREDICTION METHOD WORKFLOW

Predictive maintenance follows data collection. Pre-processing follows database export. Filtering data to predict only important facts. Step 1 of Figure 12 uses IoT monitoring sensor data to determine the degradation feature value at the diagnostic time. Next, feature value trends determine the predicted failure time distribution (step 2). The distribution can be calculated using the histogram of the remaining useful life based on many similar failure events. Finally, the distribution estimates maintenance costs for each projected future maintenance (step 3). Considering the distribution's uncertainty, the maintenance time is the projected time before an unexpected failure.

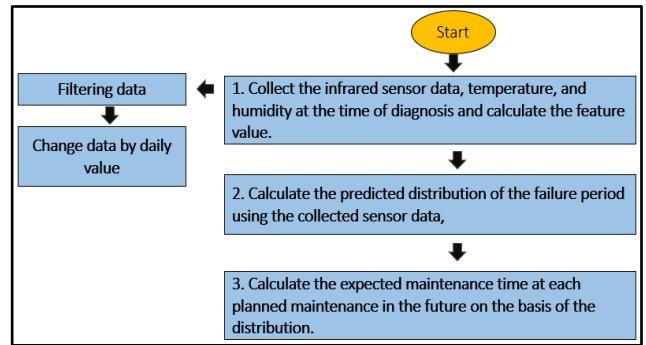


FIGURE 12. Diagrammatic representation of predictive maintenance strategies.

ARIMA model predictions combine recent data with the long-term historical trend. The ARIMA model predicts this integration because it intuitively represents numerous practical time series. Time-domain models like ARIMA are used to fit and forecast temporally correlated time series. ARIMA models can describe stationary or nonstationary time series data. Seasonal time series are nonstationary. This is because stationary time series are unaffected by observation timing. Trends and seasonality are nonstationary time series. Stationary time series have a cyclical pattern but no trend or seasonality. ARIMA is built on AR, MA, and ARMA models.

ARIMA models estimate the next time series degradation step by linearly combining the present value, previous values, nonseasonal changes, and lagged prediction errors. Differences eliminate data non-stationarity. Stationary time series are independent of observation time. MA combines regression mistakes linearly, while AR reveals the regression variable depending on past values.

Prediction algorithms are assessed based on several factors. The recommended model's performance evaluation layer assesses the model's accuracy using MAE, RMSE, and MAPE. Model mistakes are these measures. The mean squared error (MSE) is used to minimize the error range. MAPE measures the prediction difference as a percentage of the targeted data, while RMSE measures the error between actual and anticipated data.

G. SCHEDULING METHOD WORKFLOW

Figure 13 shows the scheduling flowchart. After being exported from the IoT monitoring system, the data will be loaded into the scheduling system and time-processed. Based on your hourly usage pattern, determine peak and off-peak times. A SimPy simulation showed daily janitorial tasks. SimPy was written in Python. A discrete-event simulation library is SimPy. The simulator considers conventional cleaning timings, minimum and maximum wait times, cleaning durations, and janitor capacity. Modeling lets us determine a cleaner's time slots. After programming a 5-day, 8-hour workweek, the simulation will conclude.

We simulated the user's steps using Monte Carlo. Monte Carlo simulations generate multiple possible outcomes by

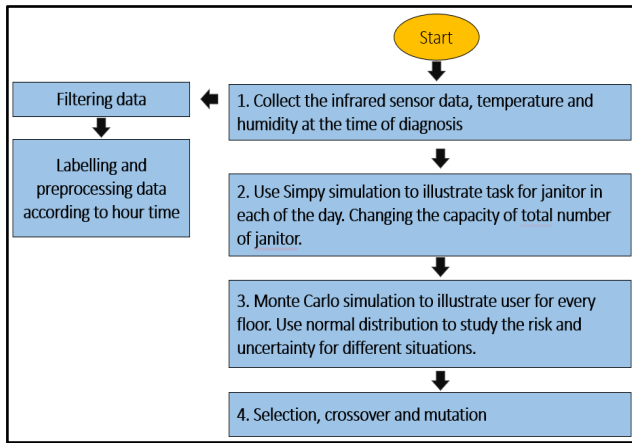


FIGURE 13. A diagram depicting a scheduling algorithm.

using random numbers as inputs into a mathematical model. Probability-based solutions to difficult mathematics problems. It studies risk and uncertainty in many contexts in academia. Risk and uncertainty assessments benefit greatly from the normal distribution. The simulation data follow a normal distribution;

- Standard deviation = 0.4
- Number of repetitions = 12
- Number of simulations = 253
- User target value = 0 to 25
- User probability = 0 to 1

Scheduling involves user input, data point relationships, system constraints, and genetic algorithm (GA) optimization. Despite not being the optimal choice for all topics, genetic algorithms (GAs) can sometimes provide a reasonable response. Despite being dubbed genetic algorithms, GA can solve even the hardest problems, unlike traditional algorithms, which solve issues step-by-step. GA optimizes problem-solving. If genetic operators execute ideally, GA can aid with functions specified across complex, discrete structures. GA requires this.

Inputs include the number of floors in each building, the number of janitors, the working days, and the starting time and timeslot. Systems can have harsh and soft limitations. However, strict boundaries are unchangeable. The scheduling engine always follows severe limitations. Soft limitations are negotiable. Soft constraints are less rigid. The scheduling engine will try to follow your soft limitations, but it may stray if necessary. The study limits janitors to one shift each day and one per level. Due to such constraints, the janitor cannot take a vacation during business hours and must evenly space out their shifts.

This study uses evolutionary computation to find a schedule within a suitable timeframe. A genetic algorithm uses natural selection and population genetics to search probabilistically. Figure 14 shows a quick genetic algorithm flow. Initial schedules are generated. The crossover and mutation operations merge two schedules into one, and then the schedule

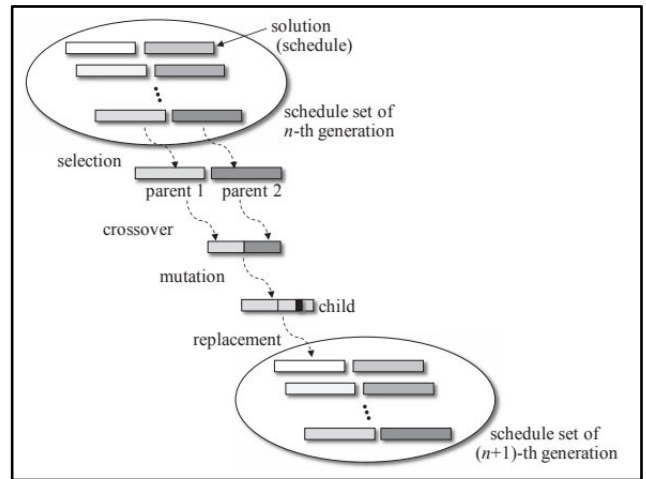


FIGURE 14. Scheduling step of a genetic algorithm.

set is evolved by replacing a schedule in the old set with the newly formed schedule. Repeat until the schedule set converges.

The approach presents a timeline as a matrix. The matrix rows represent the overall number of users, floors, and hours in the day, while the columns represent the real schedule. Each matrix cell might be 0 or 1. 0 means no cleaning, and 1 means yes. This investigation’s encoding gave each floor covering kind its row. To clarify, the floor and total users are treated separately and given sufficient time to perform properly. This method simplifies encoding over defining numerous values for each matrix element. Two scheduling conditions should be considered when dividing jobs among available periods. Interruptible and non-interruptible scheduling. We use a finer time slot to shift an interruptible activity onto several non-interruptible jobs.

Selecting two existing schedules will create a new generation of schedules. This analysis uses the normal operating method, which yields the best schedule with four times the likelihood of the worst schedule. The pointer selects and creates the parenting schedule for the current time slot. Click the time slot. This is a probabilistic rule that favors better scheduling. We’re using a probabilistic technique to avoid unfair classifications. Deterministic selection of the best schedules may quickly dominate the scheduling set, causing early convergence to a local optimum. We risk becoming dependent on their schedules.

After parental unit selection, the population undergoes crossover. Crossover recommends combining the productive parts of both parents to conceive a child, hoping to simplify routines. Crossovers create new generations. This includes randomly identifying a crossover location and swapping genes from both parents. Slicing a matrix column generates the crossing point.

Mutation operations, which change the developed child for variety, should be integrated into crossover operations, but this paper does not discuss mutation operations. The classic one-point crossover approach may not be practical since it

does not follow activity time limits. A scheduling unit's sliding window hitting the crossover point generates a similar problem. Thus, we must adapt so that the offspring inherits the scheduling unit from the first parent and the second parent fills the open spots. If there are empty spots, it will pick a random place in the sliding window and fill them with data.

This section covers IoT architecture and overview. This chapter also discusses the predictive maintenance and scheduling strategy, research implementation, and approach formulation and evaluation. This investigation dissects experimental software and hardware. From hardware and data collection through scheduling, the proposed method is detailed.

IV. RESULTS AND DISCUSSION

A. PREDICTION EVALUATION DESIGN AND PERFORMANCE

The ARIGA model is compared to the Long Short-Term Memory (LSTM) model to determine its efficacy. This section evaluates the RUL prediction approach using the restroom experiment's results. The factory data sheet accelerated deterioration time and was used to achieve experimental results. To evaluate a model's prediction ability, it must be tested on data not used to fit it. To compare models, it is usual to train a model using only some of the data (called "in-sample data") and then test its ability to reliably predict fresh data (called "out-of-sample data"). Thus, the sample data is split into a training set and an evaluation set. The training set estimates the model's parameters, while the testing set selects the most accurate model.

In-sample model fitting and selection are done on the first 600-time series observations. Time series data were split into a training dataset (70%) and a test dataset (30%) to appropriately evaluate models. The model evaluation used both datasets. We can use all observations. Because it outperformed the others in the testing sample, one model will represent all candidates in out-of-sample testing. All model comparisons use data outside the sample. In the example study, ARIGA models, an upgraded ARIMA model, outperform ARMA models. The AR term is always needed, and the MA term may improve AR component operations. Variable order is crucial when creating an ARIMA model. Time series make seasonal variations and non-stationarity obvious. The nonseasonal difference ($d = 1$) and seasonal difference ($D = 1$) abolished non-stationarity. The first-order difference can analyse the ACF and PACF plots. Figures 15 and 16 exhibit differential time series ACF and PACF plots. The ARIMA model's seasonal ACF plot indicated a large spike at lag 12 ($Q = 1$), however the PACF plot showed no such spike at lags 12 or 24 ($P = 0$). Seasonal model analysis employed the ACF plot.

Figures 17 and 18 show how this model predicts journey length. ARIMA models calculate using the same data set. The graph's non-linearity indicates users' daily activity levels. The value changes, which affects students enrolled at the school and increases usage during that time.

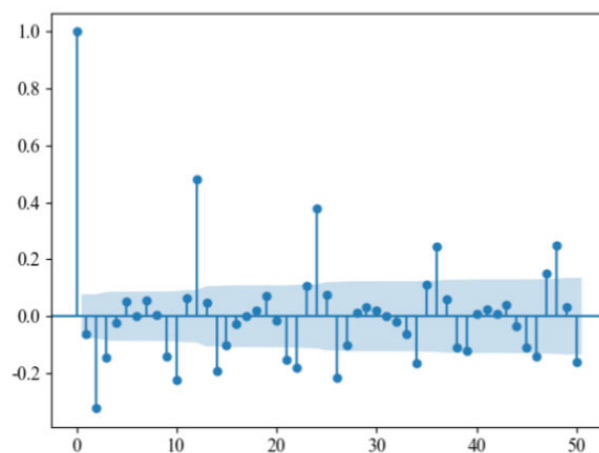


FIGURE 15. Autocorrelation function graph.

LSTM Recursive Neural Networks (RNNs) can store and learn from many observations. The multi-stage univariate prediction was used. First, 70% of the dataset is used for training and 30% for testing. To provide fair comparisons, the ARIMA algorithm is used. Use a fixed random number seed, such as 7. This ensures exact results replication. One algorithm function lets the LSTM model be constructed and trained. Fit LSTM develops and trains. The function requires three inputs: the size of the training dataset, the number of users (indicating the number of times a piece of the dataset is used to calibrate the model), and the number of neurons (the number of memory units or blocks). Compiling and parsing the network ensures that it follows Theano's mathematical notations and rules. When compiling your model, we will be asked to choose a loss function and optimization strategy. Figure 19 shows predictions from either model. ARIGA's graph is closer to the data than LSTM's, improving accuracy. This contributes to accuracy.

Table 3 shows the root-mean-squared error (RMSE) when the two models used only one prediction method. The ARIGA model has lower RMSE, MAE, and MAPE than the LSTM model. The ARIGA prediction model offers better fitting and stability. The entropy weight method first compares the information provided by each prediction methodology, then adjusts the relative value of each strategy based on the predicted day. As a result, the ARIGA prediction model, based on the entropy method, fully integrates information, uses the explicit and implicit information in each prediction model, balances the deviation of each prediction model, and avoids the concentrated influence of a large number of factors on prediction accuracy. Since predictions were more accurate, the suggested strategy is better and feasible.

B. SCHEDULING EVALUATION DESIGN AND PERFORMANCE

This section presents the performance evaluation findings to assess the scheduling system's effectiveness. An FCI toilet and vaccination center's IoT sensor data were used to create

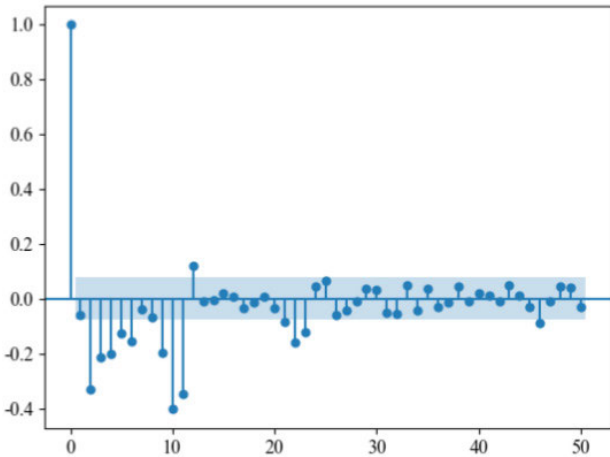


FIGURE 16. Partial autocorrelation function graph.

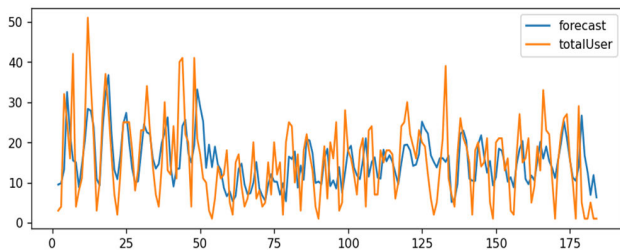


FIGURE 17. Real and forecasting ARIMA prediction graph.

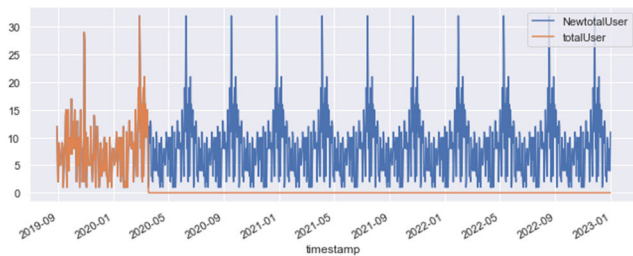


FIGURE 18. Actual and ARIMA prediction graph.

scheduling simulators. Figure 20 shows the janitor job schedule requirements and limits. The janitor must break the yellow boxes in the timetable. Weekly availability data determines the number of janitors, and they must be scheduled to clean different floors at different times. Janitors work from 7 am to 4 pm. Monday morning scenario: Janitor 1 cleans Floor 1 from 7 to 9 a.m., while Janitor 2 cleans Floor 2’s facilities at the same time.

SimPy simulates janitor shifts. A janitor cleans 1,000 square feet in one hour and thirty minutes. Each FCI floor features a 193.75-square-foot men’s and women’s restroom. Janitors take 20 minutes to clean each floor. Python-based SimPy simulates discrete events. Parallel processing models communications, customers, trucks, and airplanes. This SimPy simulation requires a specified time slot of eight hours, an interval time of one hour, a minimum waiting time of one, a maximum waiting time of two, a worker capacity of

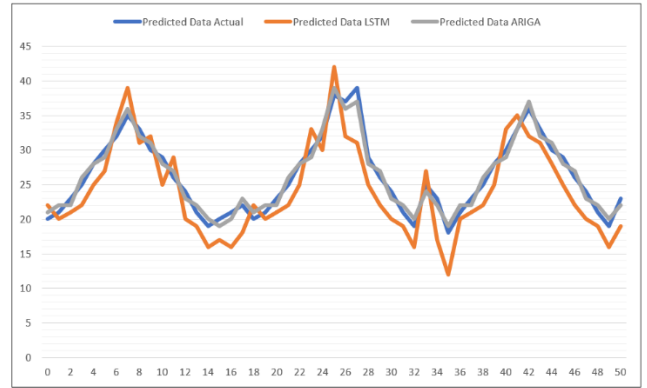


FIGURE 19. Graph of LSTM and ARIGA model.

TABLE 3. Measures used to assess the accuracy of forecasting models.

Method	RMSE	MAE	MAPE
LSTM model	3.09	2.43	4.16%
ARIGA model (7,1,1)	2.95	2.12	3.92%

one, and a cleaning time of twenty minutes. Figure 21 shows simulation findings. The output shows that one janitor can clean restrooms seven times a day.

Monte Carlo will simulate the user pattern next. Monte Carlo simulations can show how input changes affect outputs. Monte Carlo methods can address optimization and numerical integration. These algorithms cleverly draw samples from distribution to imitate system behavior. This chance-based method simplifies complex numerical issues. This field analyses risk and danger. This simulation uses the normal distribution since its parameters are accurate;

- Standard deviation = 0.4
- Number of repetitions = 12
- Number of simulations = 253 (total data for 1 month)
- User target value = 0 to 25
- User probability = 0 to 1

Figure 22 depicts users’ daily behavior by floor and hour in a histogram. The y-axis shows the number of times each toilet is used every day, while the x-axis shows the hours of the day. This area has seven compartments. The restaurant gets a lot of guests during lunchtime, when most people are on break. The user pattern on the third-floor graph does not alter between eleven and two. Figure 23 shows the planned user as a blue line and the actual user as a red line. Time is represented by x, and system users, both hypothetical and real, are represented by y. This Monte Carlo simulation shows that the population under examination can maintain its behavior across time. This development could aid numerous sectors, including user behavior analysis and theoretical physics.

Each floor had a varied amount of user patterns, therefore the research was done on four floors (ten, twenty, and thirty, respectively). Data settings for each parameter can be found

Time/Weekdays		Monday	Tuesday	Wednesday	Thursday	Friday
FLOOR 1	7:00 - 8:00	Janitor 1		Janitor 1		Janitor 1
	8:00 - 9:00		Janitor 2		Janitor 2	
	9:00 - 10:00	Janitor 1		Janitor 1		Janitor 1
	10:00 - 11:00					
	11:00 - 12:00	REST	REST	REST	REST	REST
	12:00 - 13:00	Janitor 2	Janitor 1	Janitor 2	Janitor 1	Janitor 2
	13:00 - 14:00					
	14:00 - 15:00					
FLOOR 2	7:00 - 8:00		Janitor 2		Janitor 2	
	8:00 - 9:00	Janitor 2	Janitor 1	Janitor 2	Janitor 1	Janitor 2
	9:00 - 10:00					
	10:00 - 11:00	Janitor 1		Janitor 1		Janitor 1
	11:00 - 12:00	REST	REST	REST	REST	REST
	12:00 - 13:00	Janitor 2		Janitor 2		Janitor 2
	13:00 - 14:00					
	14:00 - 15:00					
FLOOR 3	7:00 - 8:00	Janitor 2	Janitor 1	Janitor 2	Janitor 1	Janitor 2
	8:00 - 9:00					
	9:00 - 10:00					
	10:00 - 11:00	Janitor 1		Janitor 1		Janitor 1
	11:00 - 12:00	REST	REST	REST	REST	REST
	12:00 - 13:00	Janitor 1		Janitor 1		Janitor 1
	13:00 - 14:00		Janitor 2		Janitor 2	
	14:00 - 15:00					
FLOOR 4	7:00 - 8:00					
	8:00 - 9:00					
	9:00 - 10:00		Janitor 1		Janitor 1	
	10:00 - 11:00					
	11:00 - 12:00	REST	REST	REST	REST	REST
	12:00 - 13:00	Janitor 1		Janitor 1		Janitor 1
	13:00 - 14:00	Janitor 2		Janitor 2		Janitor 2
	14:00 - 15:00	Janitor 1		Janitor 1		Janitor 1
15:00 - 16:00	Janitor 1		Janitor 1		Janitor 1	

FIGURE 20. Here is a sample schedule for janitors, complete with available shifts.

Toilet Schedule

Time	User	Status
0	Time Slot 00:	Request Cleaning Toilet
0	Time Slot 00:	Cleaning in progress 0.000
6	Time Slot 00:	Cleaning Finished
61	Time Slot 01:	Request Cleaning Toilet
61	Time Slot 01:	Cleaning in progress 0.000
76	Time Slot 02:	Request Cleaning Toilet
77	Time Slot 02:	Delay after 1.087
84	Time Slot 01:	Cleaning Finished
210	Time Slot 03:	Request Cleaning Toilet
210	Time Slot 03:	Cleaning in progress 0.000
215	Time Slot 03:	Cleaning Finished
243	Time Slot 04:	Request Cleaning Toilet
243	Time Slot 04:	Cleaning in progress 0.000
247	Time Slot 04:	Cleaning Finished
285	Time Slot 05:	Request Cleaning Toilet
285	Time Slot 05:	Cleaning in progress 0.000
290	Time Slot 05:	Cleaning Finished
348	Time Slot 06:	Request Cleaning Toilet
348	Time Slot 06:	Cleaning in progress 0.000
348	Time Slot 06:	Cleaning Finished
401	Time Slot 07:	Request Cleaning Toilet
401	Time Slot 07:	Cleaning in progress 0.000
410	Time Slot 07:	Cleaning Finished

} 7 timeslot for 1 janitor

FIGURE 21. SimPy simulation result for janitor.

here. First, there is the total number of floors, which in this case is four, and the maximum number of persons per floor is thirty. The janitor’s identification number follows, followed by four random names as references. Next is cleaning time, divided into eight rows starting with C1 and ending with C8. C1 is the start of the workday, usually around 7 am, and C8 is the finish, usually around 4 pm. The number of users is calculated using sensor data from the Internet of Things.

In Figure 24 to 28, the janitor schedule is compared to the recently suggested ARIGA model, an improved genetic

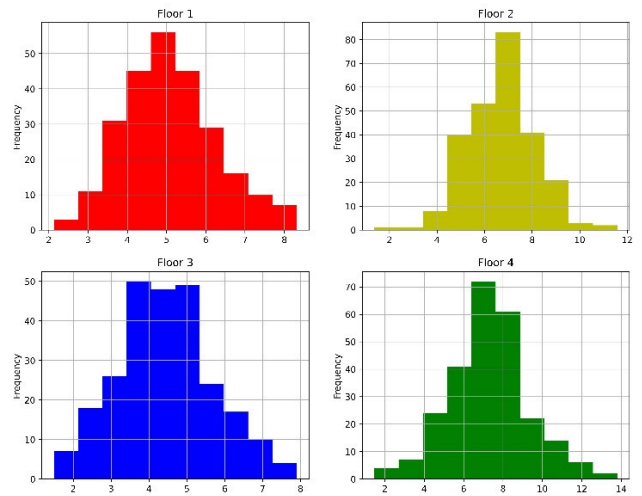


FIGURE 22. Histogram graph of user pattern with the selected hour.

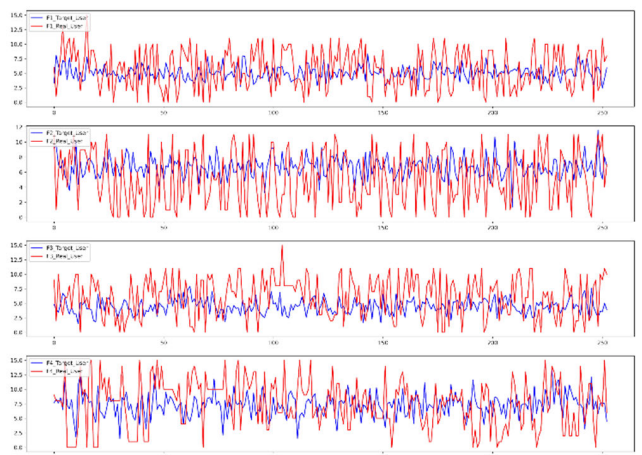


FIGURE 23. Line graph of real user with the targeted user.

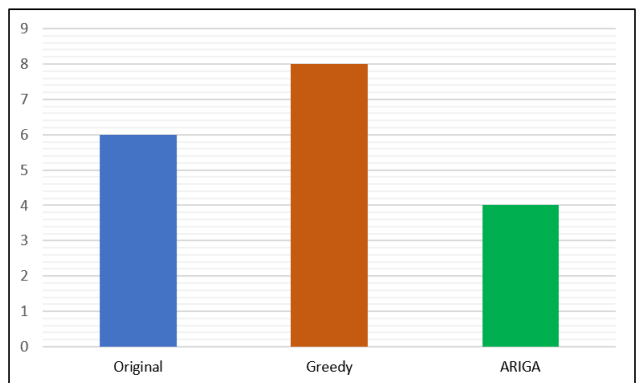


FIGURE 24. Cleaning demand = 10.

algorithm scheduling system. In the original system, all available time slots between the announcement date and the due date were chosen at random. The Greedy scheduler is also modelled for comparison. The Greedy scheduler gives each unit a cleaning time window.

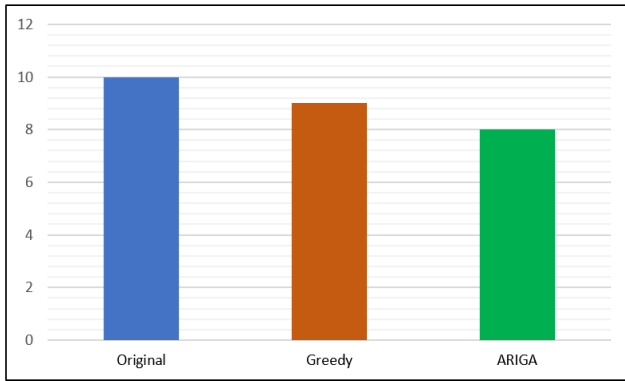


FIGURE 25. Cleaning demand = 20.

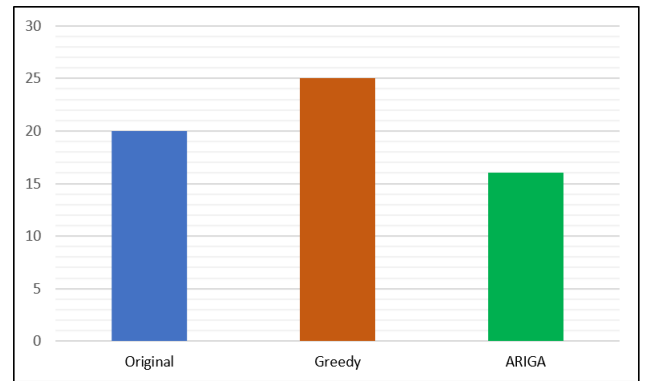


FIGURE 28. Cleaning demand = 50.

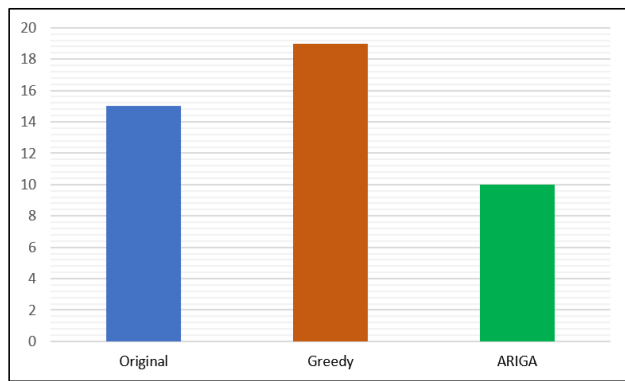


FIGURE 26. Cleaning demand = 30.

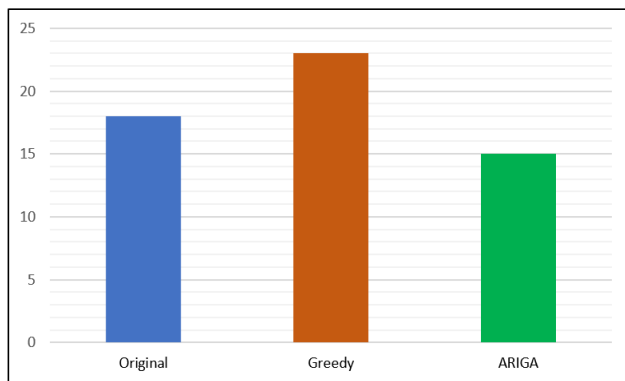


FIGURE 27. Cleaning demand = 40.

As shown in the figure, the proposed scheduling system performs better than the competition regardless of building size. The ARIGA model enhances performance by 24.7% on average and 15.0% overall compared to the baseline scheduling technique. In a high-rise structure with more than 50 cleaning demands, the proposed scheduling method performs better. The ARIGA model schedules most cleaning time demands for off-peak hours while trying not to exceed the threshold for a high number of cleaners. Since the ARIGA model was created, this is why. As cleaning time needs rise, the needed total number of janitors may surpass the threshold

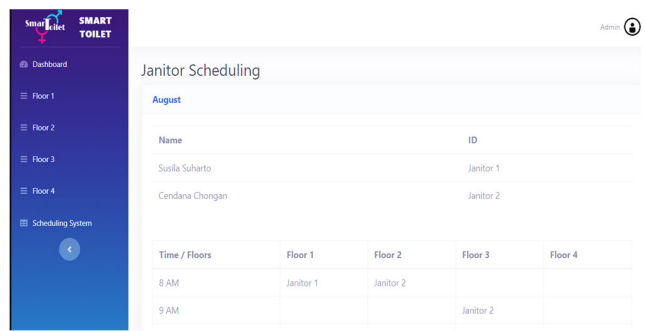


FIGURE 29. Janitor scheduling dashboard for facility view.

for increasing the number of janitors. A mechanism that lets users exchange their schedules has been proposed to fix this issue.

The Greedy scheduler performs marginally better than the baseline system in a scenario with 20 cleanliness needs. When the building’s floors grow, the Greedy scheduler’s performance worsens much more than the first schedulers. The Greedy scheduler can reduce performance by 29.4% compared to the first scheduling. Progressive stage systems can lead to unexpected results like this. The Greedy scheduler avoids the peak period, but it doesn’t account for price progression, therefore it may produce excessive load in other periods.

Figure 29 shows how to see and manage janitorial schedules on the smart toilet dashboard’s Scheduling System tab. By changing the janitor’s name and ID, the facility management staff can receive help. The facilities management team can tell the cleaning firm how many janitors are needed to clean all the floors.

In all analyses, the proposed ARIGA approach performs well. Data collection and querying are simplified using ARIGA. During pre-processing, the ARIGA preserves space, and the findings reveal that the model’s space requirements increase linearly with data size. ARIGA and LSTM, two archetypal approaches for forecasting time series data, are analyzed and compared in this paper. Using a range of cleaning demand scenarios, we found that the proposed algorithm

could greatly reduce the number of cleaning employees, saving salary.

V. CONCLUSION

This research focuses on three goals: The main goal of this study is to build and construct an IoT smart toilet management system. We covered the pros and cons of a few IoT solutions, including hardware functionality, in section II. This study's capacity to support the toilet system required examining the system's functionality and cost, even if its main goal is to recommend and create an IoT smart toilet management system. The sensor's cost and sensitivity, when installed in the bathroom's toilet, must be evaluated. Data pre-processing hardware must be reliable, consistent, and meaningful in the suggested model. This project's second purpose is to schedule a smart toilet efficiently within our resource limits. Our revised ARIGA model is presented here. ARIGA can schedule and predict maintenance for a building's IoT sensors since it uses a prediction algorithm and a scheduling algorithm. This allows ARIGA to schedule and predict building IoT sensor maintenance. Genetic algorithms are used to change the janitor's shift schedule to meet the second goal. The proposed approach reduced the number of cleaners by 24.7% on average and 15% overall in experiments on various floors and user populations. By comparing experiment results, this was determined. The model is currently evaluating the greedy algorithm and baseline scheduling method. Section III details the plan's strategy. The study's goal is to use the presented technology to do predictive maintenance on smart toilet systems. Recent advances in machine learning, particularly deep learning algorithms, are attracting researchers from many fields. Then, how efficient and precise these unique ways are compared to the conventional method is essential. We will compare the ARIGA and LSTM, two popular time series prediction methods, in the next part to see which is more accurate. The two models were employed on the same Internet of Things sensor data. The ARIGA model enhanced prediction by decreasing RMSE, MAE, and MAPE. For the ARIGA to lower AIC and BIC, it must be tuned to the right AR value.

ACKNOWLEDGMENT

The authors thank Multimedia University for giving them the chance and providing good facilities with full equipment and tools to build the project.

REFERENCES

- [1] E. Fazeldehkordi and T.-M. Grønli, "A survey of security architectures for edge computing-based IoT," *IoT*, vol. 3, no. 3, pp. 332–365, Jun. 2022, doi: [10.3390/iot3030019](https://doi.org/10.3390/iot3030019).
- [2] R. K. Ramasamy, V. Rajendran, and S. Murthy, "SMART TOILET: An IoT implementation for optimization of resources," Tech. Rep., Jul. 2018, pp. 25–27.
- [3] L. L. Thye. (Feb. 26, 2016). *The Shocking State of Our Public Toilets—Letters*. The Star Online. Kuala Lumpur, Malaysia. [Online]. Available: <https://www.thestar.com.my/opinion/letters/2016/02/22/the-shocking-state-of-our-public-toilets/>
- [4] J. Alhilman, F. Atmaji, and N. Athari, "Software application for maintenance system: A combination of maintenance methods in printing industry," in *Proc. 5th Int. Conf. Inf. Commun. Technol.*, May 2017, pp. 1–6.
- [5] W. Jin, Z. Liu, and Z. Shi, "CPS-enabled worry-free industrial applications," Tech. Rep., 2017.
- [6] E. Sezer, D. Romero, F. Guedea, M. MacChi, and C. Emmanouilidis, "An industry 4.0-enabled low cost predictive maintenance approach for SMEs: A use case applied to a CNC turning centre," in *Proc. IEEE Int. Conf. Eng. Technol. Innov.*, Jun. 2018, pp. 1–8.
- [7] K. S. Kiangala and Z. Wang, "Initiating predictive maintenance for a conveyor motor in a bottling plant using industry 4.0 concepts," *Int. J. Adv. Manuf. Technol.*, vol. 97, nos. 9–12, pp. 3251–3271, Aug. 2018, doi: [10.1007/s00170-018-2093-8](https://doi.org/10.1007/s00170-018-2093-8).
- [8] S. Ray, Y. Jin, and A. Raychowdhury, "The changing computing paradigm with Internet of Things: A tutorial introduction," *IEEE Des. Test.*, vol. 33, no. 2, pp. 76–96, Apr. 2016, doi: [10.1109/MDAT.2016.2526612](https://doi.org/10.1109/MDAT.2016.2526612).
- [9] K. Xu, Y. Qu, and K. Yang, "A tutorial on the Internet of Things: From a heterogeneous network integration perspective," *IEEE Network*, vol. 30, no. 2, pp. 102–108, Mar. 2016, doi: [10.1109/MNET.2016.7437031](https://doi.org/10.1109/MNET.2016.7437031).
- [10] S. O. Song, C. H. Jung, Y. D. Song, C.-Y. Park, H.-S. Kwon, B. S. Cha, J.-Y. Park, K.-U. Lee, K. S. Ko, and B.-W. Lee, "Background and data configuration process of a nationwide population-based study using the Korean national health insurance system," *Diabetes Metab. J.*, vol. 38, no. 5, pp. 395–403, 2014, doi: [10.4093/dmj.2014.38.5.395](https://doi.org/10.4093/dmj.2014.38.5.395).
- [11] R. Blanchet and N. Edwards, "A need to improve the assessment of environmental hazards for falls on stairs and in bathrooms: Results of a scoping review," *BMC Geriatrics*, vol. 18, no. 1, pp. 1–16, Dec. 2018, doi: [10.1186/s12877-018-0958-1](https://doi.org/10.1186/s12877-018-0958-1).
- [12] M. Raugei, D. Morrey, A. Hutchinson, and P. Winfield, "A coherent life cycle assessment of a range of lightweighting strategies for compact vehicles," *J. Cleaner Prod.*, vol. 108, pp. 1168–1176, Dec. 2015, doi: [10.1016/j.jclepro.2015.05.100](https://doi.org/10.1016/j.jclepro.2015.05.100).
- [13] X. Wen, X. Sun, Y. Sun, and X. Yue, "Airline crew scheduling: Models and algorithms," *Transp. Res. E, Logist. Transp. Rev.*, vol. 149, pp. 1–18, Apr. 2021, doi: [10.1016/j.tre.2021.102304](https://doi.org/10.1016/j.tre.2021.102304).
- [14] J. Sangai and A. Bellabdaoui, "Workload balancing in nurse scheduling problem," Tech. Rep., 2017, pp. 82–87.
- [15] I. Jacobson, I. Spence, and P.-W. Ng, "Is there a single method for the Internet of Things?" *Commun. ACM*, vol. 60, no. 11, pp. 46–53, Oct. 2017, doi: [10.1145/3106637](https://doi.org/10.1145/3106637).
- [16] A. Taivalsaari and T. Mikkonen, "A roadmap to the programmable world: Software challenges in the IoT era," *IEEE Softw.*, vol. 34, no. 1, pp. 72–80, Jan. 2017, doi: [10.1109/MS.2017.26](https://doi.org/10.1109/MS.2017.26).
- [17] H. Mora, D. Gil, R. M. Terol, J. Azorín, and J. Szlymski, "An IoT-based computational framework for healthcare monitoring in mobile environments," *Sensors*, vol. 17, no. 10, p. 2302, Oct. 2017, doi: [10.3390/S17102302](https://doi.org/10.3390/S17102302).
- [18] X. Zhang, J. Zhang, L. Li, Y. Zhang, and G. Yang, "Monitoring citrus soil moisture and nutrients using an IoT based system," *Sensors*, vol. 17, no. 3, p. 447, Feb. 2017, doi: [10.3390/s17030447](https://doi.org/10.3390/s17030447).
- [19] G. Manes, G. Collodi, L. Gelpi, R. Fusco, G. Ricci, A. Manes, and M. Passafiume, "Realtime gas emission monitoring at hazardous sites using a distributed point-source sensing infrastructure," *Sensors*, vol. 16, no. 1, p. 121, Jan. 2016, doi: [10.3390/S16010121](https://doi.org/10.3390/S16010121).
- [20] W.-F. Cheung, T.-H. Lin, and Y.-C. Lin, "A real-time construction safety monitoring system for hazardous gas integrating wireless sensor network and building information modeling technologies," *Sensors*, vol. 18, no. 2, p. 436, Feb. 2018, doi: [10.3390/s18020436](https://doi.org/10.3390/s18020436).
- [21] A. P. Plageras, K. E. Psannis, C. Stergiou, H. Wang, and B. B. Gupta, "Efficient IoT-based sensor BIG data collection-processing and analysis in smart buildings," *Future Gener. Comput. Syst.*, vol. 82, pp. 349–357, May 2018, doi: [10.1016/J.FUTURE.2017.09.082](https://doi.org/10.1016/J.FUTURE.2017.09.082).
- [22] O. Blanco-Novoa, T. M. Fernández-Caramés, P. Fraga-Lamas, and L. Castedo, "A cost-effective IoT system for monitoring indoor radon gas concentration," *Sensors*, vol. 18, no. 7, p. 2198, Jul. 2018, doi: [10.3390/S18072198](https://doi.org/10.3390/S18072198).
- [23] M. Benammar, A. Abdaoui, S. Ahmad, F. Touati, and A. Kadri, "A modular IoT platform for real-time indoor air quality monitoring," *Sensors*, vol. 18, no. 2, p. 581, Feb. 2018, doi: [10.3390/s18020581](https://doi.org/10.3390/s18020581).
- [24] S. K. Sood and I. Mahajan, "Wearable IoT sensor based healthcare system for identifying and controlling chikungunya virus," *Comput. Ind.*, vol. 91, pp. 33–44, Oct. 2017, doi: [10.1016/J.COMPIND.2017.05.006](https://doi.org/10.1016/J.COMPIND.2017.05.006).
- [25] J. Mao, Q. Lin, and J. Bian, "Application of learning algorithms in smart home IoT system security," *Math. Found. Comput.*, vol. 1, no. 1, pp. 63–76, 2018, doi: [10.3934/mfc.2018004](https://doi.org/10.3934/mfc.2018004).

- [26] K. Osathanunkul, K. Hantrakul, P. Pramokchon, P. Khoenkaw, and N. Tantitharanukul, "Configurable automatic smart urinal flusher based on MQTT protocol," in *Proc. Int. Conf. Digit. Arts, Media Technol. (ICDAMT)*, 2017, pp. 58–61, doi: [10.1109/ICDAMT.2017.7904934](https://doi.org/10.1109/ICDAMT.2017.7904934).
- [27] D. D. Koo, J. J. Lee, A. Sebastiani, and J. Kim, "An Internet-of-Things (IoT) system development and implementation for bathroom safety enhancement," *Proc. Eng.*, vol. 145, pp. 396–403, Jan. 2016, doi: [10.1016/j.proeng.2016.04.004](https://doi.org/10.1016/j.proeng.2016.04.004).
- [28] P. P. Kumar, R. Akshay, N. Achha, K. Sagar, V. Thirupathi, and M. Srinivas, "Futuristic IoT-enabled toilet maintenance system to avoid disease transmission at public toilets in smart cities," in *Proc. Int. Conf. Sustain. Comput. Data Commun. Syst. (ICSCDS)*, Apr. 2022, pp. 1032–1036, doi: [10.1109/ICSCDS53736.2022.9760987](https://doi.org/10.1109/ICSCDS53736.2022.9760987).
- [29] S. Park, D. D. Won, B. J. Lee, D. Escobedo, A. Esteva, A. Aalipour, T. J. Ge, J. H. Kim, S. Suh, E. H. Choi, and A. X. Lozano, "A mountable toilet system for personalized health monitoring via the analysis of excreta," *Nat. Biomed. Eng.*, vol. 4, no. 6, pp. 624–635, Jun. 2020, doi: [10.1038/S41551-020-0534-9](https://doi.org/10.1038/S41551-020-0534-9).
- [30] V. Mithya, N. D. Prabha, S. S. Samleim, and M. Madhumitha, "Smart toilets using turbidity sensor," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 5, pp. 413–417, 2019.
- [31] R. Sujeetha, D. Abhinav, R. Rithik, and S. Abishek, "Toilet management system using IoT," *Int. J. Sci. Technol. Res.*, vol. 8, no. 12, pp. 3193–3196, 2019.
- [32] M. Nayana, B. Chide, M. Nilesh, and P. Bobade, "Review: IoT based smart washroom," *Int. Res. J. Eng. Technol.*, vol. 7, no. 1, pp. 2090–2092, 2020. [Online]. Available: <https://www.irjet.net>
- [33] C. Greed, "The role of the public toilet: Pathogen transmitter or health facilitator?" *Building Services Eng. Res. Technol.*, vol. 27, no. 2, pp. 127–139, May 2006, doi: [10.1191/0143624406bt1510a](https://doi.org/10.1191/0143624406bt1510a).
- [34] J. Z. Sikorska, M. Hodkiewicz, and L. Ma, "Prognostic modelling options for remaining useful life estimation by industry," *Mech. Syst. Signal Process.*, vol. 25, no. 5, pp. 1803–1836, Jul. 2011, doi: [10.1016/j.ymsp.2010.11.018](https://doi.org/10.1016/j.ymsp.2010.11.018).
- [35] S. Wang, A. Gittens, and M. W. Mahoney, "Sketched ridge regression: Optimization perspective, statistical perspective, and model averaging," *J. Mach. Learn. Res.*, vol. 18, pp. 1–50, Jul. 2018.
- [36] N. J. Jarvis, "Simple physics-based models of compensatory plant water uptake: Concepts and eco-hydrological consequences," *Hydrol. Earth Syst. Sci.*, vol. 15, no. 11, pp. 3431–3446, Nov. 2011, doi: [10.5194/hess-15-3431-2011](https://doi.org/10.5194/hess-15-3431-2011).
- [37] Y. Yang, "Prediction and analysis of aero-material consumption based on multivariate linear regression model," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Big Data Anal. (ICCCBDA)*, Apr. 2018, pp. 628–632, doi: [10.1109/ICCCBDA.2018.8386591](https://doi.org/10.1109/ICCCBDA.2018.8386591).
- [38] R. Harimurti, Y. Yamasari, and B. I. G. P. Asto, "Predicting student's psychomotor domain on the vocational senior high school using linear regression," in *Proc. Int. Conf. Inf. Commun. Technol. (ICOI/ACT)*, Mar. 2018, pp. 448–453, doi: [10.1109/ICOI/ACT.2018.8350768](https://doi.org/10.1109/ICOI/ACT.2018.8350768).
- [39] E. C. Jackson, J. A. Hughes, and M. Daley, "On the generalizability of linear and non-linear region of interest-based multivariate regression models for fMRI data," in *Proc. IEEE Conf. Comput. Intell. Bioinf. Comput. Biol. (CIBCB)*, May 2018, pp. 1–8, doi: [10.1109/CIBCB.2018.8404973](https://doi.org/10.1109/CIBCB.2018.8404973).
- [40] R. C. Parpala and R. Iacob, "Application of IoT concept on predictive maintenance of industrial equipment," in *Proc. MATEC Web Conf.*, vol. 121, 2017, p. 2008, doi: [10.1051/mateconf/201712102008](https://doi.org/10.1051/mateconf/201712102008).
- [41] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi, "Machine learning for predictive maintenance: A multiple classifier approach," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 812–820, Jun. 2015, doi: [10.1109/TII.2014.2349359](https://doi.org/10.1109/TII.2014.2349359).
- [42] P. Ramos, J. M. Oliveira, and P. Silva, "Predictive maintenance of production equipment based on neural network autoregression and ARIMA," in *Proc. 21st Int. EurOMA Conf. Oper. Manag. Innov. Econ.*, 2014, pp. 1–10.
- [43] M. Syafrudin, G. Alfian, N. Fitriyani, and J. Rhee, "Performance analysis of IoT-based sensor, big data processing, and machine learning model for real-time monitoring system in automotive manufacturing," *Sensors*, vol. 18, no. 9, p. 2946, Sep. 2018, doi: [10.3390/s18092946](https://doi.org/10.3390/s18092946).
- [44] J. Luo, Z. Zhang, Y. Fu, and F. Rao, "Time series prediction of COVID-19 transmission in America using LSTM and XGBoost algorithms," *Results Phys.*, vol. 27, Aug. 2021, Art. no. 104462, doi: [10.1016/J.RINP.2021.104462](https://doi.org/10.1016/J.RINP.2021.104462).
- [45] J. Mejia, L. Avelar-Sosa, B. Mederos, E. S. Ramírez, and J. D. D. Roman, "Prediction of time series using an analysis filter bank of LSTM units," *Comput. Ind. Eng.*, vol. 157, Jul. 2021, Art. no. 107371, doi: [10.1016/J.CIE.2021.107371](https://doi.org/10.1016/J.CIE.2021.107371).
- [46] J.-R. Riba, Á. Gómez-Pau, J. Martínez, and M. Moreno-Eguilaz, "On-line remaining useful life estimation of power connectors focused on predictive maintenance," *Sensors*, vol. 21, no. 11, p. 3739, May 2021, doi: [10.3390/S21113739](https://doi.org/10.3390/S21113739).
- [47] F. M. Khan and R. Gupta, "ARIMA and NAR based prediction model for time series analysis of COVID-19 cases in India," *J. Saf. Sci. Resilience*, vol. 1, no. 1, pp. 12–18, Sep. 2020, doi: [10.1016/J.JNLSSR.2020.06.007](https://doi.org/10.1016/J.JNLSSR.2020.06.007).
- [48] L. Karlsson, "Predictive maintenance for RM12 with machine learning," Tech. Rep., 2020.
- [49] A. Binding, N. Dykeman, and S. Pang, "Machine learning predictive maintenance on data in the wild," in *Proc. IEEE 5th World Forum Internet Things (WF-IoT)*, Apr. 2019, pp. 507–512, doi: [10.1109/WF-IoT.2019.8767312](https://doi.org/10.1109/WF-IoT.2019.8767312).
- [50] I. E. Mouayni, G. Demesure, H. B.-E. Haouzi, P. Charpentier, and A. Siadat, "Jobs scheduling within industry 4.0 with consideration of worker's fatigue and reliability using greedy randomized adaptive search procedure," *IFAC-PapersOnLine*, vol. 52, no. 19, pp. 85–90, 2019, doi: [10.1016/J.IFACOL.2019.12.114](https://doi.org/10.1016/J.IFACOL.2019.12.114).
- [51] I. O. Essiet, Y. Sun, and Z. Wang, "Optimized energy consumption model for smart home using improved differential evolution algorithm," *Energy*, vol. 172, pp. 354–365, Apr. 2019, doi: [10.1016/j.energy.2019.01.137](https://doi.org/10.1016/j.energy.2019.01.137).
- [52] A. Liu and Z. Wang, "Grid task scheduling based on adaptive ant colony algorithm," in *Proc. Int. Conf. Manage. e-Commerce e-Government*, Oct. 2008, pp. 415–418, doi: [10.1109/ICMECG.2008.50](https://doi.org/10.1109/ICMECG.2008.50).
- [53] Z. Pooranian, M. Shojafar, J. H. Abawajy, and A. Abraham, "An efficient meta-heuristic algorithm for grid computing," *J. Combinat. Optim.*, vol. 30, no. 3, pp. 413–434, Oct. 2015, doi: [10.1007/s10878-013-9644-6](https://doi.org/10.1007/s10878-013-9644-6).
- [54] A. H. Kashan, "League championship algorithm: A new algorithm for numerical function optimization," in *Proc. Int. Conf. Soft Comput. Pattern Recognit.*, Dec. 2009, pp. 43–48, doi: [10.1109/SoCPAR.2009.21](https://doi.org/10.1109/SoCPAR.2009.21).
- [55] S. M. Abdulhamid, M. S. A. Latiff, and I. Idris, "Tasks scheduling technique using league championship algorithm for makespan minimization in IAAS cloud," *Nature*, vol. 388, pp. 539–547, Oct. 2018.
- [56] A. G. Delavar and Y. Aryan, "HSGA: A hybrid heuristic algorithm for workflow scheduling in cloud systems," *Cluster Comput.*, vol. 17, no. 1, pp. 129–137, Mar. 2014, doi: [10.1007/s10586-013-0275-6](https://doi.org/10.1007/s10586-013-0275-6).
- [57] T. Wang, Z. Liu, Y. Chen, Y. Xu, and X. Dai, "Load balancing task scheduling based on genetic algorithm in cloud computing," in *Proc. IEEE 12th Int. Conf. Dependable, Autonomic Secure Comput.*, Aug. 2014, pp. 146–152, doi: [10.1109/DASC.2014.35](https://doi.org/10.1109/DASC.2014.35).
- [58] A. Lokman and R. K. Ramasamy, "SMART TOILET: Threats and challenges identifying human presence using IoT sensors," in *Proc. 3rd Int. Conf. Big Data Internet Things*, Aug. 2019, pp. 56–60, doi: [10.1145/3361758.3361765](https://doi.org/10.1145/3361758.3361765).
- [59] I. Ullah, R. Ahmad, and D. Kim, "A prediction mechanism of energy consumption in residential buildings using hidden Markov model," *Energies*, vol. 11, no. 2, p. 358, Feb. 2018, doi: [10.3390/EN11020358](https://doi.org/10.3390/EN11020358).



AMAR LOKMAN received the degree in electronic engineering from the Islamic Science University of Malaysia (USIM). He is currently pursuing the Master of Science degree in information technology with the Faculty of Computing and Informatics (FCI), Multimedia University.



R. KANESARAJ RAMASAMY (Senior Member, IEEE) received the Ph.D. degree from Multimedia University, Cyberjaya, Malaysia, with thesis titled “Adaptive and Dynamic Web Service Composition for Cloud-Based Mobile Application.” He is currently a Senior Lecturer with the Faculty of Computing and Informatics, Multimedia University. He is also certified by the International Software Testing Qualification Board (ISTQB), which allows him to practice as a professional Software

Tester. He has nine years of experience in the software industry in both the development and implementation phases. He was involved in research project funded by JICA and SASTREPS to develop an early warning system for floods and landslides in Malaysia. He has published in several conferences and journals. His research interests include service oriented computing and the Internet of Things (IoT). He was awarded as a Professional Technologist (Ts) by the Malaysian Board of Technology (MBOT), the Microsoft Office 2016 Specialist, Microsoft Certified Professional, and the specialist in both web development and database technology. In 2018, he was awarded the Telekom Malaysia Research Grant as a Project Leader for the IoT Project to implement the prototype of an actual building. Besides a research grant, he is also the IoT Trainer at Telekom Malaysia. Other than the IoT training, he also provides training on mobile applications for all levels (school students and executives).



CHOO-YEE TING was the Deputy Dean of the Institute for Postgraduate Studies, Multimedia University, Cyberjaya, Malaysia. He is currently working as a Professor with the Faculty of Computing and Informatics, Multimedia University. He has been the Trainer of MDeC and INTAN, for courses related to big data and data science. He is also the consultant, a mentor, and an assessor for projects under MDeC funding. Also, he is working on trouble ticket resolution prediction at Telekom

Malaysia, seat capacity optimization at AirAsia, and dengue outbreak prediction at the Government of the Philippines. He was certified in Microsoft Technology Associate (Database), IBM DB2 CDA, and the Coursera Data Science Specialization (John Hopkins University). He is now appointed by the Academy of Science Malaysia to be the Data Scientist for supporting the Government in National Immunization Program for COVID-19 for vaccine supply modeling. He has been active in research projects related to predictive analytics and big data. Most of the projects were funded by MOE, MOSTI, Telekom Malaysia, MDeC, and industries. In 2002, he was awarded as a fellow of Microsoft Research by Microsoft Research Asia, Beijing, China. In 2014, he and his team members won the Two National-Level Big Data Analytics Competitions.

• • •