ITC 3/51
Information Technology and Control
Vol. 51 / No. 3 / 2022
pp. 545-562
DOI 10.5755/j01.itc.51.3.31273

Design and Implementation of a Self-Learner Smart Home System Using Machine Learning Algorithms

Received 2022/04/26 | Accepted after revision 2022/08/10

http://dx.doi.org/10.5755/j01.itc.51.3.31273

HOW TO CITE: Guven, C. T., Aci, M. (2022). Design and Implementation of a Self-Learner Smart Home System Using Machine Learning Algorithms. *Information Technology and Control*, 51(3), 545-562. http://dx.doi.org/10.5755/j01.itc.51.3.31273

Design and Implementation of a Self-Learner Smart Home System Using Machine Learning Algorithms

C.T. Guven, M. Acı

Faculty of Engineering, Mersin University, Ciftlikkoy Campus 33343, Mersin, Turkey; phone: +90 324 361 00 33; fax: +90 324 361 00 32; e-mails: ctamerguven@gmail.com, maci@mersin.edu.tr

Corresponding author: maci@mersin.edu.tr

Smart home systems are the integration of technology and services through the network for a better quality of life. Smart homes perform daily housework and activities more easily without user intervention or with remote control of the user. In this study, a machine learning-based smart home system has been developed. The aim of the study is to design a system that can continuously improve itself and learn instead of an ordinary smart home system that can be remotely controlled with the help of machine learning. The developed machine learning model predicts the routine activities of the users in the home and performs some operations for the user autonomously. The dataset used in the study consists of real data received from the sensors as a result of the daily use. Naive Bayes (NB) (i.e. Gaussian NB, Bernoulli NB, Multinomial NB, and Complement NB), ensemble (i.e. Random Forest, Gradient Tree Boosting and eXtreme Gradient Boosting), linear (i.e. Logistic Regression, Stochastic Gradient Descent, and Passive-Aggressive Classification), and other (i.e. Decision Tree, Support Vector Machine, K Nearest Neighbor, Gaussian Process Classifier (GPC), Multilayer Perceptron) machine learning-based algorithms were utilized. The performance of the proposed smart home system was evaluated using several performance metrics: The best results were obtained from the GPC algorithm (i.e. Precision: 0.97, Recall: 0.98, F1-score: 0.97, Accuracy: 0.97).

KEYWORDS: Smart home, machine learning, remote control, autonomous control, self-learning.

1. Introduction

Automation is a technique of operating or controlling a process by electronic devices, reducing human involvement to a minimum [16]. The technology that foresees the integration of almost all electrical devices into our daily lives by connecting to the Internet is defined as the Internet of Things (IoT). According to studies, the number of active IoT devices is expected to grow to 22 billion by 2025 [36]. The use of IoT technology with different features, suitable for all budgets, and high availability encourages people to set up their systems. Commonly used development cards are Raspberry Pi (RPi), Arduino, Asus Tinker Board, Particle Photon, Intel Galileo and Edison, Samsung ARTIK, and BeagleBone [53].

People's expectations about what a house should do or how services will be provided at home have changed over time with the development of technology and services, [28]. Smart home systems perform daily housework and activities more easily without user intervention or with remote control of the user [19, 29]. The baseline infrastructure for a home to be considered a smart home includes several sensors and actuators, user interfaces (such as voice control and graphic displays), building services (ventilation, heating, and lights), and appliance networks [2]. Researchers generally agree that a smart home system is comprised of three primary elements: internal network, home automation, and intelligent control [47]. Bluetooth, Internet, ZigBee, Near Field Communication (NFC), Global System for Mobile (GSM) communications, and Radio Frequency Identification (RFID) are generally used as connection types in smart home systems [1].

The architecture of a smart home is determined by the way devices communicate with one another, how and where the information from sensors and appliance usage habits is stored, how this information is processed and trends are extracted, and how the user can interact with the devices and vice versa [39]. Several types of architectures have been investigated in previous studies [42, 43, 46, 55, 63].

Smart home control, remote monitoring, and Human Activity Recognition (HAR) are the most widely studied topics in smart home systems [54].

There are two main approaches in HAR studies: knowledge-driven and data-driven models. Knowledge-driven methods [15, 49] use prior domain knowledge to model current activities. The data-driven approach [14, 25] utilizes probabilistic and statistical machine learning strategies to analyze and model sensor data.

The rule bases used by knowledge-driven models can be generalized to the extent that residents use the smart home system. Most of the data-driven models learn from pre-existing datasets that contain user behaviors. Activities are detected through cameras, wearable technology, RFID, or smartphones. Although studies made dependent on another device are not very reliable, they are not effective in terms of project's usability in real life. Besides, data-driven and knowledge-driven models do not generally focus on the whole smart home. Operations are carried out according to a single room in the house or specific activities. Both models are not self-learning systems, and therefore their adaptation capabilities are weak.

This study has developed a smart home system working with RPi, Arduino, and various sensors using machine learning, which is a field of computer science that gives computers the ability to learn without being explicitly programmed. The aim of the study is to design a system that can continuously improve itself and learn with the help of learning algorithms in machine learning methods, instead of an ordinary smart home system that can be remotely controlled. In our smart home system, firstly, the dataset that constitutes the system's memory must reach a certain saturation. When the dataset reaches a certain saturation, the system performs some activities autonomously for the user's benefit.

There are two basic protocols in the developed smart home system: the first protocol consists of the machine learning algorithm, control system, and basic security system that works when the users are at home. The second protocol consists of advanced security and remote control systems.

The developed machine learning model predicts the routine activities of the users in the home and performs some operations for the user. This system provides users with a more comfortable life and prevents possible human-induced disasters. The smart home system is made smarter by adding the data obtained from the sensors used in the house at certain time intervals to the dataset. Thus, the system can detect any change and frequently repeated activities. The users can control certain electronic items with an Android phone via Bluetooth at home. They can also see the temperature values with the Android application instantly. Flammable gas alarms, fire alarms, and flood detectors work in the basic security system. If the system detects any alarm situation, it sends a warning SMS to the users and shuts down the network where the alarm is detected for a certain time.

Advanced security and remote control systems come into play in the second protocol when there is no user at home. There are flammable gas, fire, flood, and motion detectors, camera image capture, sound alarm, and door alarm in the advanced security system. The system sends a warning SMS to residents in case of any alarm. At the same time, users can see the temperature and humidity values, alarm status, and control panel from the web interface and mobile application.

In our study, Naive Bayes (NB) (i.e., Gaussian NB (GNB), Bernoulli NB (BNB), Multinomial NB (MNB), and Complement NB (CNB)), ensemble (i.e., Random Forest (RF), Gradient Tree Boosting (GTB) and eXtreme Gradient Boosting (XGBoost)), linear (i.e., Logistic Regression (LR), Stochastic Gradient Descent (SGD), and Passive-Aggressive Classification (PAC)), and other (i.e., Decision Tree (DT), Support Vector Machine (SVM), K Nearest Neighbor (KNN), Gaussian Process Classifier (GPC), Multilayer Perceptron (MLP)) machine learning-based algorithms were used.

The contributions of this paper can be summarized as follows:

(1) The proposed smart home system unlike previous studies can perform activities autonomously without the data obtained from smartphones, RFIDs, or cameras. (2) The proposed smart home system can perform independently of the user using machine learning-based prediction algorithms. (3) Different machine learning algorithms were tested on the proposed smart home system, and the results were compared with each other. (4) The proposed smart home system, like many others, can allow users to control the house remotely and perform routine operations autonomously.

The rest of this paper is organized as follows: in Section 2, previous works on smart home systems are discussed. Section 3 presents the materials and methods used in the study. Section 4 presents the experimental study and discusses the results. Finally, conclusions are given in Section 5.

2. Related Studies

Smart home systems have achieved great interest in making people's lives easier and more comfortable [20]. This section discusses previous papers on smart and intelligent home automation systems. A comparative analysis of existing smart home systems has been made, which provides a clear idea of the pros and cons of the existing systems.

2.1. Bluetooth-based Systems

Bluetooth technology is designed to revolutionize the detecting and control of digital devices in people's homes and offices. It is now possible to create a network where devices can communicate with each other with built-in Bluetooth technology instead of systems where only individual devices are used. Bluetooth technology can be used cost-effectively for home automation. It generally works with the 802.15.1 standard over a 2.4 GHz frequency [30].

Naresh et al. [41] have presented the design and implementation of an automation system that can monitor and control home appliances via the ARM9 S3C2440A board. The system they developed aims to wireless control electronic devices such as lights, doors, and air conditioners in the house via Bluetooth. Asadullah and Ullah [4] have developed a low-cost and user-friendly remote-controlled home automation system using the Arduino development board, Bluetooth module, smartphone, ultrasonic distance sensor, and soil moisture sensor. The system they developed uses an ultrasonic sensor for water level detection and a soil moisture sensor for automatic plant irrigation. The prototype of the home automation system has been applied to the hardware, and successful results have been obtained. Wan and Liu [60] have presented a smart home network system based on the Nordic nrf52832 device. In this smart home system, a self-organizing network system consists of one control node, and a lot of monitor nodes were set up. The control node manages the whole network, and the monitor nodes collect the sensor information such as light intensity, temperature, humidity, PM2.5, etc. The results showed that the Bluetooth mesh wireless network system is flexible and has a low construction cost.

2.2. GSM-based Systems

GSM technology was developed by the European Telecommunications Standards Institute (ETSI) to define protocols for second-generation (2G) digital cellular networks used for mobile phones. GSM is used to transport human voice to cellular telephone networks utilizing this type of technology. In addition, GSM has many features such as Short Message Service (SMS), international roaming, fax, and data messaging service [31].

Teymourzadeh et al. [56] aimed to use GSM technology to control home appliances. In the proposed study, users can see whether any of the electrical appliances in the house is on or off via SMS, and they can also control them. The developed system prototype was made, and successful test results were obtained. Jivani [27] has created a GSM-based security control system for Android mobile phones using App Inventor. Electrical household appliances are connected to the relays and connected to the input/output digital ports of Arduino, and then the GSM module to Arduino. Users can control the relays via SMS sent by the Android application developed using App Inventor. The system has been tested by running on various phones, and successful results have been obtained. Ocak and Gökrem [44] have presented a prototype of a security system using the Arduino Uno R3 development board. The designed security system used ultrasonic distance sensors, temperature sensors, fire sensors, and gas sensors. The mobile phone in the system can send a warning SMS and e-mail to users registered in the database when there is a security-related emergency.

2.3. Internet-based Systems

With the development of wired and wireless network technologies, internet-connected mobile devices such as smartphones and tablets are now widely used. IoT is the system used to control the devices we use in our daily lives over the Internet without human intervention [57]. It is a new technology paradigm envisioned as a global network of machines and devices that can interact with each other [34].

Bingol et al. [7] developed an internet-based smart home system using Delta DVP28SV model programmable logic controller. The smart home system can be controlled by any device or operator connected to the Internet. There are ventilation, lighting, and security systems in the developed smart home. Unusual activities in the security system are reported to users via SMS. Chandramohan et al. [8] designed a home control and monitoring system using an internet protocol-connected microcontroller to remotely control devices using an Android-based smartphone application. After the system connects various electrical appliances to the Internet via Wi-Fi, it provides individual control of the devices via the Android application. The system includes lamps, fans, an LM35 temperature sensor, and light-dependent resistance (LDR). The proposed home and energy control system provides smart services and energy savings.

2.4. Artificial Intelligence and Machine Learning-based Systems

Machine learning techniques, the most well-known methods of artificial intelligence algorithms, have been used frequently in smart home studies. Predicting the probability of an event and HAR applications are the most studied topics in machine learning-based smart home systems.

Hussein et al. [22] designed an artificial intelligence-based smart home for people with disabilities. There were two Artificial Neural Networks (ANN) for adaptive learning. These networks were responsible for predicting the probability of an event. Thus, the system can be adapted to the needs of the user. A single ANN model was used in the study. Trials have not been done for different variations of the ANN, and the results have not been compared. Mehr et al. [38] conducted a study on recognizing and detecting human activities inside the smart home. They have used the smart home dataset of the Massachusetts Institute of Technology. The dataset has been tested with three different algorithms of ANNs. Güneş [18] developed a learner, web-based, low-energized, and modular home automation system in his doctoral thesis. An algorithm inspired by the Markov model and the ant colony algorithm was developed as an artificial intelligence algorithm in the study. Bianchi et al. [6] proposed an innovative HAR system using wearable devices with Convolutional Neural Networks (CNN). The system has been used to recognize a person's most common daily activities at home. Wearable technology can be a suitable method if the user is in the smart home, but it is not a very efficient solution. When home residents want to interfere with the house outside, the system ignores these activities. Du et al. [10] developed a system to solve the life problems of the elderly and disabled people in the home. Their actions inside the smart home were imitated using passive RFID tags. The Long Short-Term Memory (LSTM) method used in the study was compared only with the Naïve Bayes (NB) method. Irvine et al. [23] have used ensemble neural network, KNN, and SVM methods to solve the problem of HAR in smart homes with an open-source dataset. Liu et al. [35] presented an approach for timely daily activity recognition from an incomplete stream of sensor events. They have tested four classifiers: RF, NB, JRip, and SVM. The best results were obtained from the SVM classifier. Fahad and Tahir [13] performed activity recognition by applying a probabilistic NN on the pre-segmented activity data obtained from the sensors deployed at different locations in a smart home. They have used well-known datasets like Aruba and Milan, while the H2O auto-encoder detects the anomalies within each activity class. Previous studies on smart home systems are summarized in Table 1.

Although there have been many studies on smart home systems in recent years, previous studies have essential shortcomings:

Most studies have used rule-based systems, but machine learning has not been used.

There are also a few disadvantages seen in machine learning-based studies: either the parameters of the algorithms used were kept constant and not tested with different parameter combinations, or public datasets were used and were not customized for the residents using the smart home.

3. Material and Method

This section presents the details of hardware, software, and machine learning algorithms used in our study.

3.1. Hardware and Software

RPi and Arduino, which controlled all sensors and components, were used as microcontroller boards in our study. Also, sensors and electronic components like LDR light, flame detection, MQ5 flammable gas detection (LPG and propane gas), rain detection, sound, PIR, temperature and humidity (DHT11) sensors, relay modules, camera module, GSM module, fan module, various cables, breadboard, resistors, power cables, LM2596 voltage regulator, MCP3008 integrated, and 12V adapter were used.

Table 1

Previous studies on smart home systems

Reference	The Algorithm Used (Best in bold)	Accuracy (%)
Hussein et al. (2014)	**ANN**	80.00 – 95.00
Mehr et al. (2016)	QP, **LM**, BBP	92.81
Güneş (2016)	**RF**	Not given
Bianchi et al. (2019)	**CNN**	97.00
Du et al. (2019)	Naive Bayes, **LSTM**	78.30 – 85.00
Liu et al. (2019)	NB, JRip, **SVM**	77.70 Recall, 80.00 Precision
Irvine et al. (2020)	**Ensemble NN**, KNN, SVM	80.39
Fahad and Tahir (2021)	**Probabilistic NN**	0.90 for Aruba, 0.80 for Milan
Our study (2022)	LR, KNN, SVM, DT, XGBoost, MLP, RF, SGD, **GPC**, PAC, GNB, BNB, MNB, CNB	97.94

Arduino microcontroller board is an open-source development platform used to construct and program electronic circuits. Arduino software can be downloaded for free and can be easily programmed with libraries. It can interact with the outside world using various sensors with analog and digital inputs/outputs [17]. Due to its low power consumption and small size, Arduino Nano, shown in Figure 1(a) and its features presented in Table 2, was preferred in the study.

Table 2
Arduino Nano features [3]

Arduino Nano	Feature
Microcontroller	ATmega328
Architecture	AVR
Operating Voltage	5 V
Flash Memory	32 KB of which 2 KB used by bootloader
SRAM	2 KB
Clock Speed	16 MHz
Analog IN Pins	8
EEPROM	1 KB
DC Current per I/O Pins	40 mA (I/O Pins)
Input Voltage	7-12 V
Digital I/O Pins	22 (6 of which are PWM)
PWM Output	6 pieces
Power Consumption	19 mA
PCB Size	18 x 45 mm
Weight	7 g

RPi is a credit card-sized, low-cost, small, and portable computer developed by the Raspberry PI Foundation in England in 2009 [59]. When a keyboard, mouse, and display are connected to the RPi, it has no difference from a computer. RPi 3 Model B+, shown in Figure 1(b) and its features given in Table 3, was used in our study.

The main microcontroller used in this study is RPi. Arduino Nano was used as an auxiliary microcontroller board by connecting it to RPi with a USB cable. The purpose of using two microcontrollers is to benefit from the different good features of both development boards. The main programming language supported in RPi is Python.

Figure 1
Arduino Nano (a) and RPi 3 Model B+ (b)

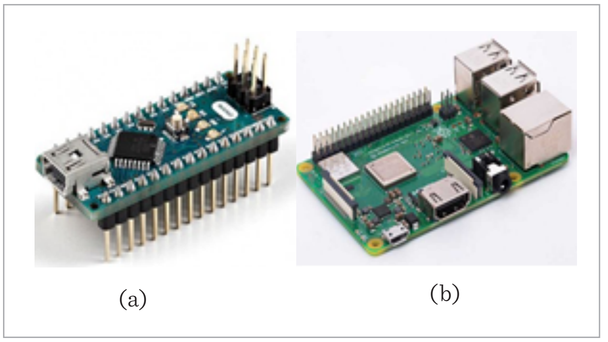


Table 3
RPi 3 Model B+ features [48]

RPi 3 Model B+	Features
Processor	Broadcom BCM-2837B0, Cortex-A53/AMRv8 Quad-Core 64-bit SoC @ 1.4GHz
Memory	1 GB LPDDR2 SDRAM
Wireless LAN	2.4GHz and 5GHz IEEE 802.11.b/g/n/ac Wireless
Ethernet	300Mbps Gigabit, PoE HAT Compatible
Bluetooth	Bluetooth 4.2, Bluetooth Low Energy (BLE)
Access	Extended 40-pin GPIO header
Storage	Micro SD Card
Video	HDMI, DSI Display Port + CSI Camera Port
Sound	4 pole 3.5 mm Stereo Output and Composite Video Port
USB	4 x USB 2.0 + MikroUSB 5 V/2,5 A DC via Micro USB Connector
Multimedia	H.264, MPEG-4 decode (1080p@30), H.264 encode (1080p@30), OpenGL ES 1.1, 2.0 graphics
Environment	Operation temperature, 0 - 50 °C
Production Lifetime	The RPi 3 Model B+ will remain in production until at least January 2023.

On the other hand, using some sensors and components in RPi is quite complex compared to Arduino. Two microcontrollers are included in the study to eliminate these problems. The smart home circuit

Figure 2
The circuit diagram of the smart home

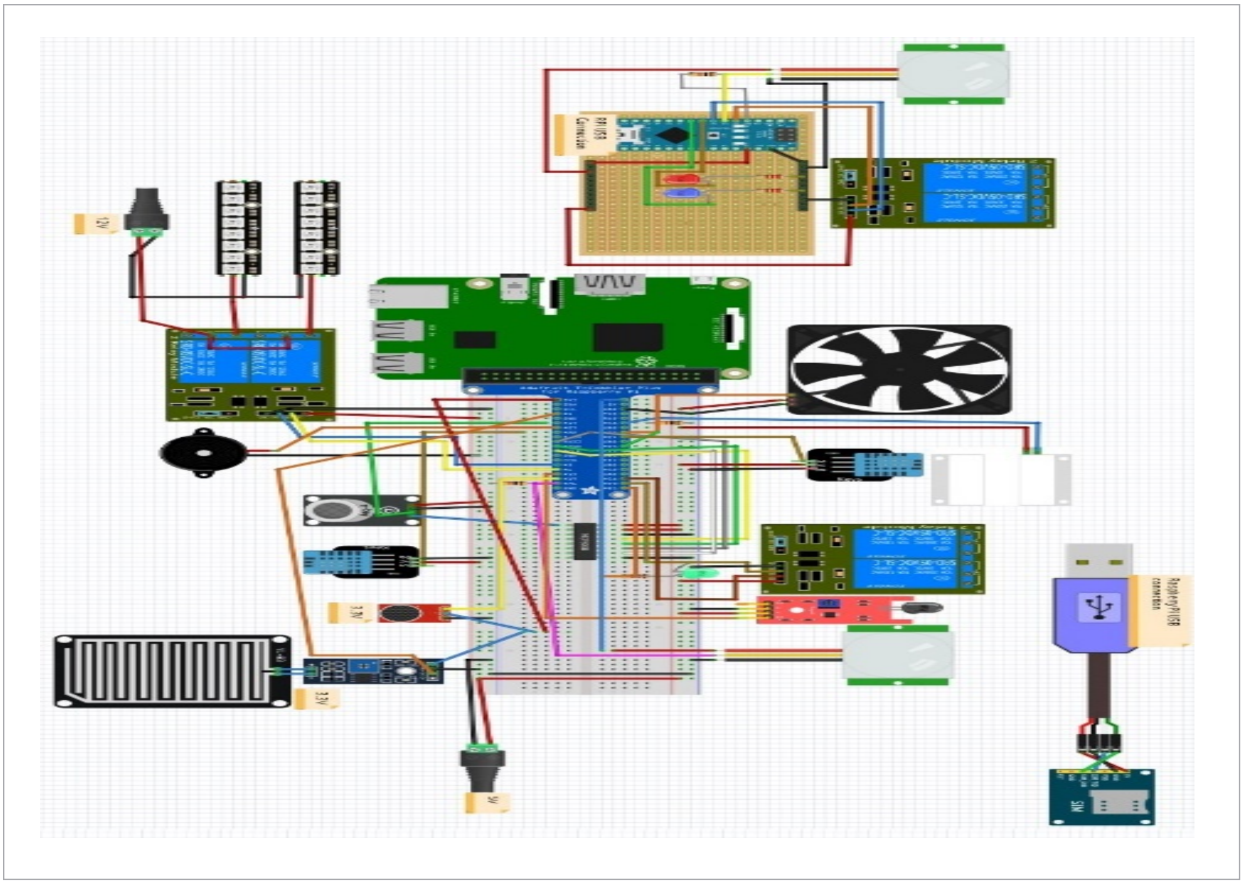


diagram with all electronic components is shown in Figure 2.

Python programming language and Arduino IDE (Integrated Development Environment) were used on Raspbian OS to develop the project. Raspbian OS has been optimized for low-performance ARM processors used in RPi. Arduino IDE is a software development platform that allows the program to be written, compiled, and loaded to the Arduino board. Scikit-Learn library was used for machine learning classification methods. Scikit-Learn is a free machine learning library for the Python programming language [50].

3.2. The Dataset

The dataset is created according to the sensors and status values instantly received from the smart home motion sensor (PIR), home temperature and humidity sensor (DHT11), outdoor temperature and humidity sensor (DHT11), gas sensor (MQ5), flame sensor, sound sensor, relays, lamps, fan, and servo motor components. It was created by adding date, day, and hour information to the values obtained from the sensors in the smart home. The dataset consists of 18 attributes and 17,531 rows of data. The dataset attributes are given in Table 4, and histograms are presented in Figure 3.

The developed smart home system inserts the values received from all sensors every half-hour, together with the date, day, and time information into the dataset. After the dataset has been created, pre-processing techniques have been carried out to obtain more successful classification results. In the pre-processing phase, outlier data produced by sensor fluctuations were removed first. Except for the date, time,

Table 4
Dataset attributes

Attribute	Description
Date (A)	Instant date value (D / W / Y)
Day (B)	Weekdays
Time (C)	Timestamp of the row added to the dataset
Resident Status (D)	Residents are at home or not (0-1)
Home Temperature (E)	Instant temperature value in the house (°C)
Home Humidity (F)	Instant humidity value in the house (%)
External Temperature (G)	Instant temperature outside the house (°C)
External Humidity (H)	Instant humidity outside the house (%)
Gas (I)	The flammable gas level (LPG / Propane Gas)
Fire (J)	Fire status (0-1)
Sound (K)	Sound level (0-1)
Living Room Light(L)	Hall lamp status (0-1)
Bedroom Light (M)	Bedroom lamp status (0-1)
Relay-1 (Coffee Maker) (N)	Status of Relay-1 (0-1)
Relay-2 (Kettle) (O)	Status of Relay-2 (0-1)
Air Condition (P)	Air conditioning status (0-1)
Curtain (R)	Curtain status (0-1)
Warning (S)	Instant alarm status (0-1)

temperature, humidity, and gas attributes, other attributes have values of 0 or 1. Therefore, the normalization process was performed only for temperature, humidity, and gas attributes. A 10-fold cross-validation process was performed before the classification algorithms were applied. Cross-validation is a technique that separates the existing dataset into training and test samples. After the dataset is divided into k equal parts, the algorithm is trained with k-1 parts. Then the algorithm is tested with the remaining part [61].

3.3. Classification Algorithms

The purpose of using classification algorithms in our study is to control the status of electronic items (i.e., relays, motorized curtains) before users according to the past data. The attributes between A and I (including A and I) were regarded as inputs, and the attributes between the J-R (J and R) were regarded as outputs in the classification algorithms. The dataset was divided into two groups, 80% training, and 20% test data, and cross-validation was performed. All algorithms used in the study have been trained and tested in the Python environment using the Scikit-Learn library.

After the classification process is completed, the obtained values are combined as a single row and added to another dataset with the current date and sensor values. The status of the electronic devices in the smart home is updated according to the new outputs, and an information message about the changes is sent to the home residents.

The following subsections are summarized the classification methods used in the study. Theoretical details were kept short since these algorithms are well-known and frequently mentioned in previous studies.

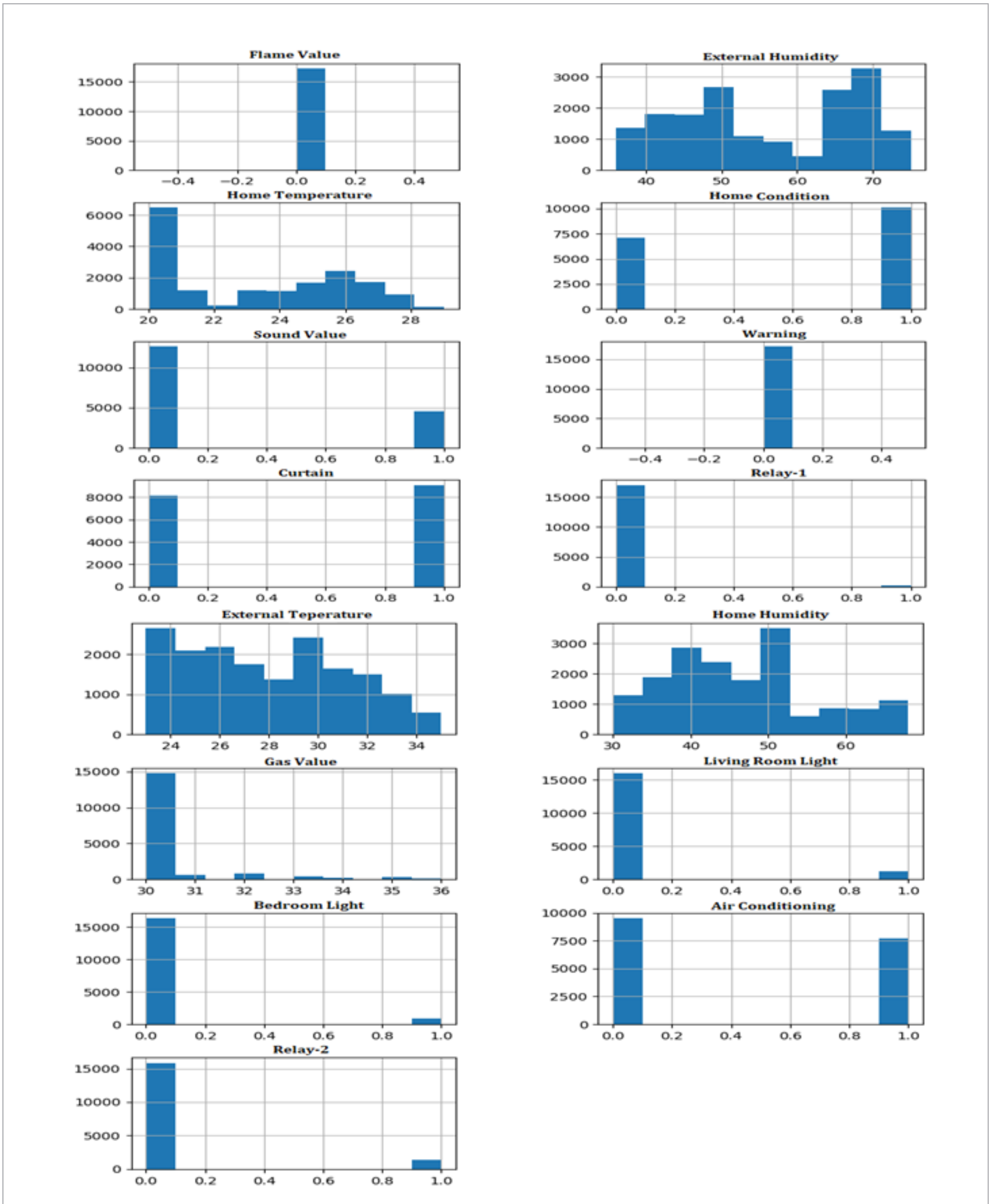
3.3.1. Naive Bayes Models

Gaussian Naive Bayes (GNB), Bernoulli Naive Bayes (BNB), Multinomial Naive Bayes (MNB), and Complement Naive Bayes (CNB) algorithms have been employed as Naive Bayes (NB) models: GNB classification is a case of NB method with an assumption of having a Gaussian distribution on attribute values given the class label [24]. BNB applies pure Bayes training and classification algorithms for data distributed according to multivariate Bernoulli distributions. There can be more than one property, but each is assumed to be a binary value variable. Therefore, samples must be represented as binary feature vectors [52]. MNB assumes that all attributes (i.e., features) are independent of each other, given the class context, and it ignores all dependencies among attributes [26]. CNB is an adaptation of the MNB algorithm that is particularly suitable for unbalanced datasets. Specifically, CNB uses the statistics of each class's complement to calculate the model's weights [51].

3.3.2. Ensemble Models

Random Forest (RF), Gradient Tree Boosting (GTB), and eXtreme Gradient Boosting (XGBoost) algorithms were used as ensemble models: The RF algorithm creates a group of classification methods based

Figure 3
Histograms of the dataset



on a combination of several decision trees. The tree-based components of these groups are formed with a certain amount of randomness. Based on this idea, the RF algorithm is defined as a general principle of randomized ensembles of decision trees [11]. The GTB method is a type of tree ensemble model in which a subsample of the train data at each iteration is randomly taken from the complete train data. This subsample will then be employed to fit the base learner and update the model for the next iteration [58]. The XGBoost algorithm is a decision tree-based ensemble machine learning algorithm that uses a gradient boosting framework. It performs better than ANN and all other algorithms in prediction problems with unstructured data [64]. XGBoost has the advantage of parallel processing; that is used all the cores of the machine it is running on [9].

3.3.3. Linear Models

Logistic Regression (LR), Stochastic Gradient Descent (SGD), and Passive-Aggressive Classification (PAC) methods were utilized as linear models: The LR algorithm, contrary to its name, is generally an algorithm that is more suitable for classification problems. LR method is used when the outcome variable is a two-level or multi-level categorical variable and takes discrete values such as 0-1, and is preferred when the dependent variable consists of categorical values such as positive-negative, low-medium-high, successful-unsuccessful [21]. Gradient Descent (GD) is a trendy optimization technique in machine learning, and it is used in most learning algorithms. In the SGD algorithm, a few random samples are selected for each iteration instead of all the data. For example, when a dataset containing one million samples is used with the classical GD algorithm, all one million samples must be used to perform an iteration. Therefore, the process turns into a very costly situation. When this dataset is used with the SGD algorithm, only one sample of data (one piece) is required to perform each iteration, and this data is randomly shuffled. Thus, in SGD, instead of the sum of the gradient of the cost function in all examples, each iteration has the gradient of the cost function of a single instance [32]. The PAC algorithm is a type of algorithm used for large-scale learning. After receiving a new data sample, it updates the model with a low loss on that new sample and makes sure that the model is close to the existing one [37].

3.3.4. Other Machine Learning- Based Models

In addition to Naive Bayes, Ensemble, and Linear models, we included Decision Tree (DT), Support Vector Machine (SVM), K Nearest Neighbor (KNN), Gaussian Process Classifier (GPC), Multilayer Perceptron (MLP) methods in our study. We classified the dataset we obtained with as many classification methods as possible and compared the results.

DTs are one of the most used algorithms in supervised learning. It has a predetermined target variable. By design, it offers a strategy that goes from top to bottom. It makes a dataset smaller pieces according to some rules. DTs can handle both categorical and numerical data [33].

The SVM is a convex optimization-based machine learning algorithm that works on the principle of structural risk minimization. SVM is mainly used to distinguish between two classes of data. Decision boundaries or hyperplanes are determined for the separation process [45].

The KNN algorithm is a supervised learning method that solves classification problems and images explained by local features. Given a local query vector and a set of properties, the KNN algorithm looks for the closest feature vectors of the query vector. It calculates the distance between the query vector and all other vectors and sorts the calculated distances. Then selects k reference vectors corresponding to the smallest distances [12].

The GPC algorithm applies Gaussian operations for more specific probability classification where test predictions take the form of class probabilities. GPC is a very effective and non-parametric classification method based on the Bayesian algorithm. GPC is a powerful algorithm to model nonlinear relationships between random variable pairs. It defines a distribution over functions that can be applied to show uncertainty about the actual functional relationship [62].

The MLP algorithm is a supervised learning algorithm that can learn a function by training a dataset. MLP is an ANN that consists of multiple sensors. It consists of an input layer to receive data input, an output layer that makes decisions or predictions about input, and a random number of hidden layers between these two layers, which is the accurate computational engine of MLP [40].

4. Results and Discussions

In this section, the technical details of the proposed smart home system and the comparison of the performance results are given.

4.1. Development of the Proposed Smart Home System

In the second protocol, advanced security and remote control systems come into play when there is no user at home. In the advanced security system, there are flammable gas alarm, fire alarm, flood detection alarm, motion detector, and camera image capture system, sound alarm, and door alarm. The system sends a warning SMS to residents in case of any alarm. At the same time, users can see the temperature and humidity values, alarm status, and control panel from the web interface and mobile application instantly. In our study, Naive Bayes (NB) (i.e. Gaussian NB (GNB), Bernoulli NB (BNB), Multinomial NB (MNB), and Complement NB (CNB)), ensemble (i.e. Random Forest (RF), Gradient Tree Boosting (GTB) and eXtreme Gradient Boosting (XGBoost)), linear (i.e. Logistic Regression (LR), Stochastic Gradient Descent (SGD), and Passive-Aggressive Classification (PAC)), and other (i.e. Decision Tree (DT), Support Vector Machine (SVM), K Nearest Neighbor (KNN), Gaussian Process Classifier (GPC), Multilayer Perceptron (MLP)) machine learning-based algorithms were used.

The smart home management system designed in our study consists of three main parts: Bluetooth control mode, web control mode, and autonomous control/security mode developed with the help of machine learning algorithms. A mobile application (Figure 4) has been developed in the Bluetooth control mode for residents to use the smart home system comfortably at home. The residents can monitor possible situations like temperature and humidity levels in the smart home system via their smartphones and perform certain activities easily. Also, residents can control the lighting system, electrical appliances, and air conditioning in the house with the mobile application. The web control (Figure 5) mode allows residents to manage and monitor the smart home system from anywhere with an Internet connection when they are away from home. While users can control lighting and electrical appliances via the web interface, they can also see the temperature and humidity levels in the house. In addition, the smart home can be monitored instantly with a security camera.

Autonomous control/security mode is developed for security and controlling issues when residents are not at home. Its user-independent features have been designed using machine learning methods.

Figure 4

A screenshot of the mobile application

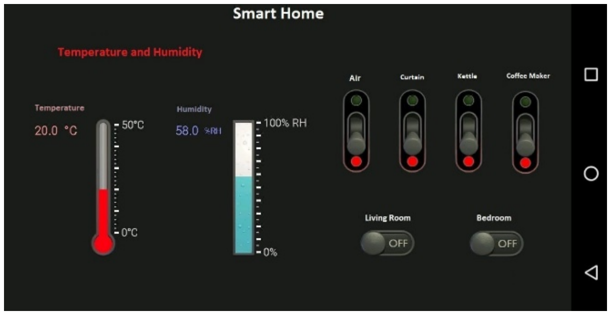
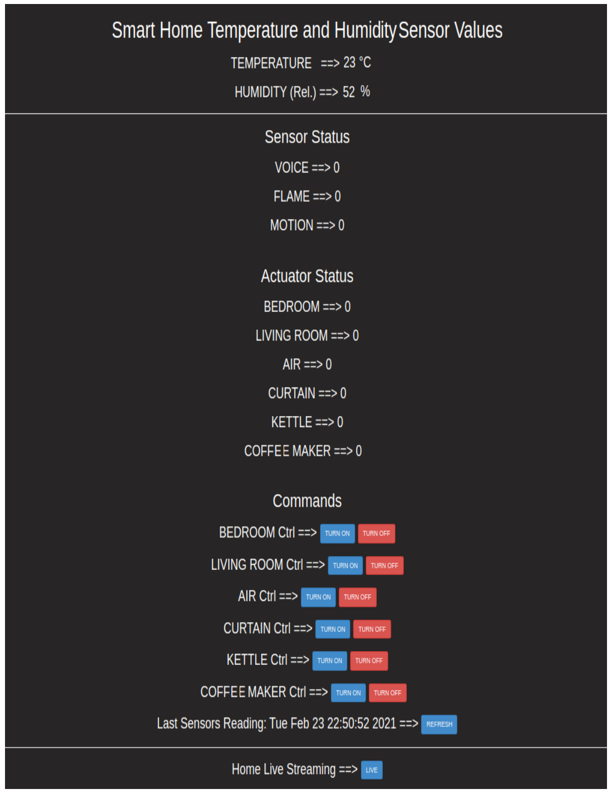


Figure 5

The web control interface



Autonomous control/security mode also enables the smart home system to control the electronic devices to detect disasters that may arise in the house due to any reason and inform the users. Door, flammable gas, sound, fire, motion, and rain alarms are available in the system. The performance of machine learning algorithms in autonomous control/security mode is analyzed in the following chapters.

The flowchart of the proposed smart home system is shown in Figure 6. The smart home management system starts with initializing hardware (i.e., GSM, Camera, Relays, Fan, Led, Buzzer, Bluetooth, Wi-Fi, and sensors). Then, the residents are, whether at home or not is, checked. If the residents are at home, a Bluetooth connection is established with residents' mobile devices, or they can access the web interface via a LAN connection. After a successful connection, the residents can view the smart home's status and controls the home appliances. If any changes are made in the smart home's status, the changes are inserted into the database as a row to be used in the machine learning algorithms. Autonomous control/security mode is activated after all residents leave the house. If any motion, voice, or activation at home is detected, the system captures images using the camera, turns on the buzzer, and sends an SMS to the authorized resident. If any fire, flood, or smoke is detected, the system turns on the buzzer, sends an SMS to the authorized resident, and cuts off the electricity.

If no danger is detected, sensor data and relays' status are used as input data to the machine learning model. The machine learning algorithm imitates regular activities at home (for example, closing curtains in sunny weather). This part of the smart home is the part where it applies the information acquired from previous user habits and environmental conditions. Suppose the status of the devices in the house needs to change according to the data received from the sensors. In that case, the necessary intervention is made by the system and inserted into the dataset for feedback. In cases where changes are not required, the dataset is updated every half-hour by reading the sensors to keep the dataset up-to-date.

4.2. Performance Metrics

The following performance metrics were used to measure the performance of the fifteen machine learning methods used in our study:

$$Error\ rate = \frac{FP+FN}{TP+FP+FN+TN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$F1\text{-}score = 2*\frac{Recall*Precision}{Recall+Precision} \tag{4}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{5}$$

TP is True Positive (correctly predicted positive class), FP is False Positive (false predicted positive class), FN is False Negative (false predicted negative class), and TN is True Negative (correctly predicted negative class). F1-score is the weighted average of recall and precision, with the best value is 1, and the worst value is 0. Error charts (Figure 7) were used to observe the suitability of different algorithms to the problem and compare their performance. The performance results of the classification algorithms are presented in Table 5.

The 10-fold cross-validation results were averaged to determine the prediction performance of autonomous control and security parameters. Appropriate parameter values of the classifiers were identified by grid-search. The grid search is a precise searching method through a default specified subset of a machine learning classifier [5]. The following inferences can be made from the results given in Table 5:

- It is evident from the figures and tables that all of the algorithms except BNB have achieved very good performance (i.e., over 0.90). Assuming that each property is a binary value in the BNB algorithm may be why it performs poorly compared to the others.
- For all the algorithms used in our evaluation, satisfactory results have been obtained. The majority of the classification algorithms have yielded very similar results. However, the best classification performance (i.e. Precision: 0.97, Recall: 0.98, F1-score: 0.97, Accuracy: 0.97) was obtained from the GPC model. The SGD classifier has the second-best performance results after GPC.

Figure 6

The flowchart of the proposed smart home system

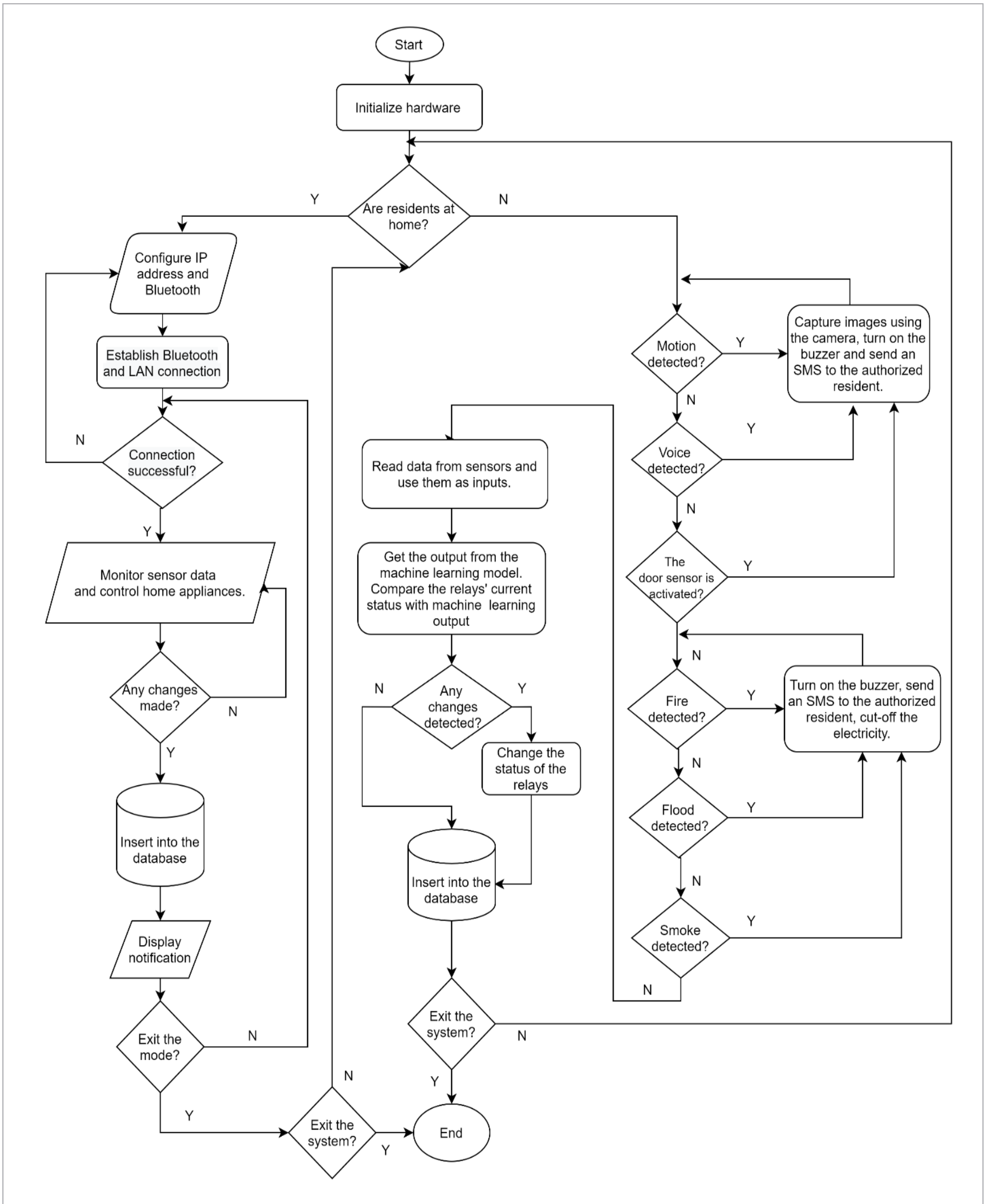


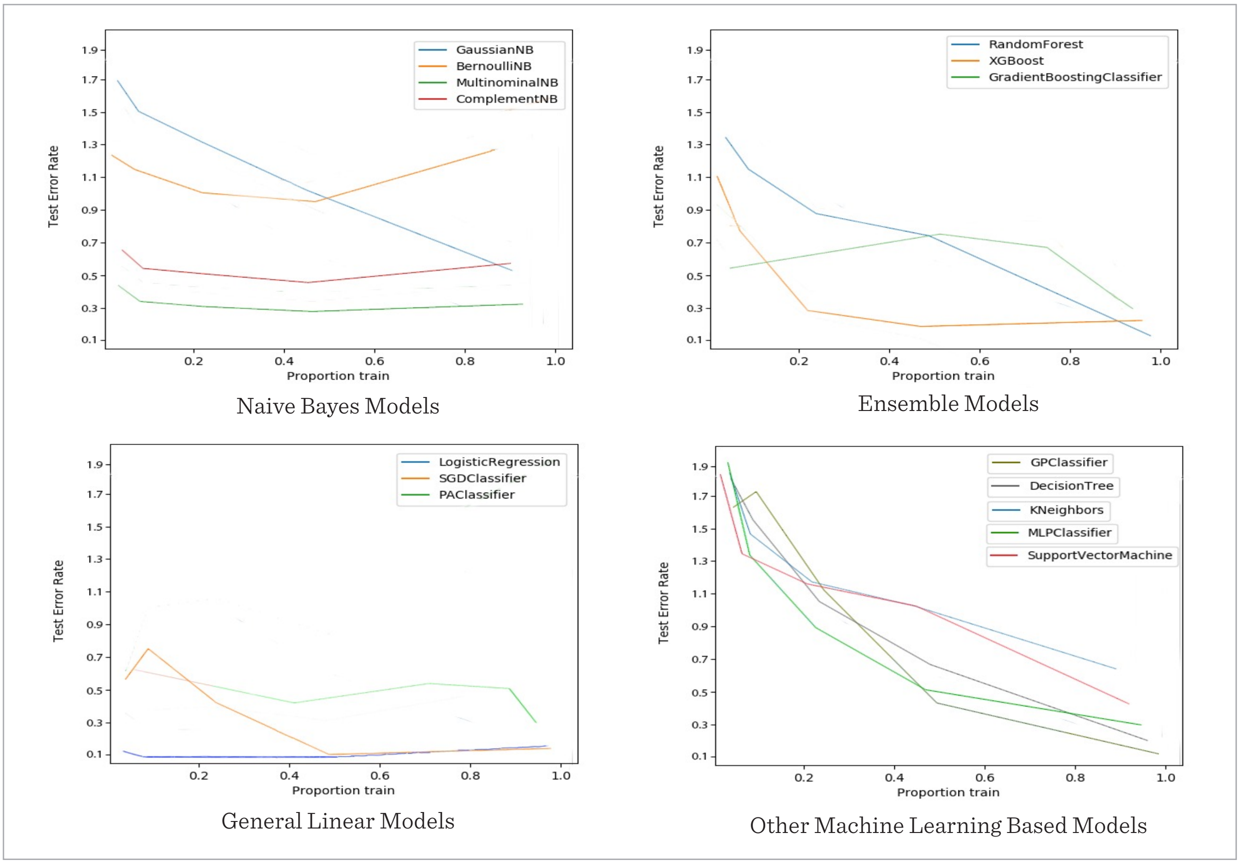
Table 5

Average precision, recall, F1-score, and accuracy results of the machine learning-based classification algorithms using 10-fold cross-validation

Model Groups	Models	Precision	Recall	F1-score	Accuracy
Naive Bayes Models	Gaussian Naive Bayes (GNB)	0.92	0.92	0.92	0.91
	Bernoulli Naive Bayes (BNB)	0.83	0.89	0.85	0.86
	Multinomial Naive Bayes (MNB)	0.95	0.95	0.95	0.94
	Complement Naive Bayes (CNB)	0.93	0.92	0.93	0.91
Ensemble Models	Random Forest (RF)	0.95	0.96	0.95	0.96
	Gradient Tree Boosting (GTB)	0.93	0.94	0.93	0.94
	XGBoost	0.94	0.95	0.94	0.95
General Linear Models	Logistic Regression (LR)	0.95	0.94	0.94	0.96
	Stochastic Gradient Descent (SGD)	0.96	0.96	0.95	0.96
	Passive Aggressive Classification (PAC)	0.93	0.93	0.93	0.94
Other Models	Decision Tree (DT)	0.95	0.94	0.94	0.96
	Support Vector Machine (SVM)	0.93	0.92	0.92	0.92
	K-Nearest Neighbour (KNN)	0.92	0.91	0.91	0.91
	Gaussian Process Classifier (GPC)	**0.97**	**0.98**	**0.97**	**0.97**
	Multilayer Perceptron (MLP)	0.94	0.93	0.93	0.94

Figure 7

Error rate charts of the classification models



Naive Bayes Models

Ensemble Models

General Linear Models

Other Machine Learning Based Models

5. Conclusions

This study presents a machine learning-based smart home system that works with various sensors and components using RPi and Arduino. The study's main purpose is to facilitate the lives of people, ensure their safety, and prevent disasters by detecting possible accidents.

The point that distinguishes this study from its counterparts is that it is a system that can learn and perform some activities autonomously for the benefit of the users, unlike the remotely controlled smart home systems developed previously. At the same time, unlike similar studies in the literature, the data obtained from smartphones, RFIDs, or cameras were not used to develop autonomous features.

In our study, the system was trained with previous user experiences and environmental factors to imitate the user's daily activities. Then, the system performs certain activities independently of the user according to the outputs obtained from the machine learning-based prediction algorithms. At the same time, another aspect of our study is that many machine learning algorithms were tested on the smart home system, and the results were compared with each other.

The developed smart home system allows users to monitor and control the house remotely as they wish and also performs routine operations autonomously. Such a system has benefits for many different types of users: It facilitates the daily lives of people who work intensively, are disabled, elderly, cannot take care of their housework as much as they should.

The successful results showed that the existing design could be improved, and better results can be obtained by using different hardware and algorithms in future works.

Acknowledgement

This study is supported by Mersin University Scientific Research Projects Department (Project Id: 2019-1-TP2-3222).

References

1. Al-Sarawi, S., Anbar, M., Alieyan, K., Alzubaidi M. Internet of Things (IoT) Communication Protocols: Review. ICIT 2017 - 8th International Conference on Information Technology, Proceedings, 2017, 685-690. https://doi.org/10.1109/ICITECH.2017.8079928
2. Ali, S. M. M., Augusto, J. C., Windridge, D. A Survey of User-Centred Approaches for Smart Home Transfer Learning and New User Home Automation Adaptation. Applied Artificial Intelligence, 2019, 33(8), 747-774. https://doi.org/10.1080/08839514.2019.1603784
3. Arduino. https://store.arduino.cc/usa/arduino-nano. Accessed on April 13, 2021.
4. Asadullah, M., Ullah, K. Smart Home Automation System Using Bluetooth Technology. International Conference on Innovations in Electrical Engineering and Computational Technologies 2017. https://doi.org/10.1109/ICIEECT.2017.7916544
5. Aya, S. A., Ormanci, Acar T., Tufekci, N. Modeling o Membrane Fouling in a Submerged Membrane Reactor Using Support Vector Regression. Desalination and Water Treatment, 2016, 57(51), 24132-24145. https://doi.org/10.1080/19443994.2016.1140080
6. Bianchi, V., Bassoli, M., Lombardo, G., Fornacciari, P., Mordonini, M., Munari, I. D. IoT Wearable Sensor and Deep Learning: An Integrated Approach for Personalized Human Activity Recognition in a Smart Home Environment. IEEE Internet of Things Journal, 2019, 6(5), 8553-8562. https://doi.org/10.1109/JIOT.2019.2920283
7. Bingol, O., Tasdelen, K., Keskin, Z., Kocaturk, Y.E. Web-based Smart Home Automation: PLC-controlled Implementation. Acta Polytechnica Hungarica, 2014, 11(3), 51-63. https://doi.org/10.12700/APH.11.03.2014.03.4
8. Chandramohan, J., Nagarajan, R., Satheeshkumar, K., Ajithkumar, N., Gopinath, P.A., Ranjithkumar, S. Intelligent Smart Home Automation and Security System Using Arduino and Wi-fi. International Journal of Engineering and Computer Science, 2017, 6(3), 2319-2342. https://doi.org/10.18535/ijecs/v6i3.53
9. Dhaliwal, S., Nahid, A. A., Abbas, R. Effective Intrusion Detection System Using XGBoost. Information, 2018, 9(7), 149. https://doi.org/10.3390/info9070149
10. Du, Y., Lim, Y., Tan, Y. A Novel Human Activity Recognition and Prediction in Smart Home Based on

Interaction. Sensors, 2019, 19(20), 4474. https://doi.org/10.3390/s19204474

11. Ellis, K., Kerr, J., Godbole, S., Lanckriet, G., Wing, D., Marshall, S. A Random Forest Classifier for the Prediction of Energy Expenditure and Type of Physical Activity from Wrist and Hip Accelerometers. Physiological Measurement, 2014, 35(11), 2191-2193. https://doi.org/10.1088/0967-3334/35/11/2191
12. Ertuğrul, Ö. F., Tağluk, M. E. A Novel Version of K Nearest Neighbor: Dependent Nearest Neighbor. Applied Soft Computing Journal, 2017, 55, 480-490. https://doi.org/10.1016/j.asoc.2017.02.020
13. Fahad, L.G., Tahir, S. F. Activity Recognition and Anomaly Detection in Smart Homes. Neurocomputing, 2021, 423, 362-72. https://doi.org/10.1016/j.neucom.2020.10.102
14. Fang, H., He, L., Si, H., Liu, P., Xie, X. Human Activity Recognition Based on Feature Selection in Smart Home Using Back-Propagation Algorithm. ISA Transactions, 2014, 53(5), 1629-1638. https://doi.org/10.1016/j.isatra.2014.06.008
15. Gayathri, K. S., Easwarakumar, K. S., Elias, S. Probabilistic Ontology Based Activity Recognition in Smart Homes Using Markov Logic Network. Knowledge-Based Systems, 2017, 121, 173-184. https://doi.org/10.1016/j.knosys.2017.01.025
16. Gunge, V. S., Yalagi, P. S. Smart Home Automation: A Literature Review. International Journal of Computer Applications, 2016, 1(1), 6-10.
17. Gunputh, S., Murdan, A. P., Oree, V. Design and Implementation of a Low-Cost Arduino-Based Smart Home System. 9th IEEE International Conference on Communication Software and Networks, 2017, 1491-1495. https://doi.org/10.1109/ICCSN.2017.8230356
18. Güneş, H., Akdaş, D. Web Based, Low Cost and Modular Home Automation System. Tehnički Vjesnik, 2016, 23(2), 533-538. https://doi.org/10.17559/TV-20141029135313
19. Güven, C. T., Acı, M. Artificial Neural Network Based Mobile Service Robot Developed for Smart Home Systems. European Journal of Science and Technology, 2019, 74-81. https://doi.org/10.31590/ejosat.636386
20. Hasan, M., Biswas, P., Bilash, M. T. I., Dipto, M. A. Z. Smart Home Systems: Overview and Comparative Analysis. 4th IEEE International Conference on Research in Computational Intelligence and Communication Networks, 2018, 264-268. https://doi.org/10.1109/ICRCICN.2018.8718722
21. Huang, D. Y., Lin, W. L., Wang, Y. Y. Approaches of Transfer Learning and Fine-Tuning on the Effects of Performance of Vehicle Classification. Journal of Computers, 2020, 31(6), 24-37. doi:10.3966/199115992020123106003
22. Hussein, A., Adda, M., Atieh, M., Fahs, W. Smart Home Design for Disabled People Based on Neural Networks. Procedia Computer Science, 2014, 37, 117-26. https://doi.org/10.1016/j.procs.2014.08.020
23. Irvine, N., Nugent, C., Zhang, S., Wang, H. Neural Network Ensembles for Sensor-Based Human Activity Recognition within Smart Environments. Sensors, 2020, 20(1), 216. https://doi.org/10.3390/s20010216
24. Jahromi, A. H., Taheri, M. A Non-parametric Mixture of Gaussian Naive Bayes Classifiers Based on Local Independent Features. 19th CSI International Symposium on Artificial Intelligence and Signal Processing, 2018, 209-212. https://doi.org/10.1109/AISP.2017.8324083
25. Jethanandani, M., Sharma, A., Perumal, T., Chang, J.-R. Multi-label Classification Based Ensemble Learning for Human Activity Recognition in Smart Home. Internet of Things, 2020, 12, 100324. https://doi.org/10.1016/j.iot.2020.100324
26. Jiang, L., Wang, S., Li, C., Zhang, L. Structure Extended Multinomial Naive Bayes. Information Sciences, 2016, 329, 346-56. https://doi.org/10.1016/j.ins.2015.09.037
27. Jivani, M. N. GSM Based Home Automation System Using App-Inventor for Android Mobile Phone. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 2014, 3(9), 12121-12128. https://doi.org/10.15662/ijareeie.2014.0309042
28. Jose, A. C., Malekian, R. Smart Home Automation Security: A Literature Review. Smart Computing Review, 2015, 5(4), 269-285. https://doi.org/10.6029/smartcr.2015.04.004
29. Kadam, M.R., Pranav Mahamuni, M., Parikh M. Y. Smart Home System. International Journal of Innovative Research in Advanced Engineering (IJIRAE), 2015, 2(1), 2349-2163.
30. Kamelia, L., Sunan, U., Djati Bandung, G., Mada, W. S., Uin, S., Gunung, S. Door-automation System Using Bluetooth-Based Android for Mobile Phone. ARPN Journal of Engineering and Applied Sciences, 2014, 9(10), 1759-1762.
31. Kartal, M., Golezani, J. J., Doken, B. A Triple Band Frequency Selective Surface Design for GSM Systems by Utilizing a Novel Synthetic Resonator. IEEE Transac-

tions on Antennas and Propagation, 2017, 65(5), 2724-2727. https://doi.org/10.1109/TAP.2017.2670230

32. Ketkar, N. Stochastic Gradient Descent. In: Deep Learning with Python, Apress, 2017, 113-132. https://doi.org/10.1007/978-1-4842-2766-4_8
33. Kotsiantis, S. B. Decision Trees: A Recent Overview. Artificial Intelligence Review, 2013, 39(4), 261-283. https://doi.org/10.1007/s10462-011-9272-4
34. Lee, I., Lee, K. The Internet of Things (IoT): Applications, Investments, and Challenges for Enterprises. Business Horizons, 2015, 58(4), 431-440. https://doi.org/10.1016/j.bushor.2015.03.008
35. Liu, Y., Wang, X., Zhai, Z., Chen, R., Zhang, B., Jiang, Y. Timely Daily Activity Recognition from Headmost Sensor Events. ISA Transactions, 2019, 94, 379-390. https://doi.org/10.1016/j.isatra.2019.04.026
36. Lueth, K. L. State of the IoT 2020: 12 Billion Iot Connections, Surpassing Non-Iot for the First Time. https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time/. Accessed on May 10, 2021.
37. Martínez-Rego, D., Fernández-Francos, D., Fontenla-Romero, O., Alonso-Betanzos, A. Stream Change Detection via Passive-aggressive Classification and Bernoulli CUSUM. Information Sciences, 2015, 305, 130-145. https://doi.org/10.1016/j.ins.2015.01.022
38. Mehr, H. D., Polat, H., Cetin, A. Resident Activity Recognition in Smart Homes by Using Artificial Neural Networks. 4th International Istanbul Smart Grid Congress and Fair, 2016, 1-10. https://doi.org/10.1109/SGCF.2016.7492428
39. Mocrii, D., Chen, Y., Musilek, P. IoT-based Smart Homes: A Review of System Architecture, Software, Communications, Privacy and Security. Internet of Things, 2018, 1-2, 81-98. https://doi.org/10.1016/j.iot.2018.08.009
40. Mubalaike, A., Adali, E. Multilayer Perceptron Neural Network Technique for Fraud Detection. 2nd International Conference on Computer Science and Engineering, UBMK 2017, 383-387. https://doi.org/10.1109/UBMK.2017.8093417
41. Naresh, D., Chakradhar, B., Krishnaveni, S. Bluetooth Based Home Automation and Security System Using ARM9. International Journal of Engineering Trends and Technology, 2013, 4(9), 4052-4058.
42. Nawaratne, R., Alahakoon, D., Silva, D., De Chhetri, P., Chilamkurti, N. Self-Evolving Intelligent Algorithms for Facilitating Data Interoperability in Iot Environments. Future Generation Computer Systems, 2018, 86, 421-432. https://doi.org/10.1016/j.future.2018.02.049
43. Ning, H., Li, Y., Shi, F., Yang, LT. Heterogeneous Edge Computing Open Platforms and Tools for Internet of Things. Future Generation Computer Systems, 2020, 106, 67-76. https://doi.org/10.1016/j.future.2019.12.036
44. Ocak, F., Gökrem, L. Arduino and PC Based Security System Prototype Design. Journal of New Results in Engineering and Natural Science, 2015, 1(4), 24.
45. Pisner, D.A., Schnyer, D.M. Support Vector Machine. In: Machine Learning: Methods and Applications to Brain Disorders, Elsevier, 2019, 101-121. https://doi.org/10.1016/B978-0-12-815739-8.00006-7
46. Rahimi, M., Songhorabadi, M., Kashani, M. H. Fog-based Smart Homes: A Systematic Review. Journal of Network and Computer Applications, 2020, 153, 102531. https://doi.org/10.1016/j.jnca.2020.102531
47. Rashidi, P., Cook, D. J. Keeping the Resident in the Loop: Adapting the Smart Home to the User. IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, 2009, 39(5), 949-959. https://doi.org/10.1109/TSMCA.2009.2025137
48. Raspberry Pi. https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/. Accessed on April 15, 2021.
49. Rawashdeh, M., Zamil, M. G., Samarah, S., Hossain, M. S., Muhammad, G. A Knowledge-driven Approach for Activity Recognition in Smart Homes Based on Activity Profiling. Future Generation Computer Systems, 2020, 107, 924-941. https://doi.org/10.1016/j.future.2017.10.031
50. Scikit-learn, Machine Learning in Python. https://scikit-learn.org/stable/. Accessed on May 12, 2021.
51. Seref, B., Bostanci, E. Performance of Naïve and Complement Naïve Bayes Algorithms Based on Accuracy, Precision and Recall Performance Evaluation Criterions. International Journal of Computing Academic Research, 2019, 8(5), 75-92. https://doi.org/10.1109/ICEEE2019.2019.00033
52. Singh, G., Kumar, B., Gaur, L., Tyagi, A. Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification. International Conference on Automation, Computational and Technology Management, 2019, 593-596. https://doi.org/10.1109/ICACTM.2019.8776800
53. Singh, K. J., Kapoor, D. S. Create Your Own Internet of Things: A Survey of IoT Platforms. IEEE Consumer

Electronics Magazine, 2017, 6(2), 57-68. https://doi.org/10.1109/MCE.2016.2640718

54. Skocir, P., Krivic, P., Tomeljak, M., Kusek, M., Jezic, G. Activity Detection in Smart Home Environment. Procedia Computer Science, 2016, 96, 672-681. https://doi.org/10.1016/j.procs.2016.08.249
55. Tang, S., Kalavally, V., Ng, K. Y., Parkkinen, J. Development of a Prototype Smart Home Intelligent Lighting Control Architecture Using Sensors Onboard a Mobile Computing System. Energy and Buildings, 2017, 138, 368-376. https://doi.org/10.1016/j.enbuild.2016.12.069
56. Teymourzadeh, R., Ahmed, S. A., Chan, K. W., Hoong, M. V. Smart GSM Based Home Automation System. IEEE Conference on Systems, Process and Control, 2013, 306-309. https://doi.org/10.1109/SPC.2013.6735152
57. Torgul, B., Şağbanşua, L., Balo, F. Internet of Things: A Survey. International Journal of Applied Mathematics, Electronics and Computers, 2016,(4), 104-108. https://doi.org/10.18100/ijamec.267197
58. Vu, Q. V., Truong, V. H., Thai, H. T. Machine Learning-based Prediction of CFST Columns Using Gradient Tree Boosting Algorithm. Composite Structures, 2021, 259, 113505. https://doi.org/10.1016/j.compstruct.2020.113505
59. Wai Zhao, C., Jegatheesan, J., Chee Loon, S. Exploring IOT Application Using Raspberry Pi. International Journal of Computer Networks and Applications, 2015, 2(1), 27-34.
60. Wan, Q., Liu, J. Smart-Home Architecture Based on Bluetooth Mesh Technology. IOP Conference Series: Materials Science and Engineering, 2018, 322(7), 072004. https://doi.org/10.1088/1757-899X/322/7/072004
61. Wong, T. T., Yeh, P. Y. Reliable Accuracy Estimates from k-Fold Cross Validation. IEEE Transactions on Knowledge and Data Engineering, 2020, 32(8), 1586-94. https://doi.org/10.1109/TKDE.2019.2912815
62. Xiao, G., Cheng, Q., Zhang, C. Detecting Travel Modes Using Rule-Based Classification System and Gaussian Process Classifier. IEEE Access, 2019, 7, 116741-116752. https://doi.org/10.1109/ACCESS.2019.2936443
63. Yassine, A., Singh, S., Hossain, M. S., Muhammad, G. IoT Big Data Analytics for Smart Homes with Fog and Cloud Computing. Future Generation Computer Systems, 2019, 91, 563-573. https://doi.org/10.1016/j.future.2018.08.040
64. Zheng, H., Yuan, J., Chen, L. Short-Term Load Forecasting Using EMD-LSTM Neural Networks with a Xgboost Algorithm for Feature Importance Evaluation. Energies, 2017, 10(8), 1168. https://doi.org/10.3390/en10081168