Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.044700

ARTICLE

Efficient and Secure IoT Based Smart Home Automation Using Multi-Model Learning and Blockchain Technology

Nazik Alturki[1], Raed Alharthi[2], Muhammad Umer[3,*], Oumaima Saidani[1], Amal Alshardan[1], Reemah M. Alhebshi[4], Shtwai Alsubai[5] and Ali Kashif Bashir[6,7,8,*]

[1]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia

[2]Department of Computer Science and Engineering, University of Hafr Al-Batin, Hafar Al-Batin, Saudi Arabia

[3]Department of Computer Science & Information Technology, The Islamia University of Bahawalpur, Bahawalpur, Pakistan

[4]Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

[5]Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj, Saudi Arabia

[6]Department of Computing and Mathematics, Manchester Metropolitan University, Manchester, UK

[7]Woxsen School of Business, Woxsen University, Hyderabad, India

[8]Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon

*Corresponding Authors: Muhammad Umer. Email: umer.sabir@iub.edu.pk; Ali Kashif Bashir. Email: dr.alikashif.b@ieee.org

Received: 06 August 2023 Accepted: 18 October 2023 Published: 11 March 2024

ABSTRACT

The concept of smart houses has grown in prominence in recent years. Major challenges linked to smart homes are identification theft, data safety, automated decision-making for IoT-based devices, and the security of the device itself. Current home automation systems try to address these issues but there is still an urgent need for a dependable and secure smart home solution that includes automatic decision-making systems and methodical features. This paper proposes a smart home system based on ensemble learning of random forest (RF) and convolutional neural networks (CNN) for programmed decision-making tasks, such as categorizing gadgets as “OFF” or “ON” based on their normal routine in homes. We have integrated emerging blockchain technology to provide secure, decentralized, and trustworthy authentication and recognition of IoT devices. Our system consists of a 5V relay circuit, various sensors, and a Raspberry Pi server and database for managing devices. We have also developed an Android app that communicates with the server interface through an HTTP web interface and an Apache server. The feasibility and efficacy of the proposed smart home automation system have been evaluated in both laboratory and real-time settings. It is essential to use inexpensive, scalable, and readily available components and technologies in smart home automation systems. Additionally, we must incorporate a comprehensive security and privacy-centric design that emphasizes risk assessments, such as cyberattacks, hardware security, and other cyber threats. The trial results support the proposed system and demonstrate its potential for use in everyday life.

KEYWORDS

Blockchain; Internet of Things (IoT); smart home automation; cybersecurity

1 Introduction

The Internet of Things (IoT) is an emerging domain linked with software programs and sensors-based devices to program certain events to be controlled with smart devices either physically or remotely using Internet-based technology [1,2]. It incorporates a variety of technical advances, including RFID, nanotechnology, sensor machinery, and smart technology [3], and sensor technology, to enable efficient information gathering and processing. Rather than being a standalone technology, IoT leverages significant advancements to bridge the gap that exists between the actual and the metaverse [4]. In today's fast-paced society, when individuals have hectic schedules, there is a growing demand for convenience and assistance in various aspects of life. While IoT encompasses a broad spectrum of research areas, this study specifically focuses on smart homes and related environment domains due to their user-friendly nature. A smart house is an automated home [5,6], where all the appliances are interconnected and controlled through a smartphone, tablet, or computer with internet connectivity. Home automation has gained significant attention in recent years as individuals prefer the capacity to manage and monitor their appliances remotely from anywhere in the globe. Consequently, home automation has become an essential requirement in today's times.

The Internet of Things (IoT) offers cost-effective and versatile solutions that enhance the quality of life for users, addressing everyday challenges. Previous studies have presented different home automation systems, incorporating multiple sensors [7,8]. However, due to limitations in research efforts, it is important to outline the reasons behind proposing a comprehensive and well-structured system for home automation.

- The previously developed home automation systems have been associated with high costs and implementation challenges.
- A prior study's Bluetooth-based home automation system necessitates undesired installation processes.
- The previously mentioned home-automation systems rely on Internet connectivity, which poses a limitation as there are areas where Internet access is not available.
- Earlier studies have been unsuccessful in developing a home automation system that ensures both security and safety.
- When it comes to intelligent decision-making mechanisms, the home automation systems suggested in earlier research fall short, especially in the security arena.

Integrating security measures into home automation system design is a difficult procedure that requires the use of a formal risk analysis technique. It has been recognized as one of the main challenges in automating smart homes, underlining the need to handle this difficult yet vital task. To guarantee that a home automation system operates efficiently, it is essential to evaluate the key parameters that may contribute to system complexity. The lack of a GUI environment is a crucial element that is frequently absent in previously created home automation systems, which limits users' knowledge of how the system works. Existing home automation solutions lack device restoration capabilities and cannot predict electricity bills accurately, making them expensive and potentially harmful to home appliances. The proposed smart home automation system addresses these shortcomings. It allows users to manage and monitor sensors and appliances, and the associated mobile app enables users to modify their home design layout and perform other functions. This study has leveraged blockchain technology to implement secure communication and authentication between IoT devices, and to allow users to change device status. This efficient approach overcomes the limitations of previous solutions. To achieve this, we follow the following aims:

- This research introduces an economical home automation system that enables remote control of electrical devices without relying on devices that use IP.
- The proposed system is an IoT-based system that utilizes an accompanying mobile application. This app enables users to simply drag and drop components to conveniently create automated homes.
- A GSM modem is included in this automation system. This connection allows the management of household equipment such as security systems, lighting, and climate control via SMS.
- The projected system contains a device restoration function that enables users to return electronic equipment or appliances to their default settings. This functionality is useful for users who want to restore their devices to a previous state or factory setting.
- The developed system includes Raspberry Pi and Arduino implementations, as a crucial tool for electronics enthusiasts and amateurs. These tools are not only popular but also reasonably priced. Raspberry Pi is well-suited for networked applications due to its simple internet access, but Arduino is well-suited for live software and hardware implementations.
- The suggested system incorporates data logging features to help customers improve energy efficiency alongside the performance of their appliances and also includes a mechanism for intelligently categorizing the state of IoT devices.
- The proposed system incorporates blockchain technology to enable secure authentication and user identity.

The rest of the paper is organized as follows: Section 2 gives a quick summary of current literature as well as important advances in IoT based solutions domain. Section 3 describes the used approach in this study for an efficient solution. Section 4 presents the data and techniques applied in experiments Section 5 describes how the suggested technique is implemented, including the software and hardware used. Section 6 contains the experimental findings, along with a detailed analysis. Finally, Section 8 brings the article to a close.

2 Related Work

This section focuses on the research gap in home automation systems and smart environments. Numerous research on smart buildings and smart homes have been undertaken. As an example, Gill et al. [7] utilized the ZigBee microcontroller to facilitate device connectivity within homes. However, this system lacks support for long-range communication and has low data transfer speeds. Similarly, Al-Ali [8] used a computer-based webserver for distant communication with household machines, despite it being a costly approach because of wire incorporation. Another research presented a phone-based controller for household appliances by [9], but GUI's absence restricts its utility for consumers. Reference [10] created a unique hand-gesture-controlled home automation system, but it had problems effectively recognizing hand movements, causing users to be inconvenienced.

In reference [11], authors created a home monitoring system for electronic gadgets by connecting electrical switches to the Internet. This technology, however, had a drawback in terms of secure transmission and communication between network devices. Reference [12] developed a home automation system that included GPRS and voice recognition in their study. However, the system lacked secure user identification and authentication. Similarly, Javale et al. [13] aimed to provide remote control of home appliances for elderly and handicapped individuals using an Android APK. But, the system's functionality is restricted, only automating light controls and power toggling for electronic devices.

In reference [14], authors sought to create a Bluetooth-enabled cellphone-based home automation system. However, the graphical user interface (GUI) of the system is only compatible with cell phones running the Symbian operating system, and the Bluetooth range is restricted to 100 meters. Similarly, while Sriskanthan et al. [15] used Bluetooth-based technology, their suggested approach encountered difficulties due to the invasive nature of the installation procedure. In recent years, blockchain technology has gained attention for its reliability. By introducing blockchain into their approach, Farshidi et al. [16] overcame the technology selection challenge. Furthermore, as illustrated by [17], blockchain technology is used in e-government services by researchers.

The threat landscape for network security has significantly worsened due to the increase in cyber-attacks and intrusion tactics. In addressing this issue, Ding et al. [18] proposed an effective anomaly detection technique that considers the complex patterns of communication between network topology and node attributes. In a study by [19], researchers examined cyber-attacks that occur recently, utilizing AI-based techniques, and identified several mitigation approaches to counteract such AI attacks. Reference [20] created an authentication mechanism tailored to RFID systems. The authors explored the integration of blockchain technology applications with the IoT to develop an energy-efficient system [21]. In another work [22], the authors incorporated a Blockchain network into the surveillance system.

In reference [23], researched Internet of Things-based cyber-attacks, explored defense techniques, and emphasized the issues connected with them. Reference [24] provided a method for evaluating the security of block cyphers that are optimized for GPU computers and are based on the branch and bound technique. Reference [25] built an effective certificate-less signing architecture using blockchain technology. Reference [26] used Arduino Tmega2560 and IoT technology to help disabled people monitor and operate their household appliances. The suggested system, however, lacked a secure authentication scheme for users. Reference [27] proposed using an Elegoo Mega 2560 controller and a web server to create an automated door-opening system at workplaces or houses. This approach, however, necessitated the storage of information about the signals sent by multiple transmitters and did not allow safe authentication. Reference [28] incorporated Arduino UNO, ESP-8266, and Wi-Fi for connection in an effort to build a system to manage household appliances remotely. However, this system had delays while turning on or off appliances. Reference [29] created a secure and energy-efficient home automation system employing a wireless sensor network. However, this device is pricey and only provides temperature control.

Current home automation systems suffer from significant limitations, including poor long-range communication support, slow data transfer, high costs, lack of secure authentication, restricted functionality, and compatibility issues. Additionally, while blockchain technology has demonstrated promise in other areas, its potential to improve home automation remains untapped. To address these shortcomings and enhance user experience, convenience, and safety, further research and innovative solutions are necessary. Bridging this research gap will lead to more efficient, secure, and user-friendly smart home systems that meet the needs of modern consumers.

2.1 Research Gap and Challenges for Smart Home Automation System

In this section, we have identified some of the research gaps that this work addresses. Addressing these challenges and filling these gaps is essential for the continued growth and development of smart home automation systems.

1. **Interoperability:** Compatibility issues between devices from different manufacturers can hinder a smart home's seamless operation. This research proposes an app that runs on all smart devices.
2. **Security Concerns:** Smart devices are vulnerable to hacking and cyberattacks, which could expose personal data or compromise home security. This research proposes a blockchain-based security framework for smart home automation that is significantly more secure.
3. **Complex Installation and Setup:** Setting up and configuring smart home devices can be challenging for non-technical users, so we designed simplified setup processes and user-friendly interfaces that allow users to set up their entire home prototype with a drag-and-drop interface.
4. **Cost:** High upfront costs can deter homeowners from adopting smart home systems. This research proposes automating daily usage appliances to smart devices, which significantly reduces the cost of smart home systems.
5. **Energy Consumption:** This research proposes a unique automatic ON/OFF predictive model that turns electrical devices on and off according to their usage pattern.

3 Proposed Approach

This study presents a smart home automation system that allows users to remotely control and monitor various appliances in their homes, such as lighting, ventilation, air conditioners, heating, and sensors. The system is designed to be both time-efficient and energy-efficient, providing users with insights into device energy use. This cost-effective and energy-efficient solution has the potential to be deployed in a variety of settings, including hotels, restaurants, domestic households, and industrial environments. The system features a user-friendly graphical user interface (GUI) and a notification system with an icon-based interface, allowing users to receive notifications and stay connected to their homes from anywhere in the world. Furthermore, the suggested system is cost-effective since it can automate routinely used electrical equipment at home, removing the requirement for specialized IP devices such as RJ-45 [30]. The suggested automation system is depicted in Fig. 1 in a visual manner.

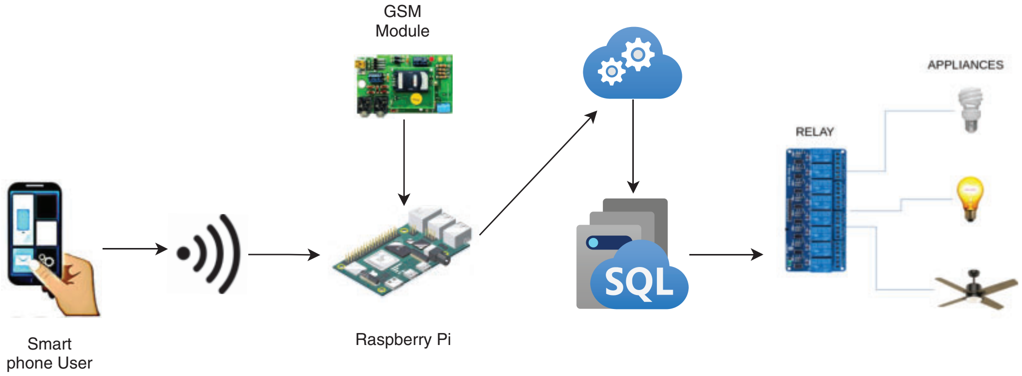


Figure 1: Schematic diagram of IoT-based smart home system

The suggested system includes an admin panel that allows users to easily create a layout of their home using a drag-and-drop interface. The process starts by adding floors to the layout, followed by adding rooms to the selected floor. Users may then add appliances to each area and place them according to the room's real construction [31]. The UI also allows you to customize the house layout by adding unique gadgets, rooms, or stories. A user's home structure is synchronized in JSON format with a server when the user has completely configured his home layout in the admin panel and the database is updated accordingly. This synchronization process utilizes VOLLEY for updating the database [32]. After syncing, the user can log in to the admin panel using their credentials and access the home structure. It takes approximately 30 s for the database to be synced and available in the admin panel. The application interface displays precise information about each level as well as the various equipment in the house. The bottom of the main screen contains three tabs. One tab reveals sensor status, while the other one provides device activity history and the identity of the person who altered the state of a given device. The third and last tab in this navigation menu allows the user to exit the application. Data durability is ensured by managing user credentials through shared preferences.

Furthermore, a supervised learning classifier is utilized to divide the status of household gadgets into two categories: "OFF" and "ON." This classifier is trained using the devices' use habits. The classifier's input data contains categorical variables such as, device_id, floor_id, and room_id alongside continuous numeric values such as device_time, room_light, room_temperature, and the device's current state. To select the best predictive model for the classification task, a comparative analysis is conducted, evaluating the performance of various conventional models. The model with the highest performance is chosen for implementation in the system. Furthermore, credentials' authentication and secure communication are guaranteed in our anticipated system between IoT devices and users that request status updates. This is accomplished by using blockchain technology, which ensures secure data transmission across smart home apps, servers, devices, and consumers.

3.1 Exploring the Potential of Blockchain for Enhanced Home Automation Security

The key focus of a home-automated system is to offer secure and reliable authentication and identification for IoT devices. To achieve this, we have implemented blockchain technology, which guarantees these objectives. Reference [33] invented blockchain technology, which has important qualities such as anonymity, decentralization, and security [34]. The IoT may construct a highly secure central server using blockchain technology, decreasing reliance and improving overall security.

Blockchain technology is leveraged in IoT-based smart homes to create a secure and organized data structure. Timestamping, encryption, hash functions, and Java-based modules are used to ensure data integrity and protect sensitive information. Additionally, the blockchain administers connectivity authorization, enhancing security, and trust within the smart home network. The verification of blockchain integrity through hash comparisons adds an extra layer of security, making it a robust solution for IoT-based smart home environments.

Timestamping: Timestamping is a fundamental feature of blockchain technology. It involves recording the exact time at which a transaction or data entry occurs. In the context of IoT-based smart homes, timestamping is crucial for maintaining a chronological record of events, such as when devices are activated, sensors collect data, or commands are executed. Timestamps help establish the order of events, ensuring data consistency and enabling forensic analysis if needed.

Data Encryption: Data encryption is employed to protect the confidentiality and integrity of data stored on the blockchain. In IoT-based smart homes, sensitive information such as user preferences, security codes, or personal data may be transmitted and recorded on the blockchain. Encryption ensures that this data is only accessible to authorized parties, preventing eavesdropping or data breaches.

Hash Functions: Hash functions are cryptographic algorithms that generate a fixed-size output (the hash) from an arbitrary input. In the context of blockchain, hash functions are used to create a unique identifier (hash) for the contents of each block. This hash serves as a digital fingerprint of the block's data. Any change in the block's data, even a single character, will result in a completely different hash. This property ensures data integrity, as any tampering with the data can be easily detected by comparing hashes.

Blockchain Module in Java: Java is a widely used programming language known for its portability and security features. In this context, a blockchain module implemented in Java likely refers to a software component responsible for managing the blockchain. This module would handle functions like block creation, hashing, validation, and interactions with other components of the smart home system.

Connectivity Authorization via Blockchain Administration: This phrase suggests that blockchain technology is used to manage and authorize connectivity between IoT devices within the smart home network. Each device's identity and access permissions are recorded on the blockchain. When a certain condition (e.g., user approval or authentication) is met, the blockchain's administration component grants authorization for devices to connect and communicate securely.

Integrity Verification: Ensuring the integrity of the entire blockchain is essential. The lines mention that the blockchain's integrity is authenticated by comparing the hash values of preceding and subsequent blocks. This is a critical security feature. Blockchain's immutability ensures that once a block is added to the chain, it cannot be altered without changing the data and recalculating the hash. By comparing hash values, the system can quickly detect any unauthorized changes or tampering attempts, maintaining the trustworthiness of the blockchain.

3.2 Workflow of Blockchain Security

Furthermore, timestamping and data encryption are used in this technology to provide a well-organized data structure. The suggested solution uses the hash function as the unique identifier for the contents of each block to construct the blockchain module in Java. Each block's SHA-256 hash is computed to get a block hash. When a certain condition is met, the block is created by providing connectivity authorization via blockchain administration. The integrity of the whole blockchain is authenticated by comparing the hash values of preceding and subsequent blocks. A user's connection request is authenticated using the Fig. 2 illustrated method. The blockchain technology working is described in Algorithm 2 in further depth. This Algorithm 2 outlines the process of creating, hashing, storing, validating, and potentially repeating steps to maintain the integrity and security of the blockchain. While Fig. 2 depicts the process of installing blockchain.

Algorithm 1: Blockchain security working algorithm

```
1: Block Creation: Begin by initiating the creation of a block using the block class.
2: Hash Generation: Generate a hash using the SHA-256 algorithm upon the successful creation of the block.
   If the block creation is unsuccessful, return to step 1 and repeat the process.
3: Block Storage: Store the newly created block after the hash has been generated.
4: Blockchain Validation: Store the newly created block after the hash has been generated.
5: Repeat or Continue: If the block is found to be invalid during validation, return to step 1 and repeat the sequence.
```

Algorithm 2: Ensembling of random forest classifier and convolutional neural network classifier (RF-CNN).

```
Input: input data $(x, y)_{i=1}^{N}$
$M_{RF}$ = Trained_RF
$M_{CNN}$ = Trained_ CNN
for $i = 1$ $to$ $M$ do
    if $M_{RF} \neq 0$ & $M_{CNN} \neq 0$ & $training\_set \neq 0$ then
        $ProbRF-ON = M_{RF}.probibility(ON - class)$
        $ProbRF-OFF = M_{RF}.probibility(OFF - class)$
        $ProbCNN-ON = M_{CNN}.probibility(ON - class)$
        $ProbCNN-OFF = M_{CNN}.probibility(OFF - class)$
        Decision function = $max\left(\frac{1}{N_{classifier}} \sum_{classifier} \left(Avg_{(ProbRF-ON, ProbCNN-OFF)}\right)\right.$
      $\left., \left(Avg_{(ProbCNN-ON, ProbCNN-OFF)}\right)\right)$
    end if
    Return final label $\widehat{p}$
end for
```

The procedure begins with the creation of a block using a Java block class. Relying on the data string, the previous hash, and the timestamp, this class computes the hash value. SHA-256 method is used to generate a hash after the block is created. The created blocks are then saved. The blockchain is then validated by comparing the computed hash value with the stored hash value. If the hash values before and following are the same, the user is allowed access. If the hash values mismatch, the whole process is restarted afresh.

4 Material and Methods

This section demonstrates the data collection, visualization, and supervised machine learning methods used to determine device status.

4.1 Data Collection and Visualization

The application produced captures data, which is subsequently saved in an Excel file. Subsequently, this data undergoes analysis to examine the connections and associations among different attributes. Light, temperature, and smoke measurements are included in the attribute data. A number of 0 in the status column indicates "OFF," whereas a value of 1 indicates "ON."

The scatter plot in Fig. 3 demonstrates the link between temperature and smoke, the x-axis displays the temperature values and the y-axis prints the smoke values. Temperature values are also displayed on the y-axis, while light readings are presented on the x-axis. A scatter plot demonstrating the relationship between smoke and light is shown in Fig. 4 with smoke values on the x-axis and light values on the y-axis. Fig. 5 is a kernel density plot, which provides a visual depiction of the data distribution.

Finally, Fig. 6 depicts the state of light and smoke, with "ON" denoting 1 and "OFF" denoting 0. The status values are shown on the x-axis, while the equivalent values are shown on the y-axis.

To ensure continuity in the event of a power outage or Raspberry Pi restarts, all devices are configured to revert to their previous states. This is accomplished by utilizing a database server to maintain and retrieve the last state of each device. Regular updates are scheduled for the devices installed in the home. For instance, if the home temperature exceeds a specific threshold, ventilation fans are automatically activated. Light sensors are fitted to manage the time of lights while keeping the day and night cycles in mind. The project is sturdy and different since it contains features such as sensor updates, data logging, Raspberry Pi compatibility, a cloud-based database, and deep learning models. The system is designed to be flexible and user-friendly, with customized device designs tailored to the specific requirements of each house.

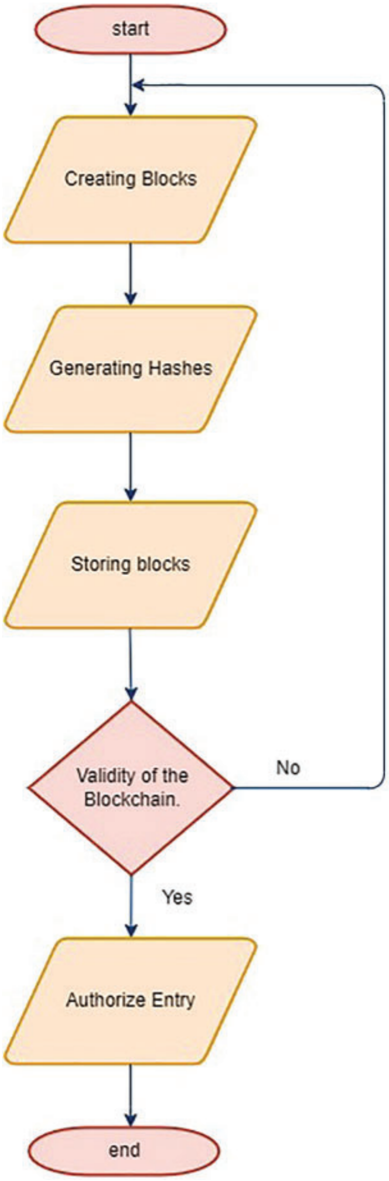


Figure 2: Flowchart of the implementation of security through blockchain technology

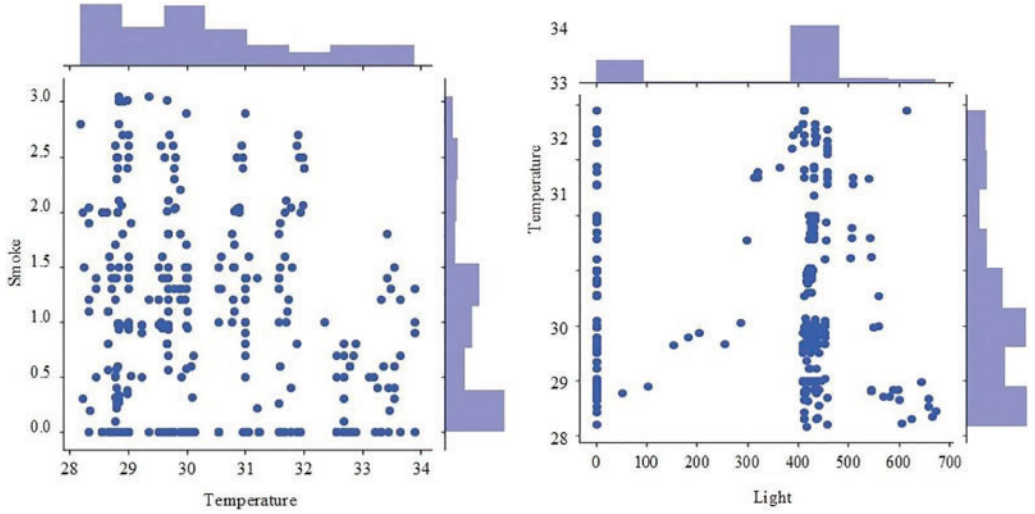


Figure 3: Scatter plot displaying the relationship between temperature, light, and smoke

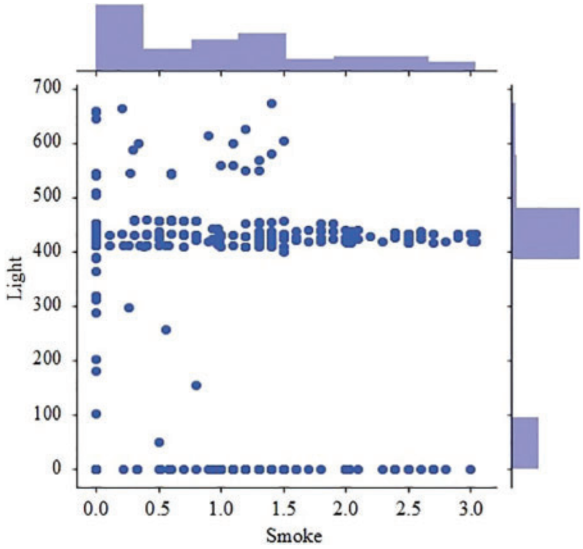


Figure 4: Scatter plot depicting the relationship between light and smoke

4.2 Supervised Learning Algorithms

Extensive trials are needed to evaluate the condition of appliances using cutting-edge models in this research. Several deep learning and ML models are used for this specific purpose, as described in the following sections.

4.2.1 Random Forest

RF creates several trees and then uses these trees randomly to reduce variation. It has reaped a lot of traction in the literature for dealing with grouping and regression challenges. RF uses a bagging approach to create predictions by aggregating the findings by majority vote. A bootstrap subset of the original data is used in this approach [35]. RF working principle is written as:

$$p = mode\{T_1(y), T_2(y), \ldots, T_m(y)\} \tag{1}$$

$$p = mode\left\{\sum_{m=1}^{m} T_m(y)\right\} \tag{2}$$

The final output, denoted by "p," is determined through majority voting among the predictions of the individual trees, represented as T_1, T_2, and T_m.

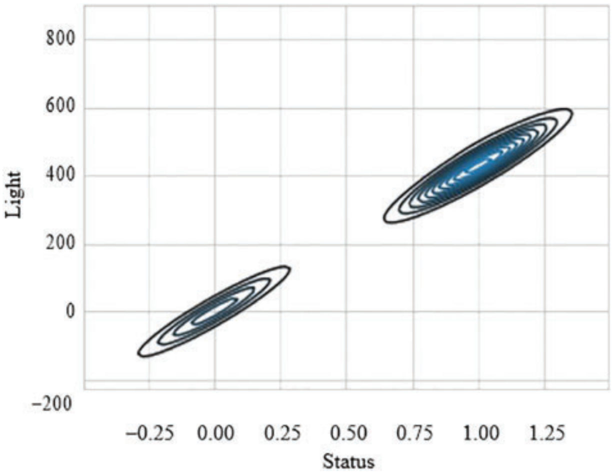


Figure 5: Scatter plot between smoke and light

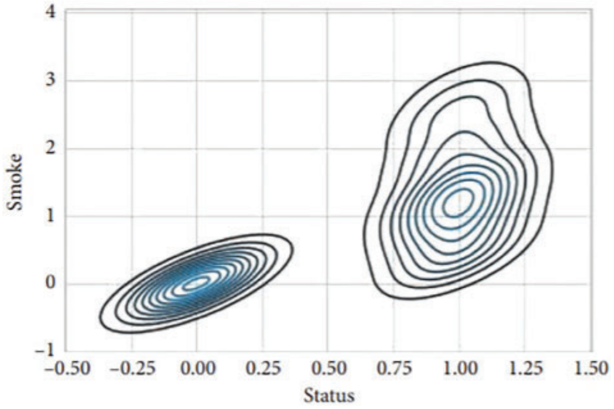


Figure 6: Scatter plot smoke status

4.2.2 Support-Vector Machine

The support-vector machine (SVM) stands as a potent machine-learning method capable of addressing classification as well as regression tasks. It achieves this by transforming the data using the kernel trick and establishing an optimal boundary line, known as hyperplanes, between different outputs. These hyperplanes effectively separate data points of one type from another. The fundamental approach of data classification involves constructing a function that assigns consistent labels to the data points while minimizing errors or maximizing the margin. A larger margin around the separating function results in fewer errors. By constructing the function, the labels become better isolated from each other. In this case, the linear kernel is employed, which provides high accuracy while minimizing time complexity. The regularisation parameter C is set at 2.0. For likelihood calculations, 500 = the option random state is used.

4.2.3 Logistic Regression

LR [36] is a popular approach for dealing with classification difficulties. It is a statistics-based model and study method that functions on the basis of the probability notion. It is particularly effective when working with binary data and aims to determine the output using one or more variables. Logistic regression employs a sigmoid function, also known as a logistic function, which establishes a relationship between the categorical data. The sigmoid function converts any input value into a number ranging from 0 and 1, producing an S-shaped curve. This enables logistic regression to estimate the likelihood of a specific class or event. The number given in logistic regression refers to the real numerical value to be translated, where "e" is the natural algorithmic base. To optimize the model, logistic regression is used with 100 iterations (max_iter). The parameter "penalty" is set to "12," which determines the model's penalization norm.

4.2.4 Stochastic Gradient Descent

SGD [37] is a method that uses the one-*vs.*-all techniques to merge many classifiers. It is suited to large datasets because it uses all data sampling in every repetition. Because of its fundamental idea, its implementation is simple. It is, however, sensitive to feature scaling, and proper hyperparameter values are required. It is a linear classifier whose cost function is regularised linear models. It provides regularised linear models with learning in order to generate an estimator. This classifier offers ease of use, and efficiency, and performs effectively with large datasets. You can utilize the sci-kit package to implement the stochastic gradient descent classifier.

4.2.5 Decision Tree

DT [38] is a basic yet effective supervised machine-learning technique that works well with both numerical and categorical input. It is extremely adaptable and has been widely used in a variety of disciplines. One of the primary benefits of decision trees is their simplicity in implementation. Decision trees utilize decision rules and feature subsets at various levels of classification. It is composed of branches with internal and leaf nodes. Feature is a representative of internal nodes while feature combinations are represented by branches leading to grouping. Every leaf node is a class representing an example. The construction quality of a DT is the hallmark of its performance on the training datasets. The decision tree is limited to a maximum growth of 300 in this example by max_depth setting = 300.

4.2.6 Gradient Boosting Machine

GBM is a strong learning classifier made up of multiple weaker classifiers [39]. GBM is based on decision trees and builds separate trees, resulting in a longer execution time. The algorithm has been improved through tweaks, specifically the PAC (probability approximately correct learning) algorithm, which enhances its performance. GBM handles missing values effectively and hence performs well on raw data. GBM requires a segregated loss mechanism to work. While regression methods commonly use logarithmic loss, classification algorithms can also utilize it. The advantage of GBM is that it can employ any differentiable loss function instead of creating a new one for each boosting iteration. Several hyperparameters in GBM must be tweaked to attain great accuracy. Setting "n" to 100, for example, shows 100 trees contributing to the forecast. Averaging all 100 decision trees' projections is needed to arrive at the final projections. The "max_depth" option can be used to limit the maximum depth of 60 levels.

4.2.7 Extra Trees classifier

The Extra Trees Classifier (ETC) algorithm [40] is comparable to the Random Forest (RF) technique but builds trees differently. Unlike RF, ETC uses original data to construct trees rather than using samples from bootstrap data. In ETC, decision-making is based on random data sampling from the k-best characteristics. The Gini index is used to find the top feature for dividing the tree. ETC and RF are regarded as equivalent since both are ensemble learning models used for categorization. The main distinction between ETC and RF lies in how the trees are constructed within the forests. In ETC, K features' random samples are drawn from feature collection randomly and distributed to each tree's test node.

4.2.8 Long Short Term Memory

LSTM is a deep learning prototype and is an extension of the Recurrent Neural Network RNN [41]. The LSTM contains a forget gate (f_k), an input gate (i_k) and an output gate (o_k). Data is channelled via these gates, retaining significant data while discarding useless information based on the set dropout value. The LSTM model also includes a memory block called C_k, where important information is stored. There are various variants of LSTM, and the one used in this study is represented by Eqs. (3)–(5).

$$i_k = \sigma(W_i s_k + V_i h_{k-1} + b_i) \tag{3}$$

$$f_k = \sigma(W_f s_k + V_f h_{k-1} + b_f) \tag{4}$$

$$o_k = \sigma(W_o s_k + V_o h_{k-1} + b_o) \tag{5}$$

$$c_k = tanh(W_c x_k + V_c h_{k-1} + b_c) \tag{6}$$

The symbols W and V in the LSTM model's Eqs. (4)–(6) denote the related weights with the matrix components. The hidden state up to the $k-1$ time step is represented by h, whereas the character s_k represents the input at that moment of the time step. The bias term is represented by the symbol b. At the $k-1$ time step, the memory block cell, represented by c, is modified. All neurons in the LSTM's output layer are linked to every neuron in the dense layer, suggesting a completely connected structure.

4.2.9 Convolutional Neural Network

CNN is a deep neural network that learns complicated characteristics by using pooling layers and convolution layers [42]. CNNs are frequently used for image segmentation and classification. The layered CNN model's resilience is enhanced by end-to-end training. As a feed-forward network model, the convolution layers of CNNs collect input data by using filters to the preceding levels' output. Activation layers, pooling layers, dropout layers, and fully linked layers are also included in CNN models. Pooling layers aid in feature selection by lowering feature dimensions, and they may be implemented as average or max-pooling. The preceding layers' output is sent to the completely linked layers, which decide the final outcome. Dropout layers are employed to avoid overfitting. The activation function is critical in identifying the significance of input information.

The activation function's output is represented by y, while the input is represented by i. The convolutional layers in CNNs extract high-level features during the training process using weights. To compute the loss function, cross-entropy is commonly employed, as shown in Eq. (7).

$$crossEntropy = -(i\ log(p) + (1 - i)\ log(1 - p)) \tag{7}$$

The class labels are represented by i, and the anticipated probability is represented by p. The sigmoid function is used to forecast output in CNN models, which are an extension of the back-propagation model. A CNN model generating output for two target classes is represented by multiple neurons in its output layer. When the device is in the "ON" state, the first neuron will have an output 1 and 0 for the other neurons. In the "OFF" state, the values are inverted, with 1 for the first neuron and 0 for the others.

4.2.10 Proposed (RF-CNN)

Voting classifiers integrate the findings of many classifiers to generate a final voting-based conclusion. Soft voting and hard voting are the two forms of voting classifiers. Soft voting is used to determine each classifier's weight, whereas hard voting is used to forecast classifier outputs. This prototype predicts the final outcome by multiplying class likelihood by classifier weight and then averaging the results for each entry. In our study, we combine a voting classifier, an RF, and a CNN to outperform previous strategies for predicting device status. Algorithm 2 demonstrates the process employed by the projected voting classifier, while the comprehensive pseudocode for the proposed model is provided in Algorithm 3.

$$\widehat{p} = argmax\left\{\sum_{i}^{n} RandomForestClassifier_i, \sum_{i}^{n} ConvolutionalNeuralNetwork_i\right\}. \tag{8}$$

where $\sum_{i}^{n} RandomForestClassifier_i$ and $\sum_{i}^{n} ConvolutionalNeuralNetwork_i$ predict the probability-based results for each test model by Random Forest Classifier and Convolutional Neural Network, respectively. RF Classifier and CNN instance's probabilities are transferred through soft voting criterion in Algorithm 2. The visual representation of the suggested ensemble model is shown in Fig. 7.

Algorithm 3: Ensemble voting classifier of random forest and CNN

```
Step 1: Load and preprocess the dataset
Load structured_data
Preprocess structured_data (e.g., normalize, handle missing values)
Step 2: Split the dataset into training and testing sets
Split structured_data into train_data and test_data
Split data into train_data and test_data
Split labels into train_labels and test_labels
Step 3: Train the Random Forest Classifier
Initialize Random_Forest_Model
Train Random_Forest_Model on train_data and train_labels
Step 4: Train the Convolutional Neural Network (CNN)
Initialize CNN_Model
Compile CNN_Model (Define optimizer, loss function, and metrics)
Train CNN_Model on train_data and train_labels
```

(Continued)

Algorithm 3 (continued)

```
Step 5: Make predictions using each model
Predictions_RF = Random_Forest_Model.predict(test_data)
Predictions_CNN = CNN_Model.predict(test_images)
Step 6: Perform ensemble voting
Initialize Ensemble_Predictions array with the same size as test_data
for each sample in test_data do
    Count the votes from Random Forest and CNN
    RF_vote = count_votes(Predictions_RF[sample])
    CNN_vote = count_votes(Predictions_CNN[sample])
    Make an ensemble decision based on majority voting
    if RF_vote > CNN_vote then
        Ensemble_Predictions[sample] = 'Class for Random Forest'
    else
        Ensemble_Predictions[sample] = 'Class for CNN'
    end if
end for
Step 7: Evaluate the ensemble model
Calculate accuracy, precision, recall, F1-score, etc., for Ensemble_Predictions and test_labels
Step 8: Display evaluation metrics and make predictions as needed
Display evaluation metrics
Make predictions for new data using the ensemble model
```

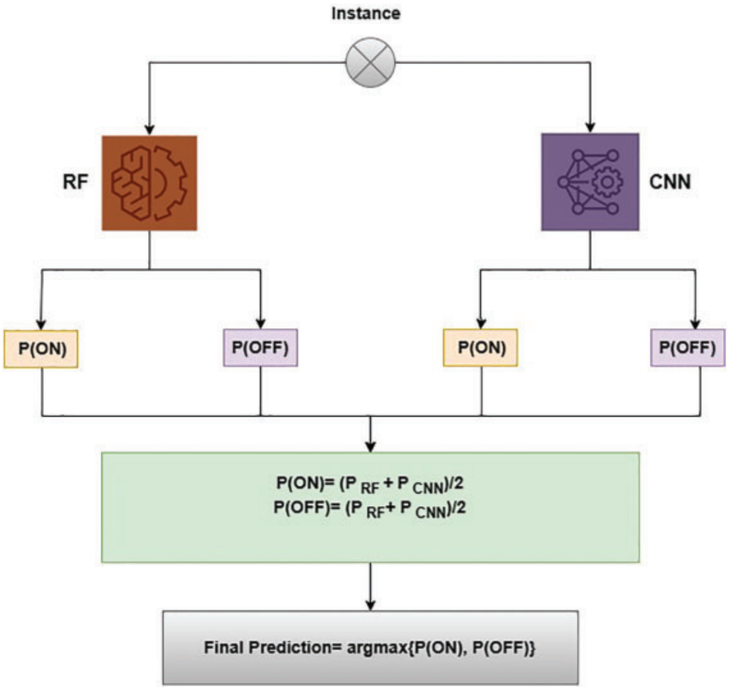


Figure 7: RF-CNN architecture

5 Implementation Detail

Fig. 1 illustrates the suggested model's entire functionality, demonstrating the integration of numerous devices to provide a thorough knowledge of solutions to smart homes. Here a project's flow is shown as an arrow path for managing equipment's status from a user's smartphone. Depending on their location, the user can interact with a server based on the Raspberry Pi through two modules [43,44]. The first module allows for interaction with IoT devices in the home without the need for an internet connection. Quick communication between the user and the device is ensured by using a local network, which is obviously the biggest advantage of this module. Internet access is required for the second module to function, as it allows global access to the device irrespective of the global position of the user, but functions only when the user is away from the home network. In this example, the connection request is sent to the Microsoft Azure Cloud for processing [45]. The credentials are then entered, which are synced with a database of Azure and routed to the appropriate server (Raspberry Pi) for processing. Individual account management is ensured by Microsoft Azure Cloud's usage of different databases for each user. When a request is initiated, services are offered to each user based on the credentials they enter. APIs are available in the cloud, even if the user is not connected to their home network. But if the user is connected to their home network, the Raspberry Pi server stores copies of the same APIs.

Data in JSON format is sent between the server and the user during communication. APIs are secured using hashing algorithms. The Raspberry Pi GPIO pins [46] are used to control and modify electronic devices' status in the system. When a request is delivered, the Raspberry Pi receives it and connects with the devices based on the user's request. The requests of the users are saved in a cloud database. This configuration allows the examination of complete user interactions on their smartphone by setting a time period. Every 30 s, the sensors in the home update their status and synchronize with the server. As a consequence, the Raspberry Pi synchronizes data from the whole database stored on the cloud and changes the mobile application values.

5.1 Hardware Components

As shown in Fig. 1, the proposed system integrates a wide range of sensors and electrical components. This section contains a detailed explanation of the hardware components used in the system. Table 1 summarises these components for simple reference.

Table 1: Details of the hardware components used in designing the smart home automation system

Components	Specification
L293D motor control shield	Supply-voltage range: 4.5–36 V; output current: 600 mA/channel
DS18B20 temperature sensor	Temperature range: −55 to 125Â°C (−67Â°F to +257Â°F)
LM393 LDR sensor	Digital switching outputs (0 and 1), external 3.3–5 V vcc
Relay circuit pack	8-relays circuit pack operates on 5 V
ARM Cortex-A7 CPU with 900 MHz quad-core, 7–12 V operational voltage	
MQ2 smoke sensor	Combustible gas, smoke

(Continued)

Table 1 (continued)

Components	Specification
Smartphone mobile	Android supported
Raspberry Pi 2B	40 GPIO pins, 1 GB RAM

The Raspberry Pi is a low-cost, single-board computer (SBC) built primarily for computer science instruction in educational institutions and developing nations. It has an ARM Cortex-A7 quad-core CPU with 900 MHz clock speed and 1 GB of RAM. The Raspberry Pi has a 100 Mbps Ethernet connection a card interface, four USB ports, and 40 GPIO pins. It also has full HDMI compatibility, enabling for camera use, and is SD card compatible. It also has a 3.5 mm audio connection for audio communication and enables composite video output. A relay is an electrical device that is commonly used in automatic control circuits. It consists of an output circuit called a controlled system or output contractor, and an input circuit called a control system or input contractor. The relay functions as a switching device, for automatically controlling the high-current circuit by using the low-current signal. It operates using electricity and serves to regulate the flow of electrical power within a circuit.

L293D [47] is a monolithic integrated driver that provides four channels for high-current and high-voltage applications. This implies that by utilizing the chip L293D, power sources and DC motors with voltages of up to 16V, including reasonably big motors, may be incorporated per channel. A maximum current of 600 mA may be delivered by the chip circuit. This chip (L293D) is composed of serial H-Bridges, which are electrical circuits capable of supplying power across a load, such as a motor, in any direction.

A single-wire temperature sensor interface DS18B20 allows users to easily measure and record temperature. It uses a bus communication system, allowing multiple devices to be connected While reading their temperature readings requires only a single Raspberry Pi GPIO pin.

The MQ2 sensor, also known as a chemiresistor, is a popular choice for detecting smoke and various gases in the MQ2 sensor series. It operates by detecting changes in resistance within its sensing element when it comes into contact with smoke particles. A main voltage divider network may be used to assess the concentration of smoke. The MQ2 sensor can detect CO (carbon monoxide), C3H8 (propane), CH4 (methane), H (hydrogen), smoke, alcohol, and LPG, with detection ranging from 200 to 10,000 parts per million (ppm). It is powered by a 5 V DC power source and consumes around 800 milliwatts (mW).

The module named GSM is equipped with a dual-mode that is commonly used in embedded applications and the Internet of Things (IoT). It operates within the frequency range of 900 to 1,800 MHz. One notable advantage of the GSM module is its low power consumption requirement. It has a multislot class functionality, such as class 8 and class 10, which allows for efficient data transfer. The TXD and RDX pins receive and transfer data, running between 3.4–4.5 V. Any voltage above this range can damage the module.

5.2 Software Components

Many mobile application development platforms are there, including Android, iOS, Windows Mobile, and Symbian. This project is completely developed using the Android platform due to its immense popularity and widespread use throughout the world. Almost every smartphone manufacturer supports Android applications, making it a flexible platform. The Android Software Development Kit (SDK) in the Java programming language is used for the development and implementation of the proposed smart home system. The Android SDK includes a complete set of tools and resources for developing Android apps.

Android APKs are created using Android Studio in this project [48]. It includes tools like libraries, handset emulators to test apps on different handsets without purchasing them, and debuggers. The Volley library, which provides easy functions for managing sensor data, is included in the project to enable the integration and use of numerous sensors. Furthermore, by utilizing the Material Design toolkit, the Android application gains an interactive user interface. In this project, LAMP is used, an abbreviation for Linux, Apache, MySQL, and PHP, as a backend development framework on the cloud within the Raspberry Pi environment.

5.3 User Interface: Navigating the Smartphone App

The smartphone app is designed with the following characteristics.

Modules: The app contains two distinct modules to enhance user experience. In the *Admin module*, users can effortlessly create a digital prototype of their home by employing a user-friendly drag-and-drop interface. Each component in the prototype corresponds to a specific Raspberry Pi pin, enabling seamless backend control of various electronic devices.

The *user module* provides users with a visual representation of their home design, as created in the admin module. It allows users to manage and control their electronic devices conveniently as shown in Fig. 8. The main page of the mobile application furnishes users with essential information about the home, including floor layouts and the inventory of installed gadgets. Navigation between various functions is facilitated through a tabbed layout.

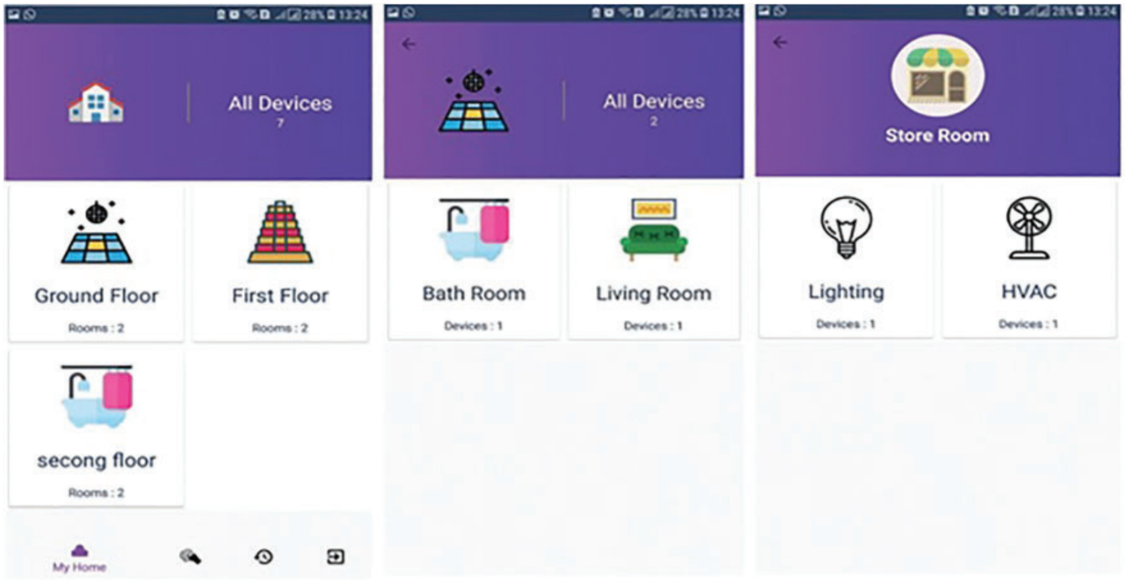


Figure 8: Operations of household gadgets

Intuitive and Interactive GUI: The graphical user interface (GUI) boasts a visually appealing and user-friendly design that simplifies user interaction as shown in Fig. 9. Dynamic icons change to reflect the real-time status of electronic devices. Users can easily toggle device states using responsive touch buttons. Moreover, an interactive intensity bar permits users to adjust fan speed and light brightness effortlessly. Active devices are clearly labelled as “active,” while inactive ones are indicated as “inactive.”

Real-Time Sensor Monitoring: The second tab on the home screen enables users to monitor sensor data, such as temperature and light levels, in real time as shown in Fig. 10. These sensor readings are updated every 30 s via the backend, ensuring users stay informed about their environment. For instance, Fig. 11 displays a light sensor value of 0.0, indicating that it is in the “OFF” mode and detecting daylight.

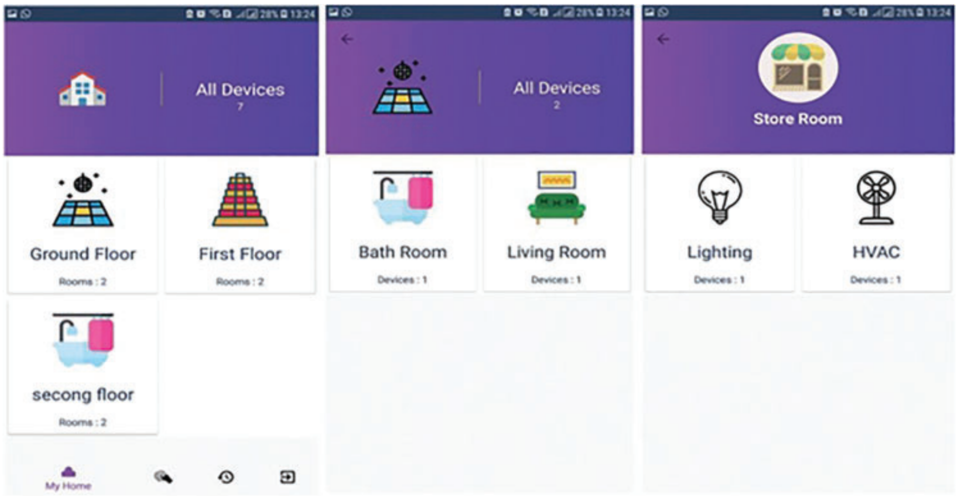


Figure 9: User Interface on the screen

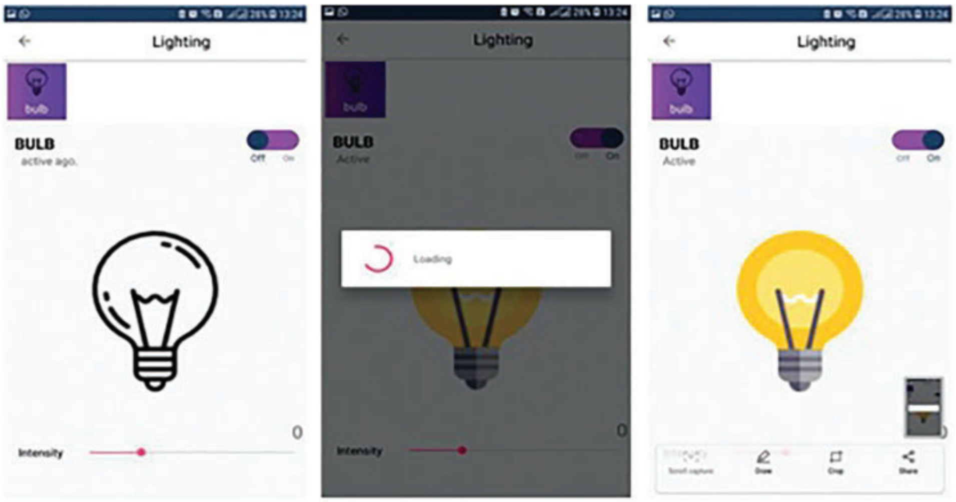
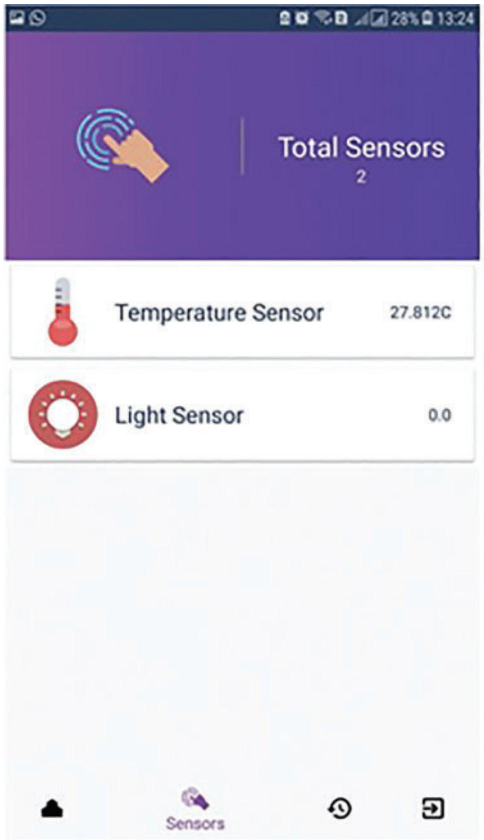


Figure 10: Household appliances’s state



Total Sensors
2

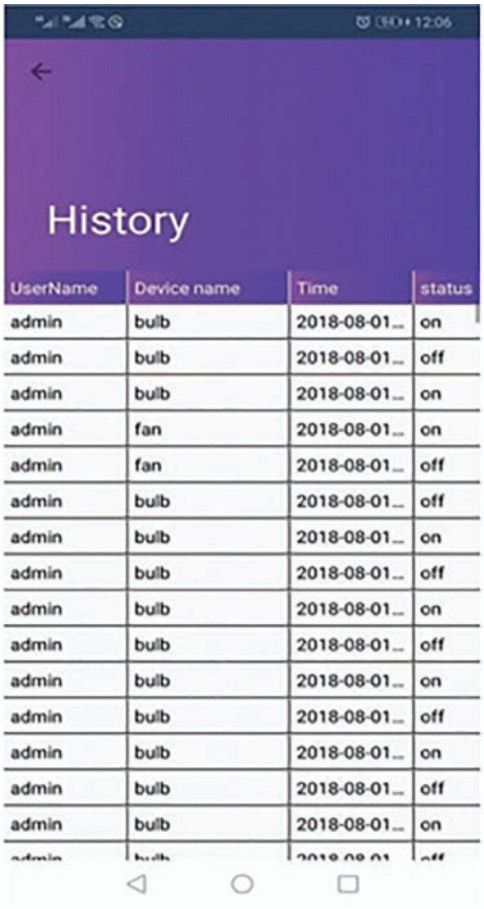
Temperature Sensor 27.812C

Light Sensor 0.0

Figure 11: Sensor data

Comprehensive Device History: The third tab on the home screen provides users with detailed insights into the history of their devices. The system maintains an extensive log of user activities, including changes in device states, accompanied by timestamped records. This log allows users to track the duration for which electronic devices have been active or inactive.

Efficient Energy Management: The system proactively notifies users when a device has been in an "ON" state for more than 2 h, encouraging responsible energy consumption. This feature promotes power monitoring, whether users are present in the room or there's excessive electricity usage, as illustrated in Fig. 12. Furthermore, the mobile app facilitates electricity consumption calculations based on the power usage data of electrical appliances and the corresponding timeframes.



History

UserName	Device name	Time	status
admin	bulb	2018-08-01...	on
admin	bulb	2018-08-01...	off
admin	bulb	2018-08-01...	on
admin	fan	2018-08-01...	on
admin	fan	2018-08-01...	off
admin	bulb	2018-08-01...	off
admin	bulb	2018-08-01...	on
admin	bulb	2018-08-01...	off
admin	bulb	2018-08-01...	on
admin	bulb	2018-08-01...	off
admin	bulb	2018-08-01...	on
admin	bulb	2018-08-01...	off
admin	bulb	2018-08-01...	on
admin	bulb	2018-08-01...	off
admin	bulb	2018-08-01...	on
admin	bulb	2018-08-01	off

Figure 12: History log

6 Experiments and Results

Fig. 13 illustrates the project’s functionality. The suggested framework is divided into a couple of scenarios. Remote access is handled by the first scenario for users who are not at home and make use of a Microsoft Azure cloud database. The user’s request is routed to the cloud using their APIs. The second scenario concerns individuals who are directly linked to a Raspberry Pi from within their houses. Fig. 14 shows how user queries are sent to the Raspberry Pi, and then to either the cloud. Without relying on the cloud, local processing allows for speedier processing.

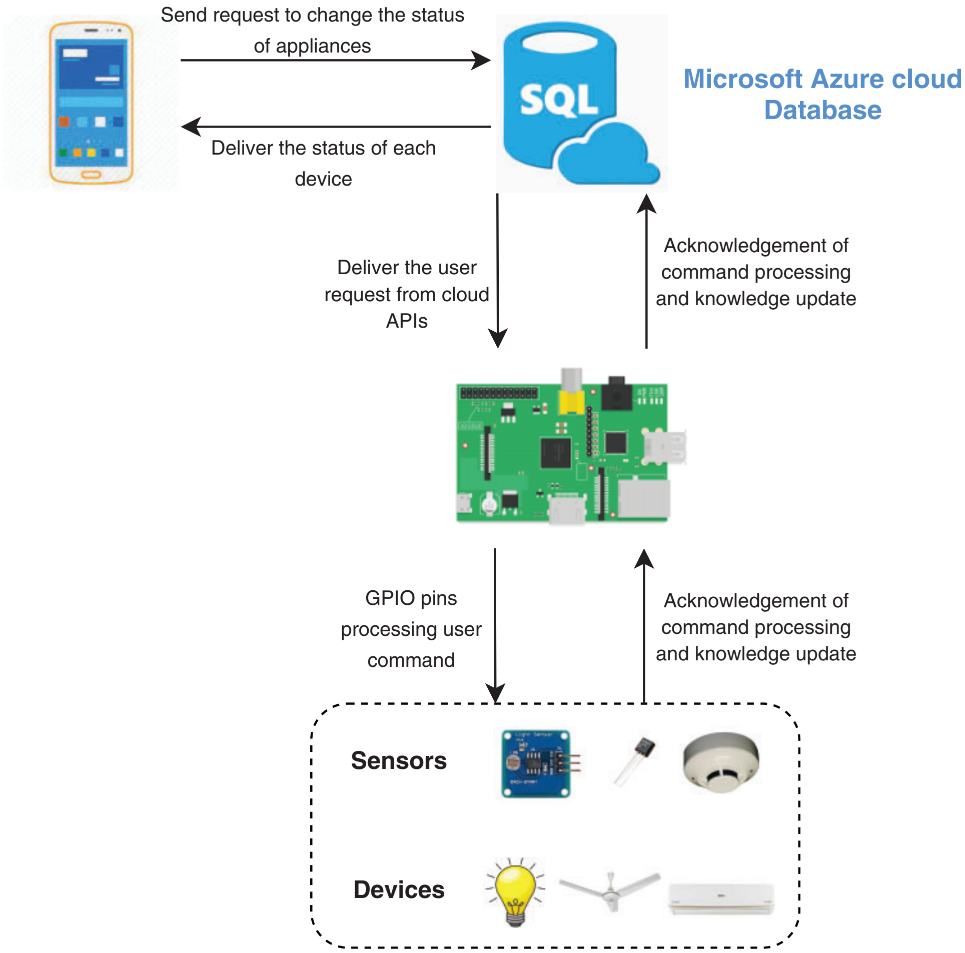

Send request to change the status of appliances
Deliver the status of each device
SQL
Microsoft Azure cloud Database
Deliver the user request from cloud APIs
Acknowledgement of command processing and knowledge update
GPIO pins processing user command
Acknowledgement of command processing and knowledge update
Sensors
Devices

Figure 13: Schematic diagram of connectivity with cloud database Microsoft Azure

6.1 Results

Extensive tests have been carried out utilizing powerful classifiers to assess the status of the appliance in smart home automation. support vector machine, random forest, decision tree, logistic regression, extra tree classifier, gradient boosting machine, convolutional neural network (CNN), long short-term memory (LSTM), and voting classifier (RF-CNN) are the classifiers used in these experiments. The documented data has been split into 70:30 training and test sets. All tests are run on a Dell PowerEdge server $T430$ GPU with a capacity of 2 GB, as well as $2x$ Intel Xeon eight-core CPUs running at 2.4 GHz and 32 GB DDR4 RAM. The studies are carried out in the Jupyter Notebook environment using the Python programming language and Anaconda. TensorFlow, sci-kit-learn, and Keras are used to build the classifiers. Table 2 shows the classifier's predictions of the "ON" and "OFF" classes for household appliances.

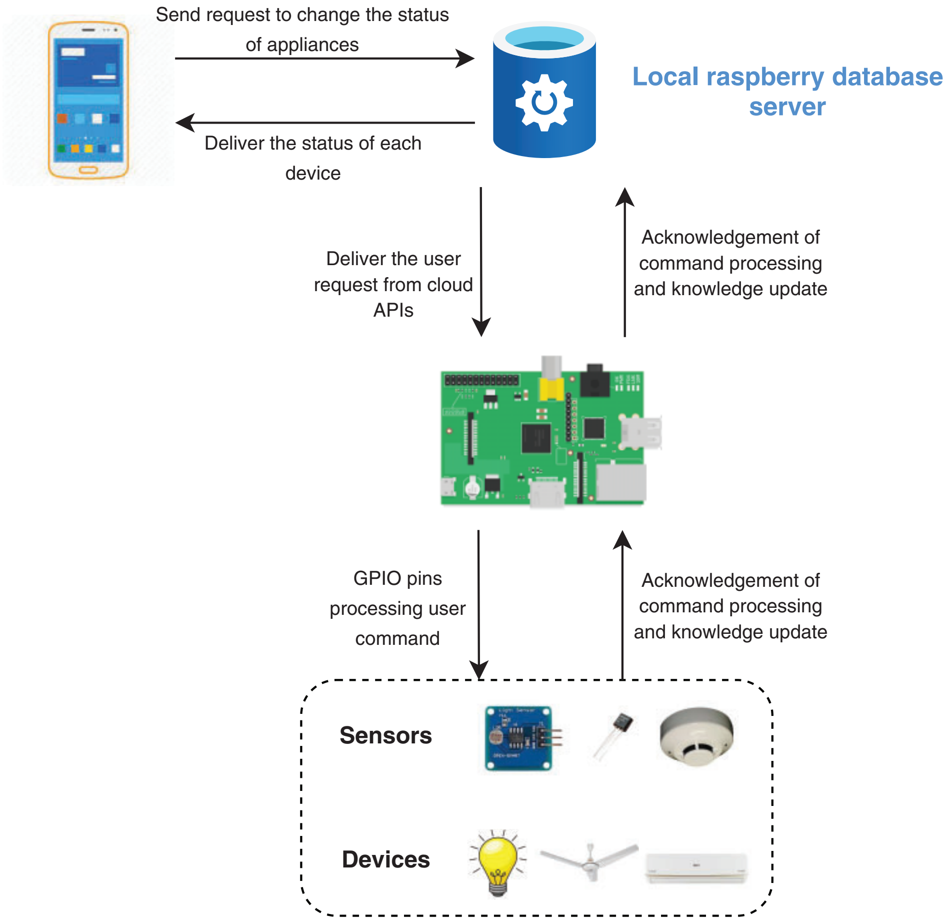

Send request to change the status of appliances
Local raspberry database server
Deliver the status of each device
Deliver the user request from cloud APIs
Acknowledgement of command processing and knowledge update
GPIO pins processing user command
Acknowledgement of command processing and knowledge update
Sensors
Devices

Figure 14: Schematic diagram of connectivity with a local database

Table 2: Experimental results of all classifiers

Classifiers	Accuracy	Precision	Recall	F-score
RF	93.9%	89.69%	90.67%	90.11%
SVM	91.4%	85.17%	87.74%	86.29%
LR	93.0%	91.29%	93.44%	92.54%
SGD	90.8%	83.33%	84.77%	84.32%
DT	92.7%	89.59%	90.57%	90.01%
GBM	87.7%	80.97%	83.35%	81.99%
ETC	93.8%	89.89%	90.87%	90.34%
LSTM	91.6%	89.17%	90.74%	90.09%
CNN	96.6%	95.57%	97.74%	96.45%
Voting classifier (RF-CNN)	98.9%	99.45%	99.89%	99.75%

SVM (91.4%), voting classifier (RF-CNN) (98.9%), gradient boosting machine (87.7%), decision tree (92.7%), Random forest (93.9%), logistic regression (93%), stochastic gradient descent (90.8%), and extra tree classifier (93.8%) achieved the highest accuracy values. Deep learning models such as CNN (96.6%) and LSTM (91.6%) are also used. It is worth mentioning that the voting of the two best-performing models, RF from machine learning and CNN from deep learning beats other models in predicting the state of household appliances, with a 98.9% accuracy rate. RF is computationally simple and often demonstrates superior performance due to its reliance on decision tree interpretation. However, the gradient boosting machine performed poorly in this circumstance, possibly due to challenges in tuning. Deep neural networks, including CNN, require ample training data to yield optimal results. When comparing CNN with RNN, CNN exhibits higher feature compatibility, RNN, on the other hand, excels at dealing with random input or output. LSTM works well with sequential data, whereas CNN investigates spatial relationships among features and works well with categorical data. As a consequence, CNN emerges as the best classifier for forecasting the status of domestic appliances and may be utilised to make sound decisions.

6.2 Significance of Proposed System

Combining Random Forest and CNN in an ensemble for classification tasks leverages the strengths of both models, resulting in improved accuracy, robustness, and the ability to handle diverse and heterogeneous data sources. Some useful aspects of using this ensemble are:

- **Feature Diversity:** Random Forest and CNN excel in extracting different types of features. Random Forest focuses on feature importance, while CNN learns hierarchical features from raw data. Ensembling these models ensures a more comprehensive coverage of feature space.
- **Reduction of Overfitting:** Ensemble methods often reduce overfitting because they combine multiple models that have different sources of errors. This can lead to more stable and reliable predictions.
- **Interpretability:** Random Forest provides feature importance scores, allowing you to interpret which features contribute the most to the classification decisions. This can enhance your understanding of the classification process and help identify important factors in the decision-making process.
- **Reduced Risk:** Ensembling can reduce the risk associated with relying on a single model. If one model fails to capture certain patterns or makes errors on specific instances, the ensemble's collective decision-making can compensate for these shortcomings.
- **Robustness:** Random Forest is known for its robustness against overfitting, making it a valuable component in the ensemble. CNN, on the other hand, can benefit from this robustness when dealing with limited data. The ensemble's overall performance can be more stable and reliable.

Furthermore, In order to check the significance of the suggested system, it is compared to previously suggested home automation system mock-ups. Several important elements are addressed while doing a performance comparison. Among these aspects, the type of devices or sensors used has a significant impact on installation and costs. Furthermore, the suggested system includes advantageous controls including concurrent sensor data, sensor data automatically recorded for optimization based on user preferences, remote accessibility, and a mechanism for system recovery. Table 3 compares home automation systems with our proposed system, highlighting the performance benchmarks considered for comparing them. Our suggested system separates itself from other systems for its extensive collection of features and functions, which are presented in Table 3. It makes it easier for the user to create a personalized home model and strategically deploy individual equipment as per the layout of the area to engage with electronic gadgets.

Table 3: Evaluation of the proposed system's performance relative to previous systems

Features	Automation systems								
	[49]	[50]	[51]	[52]	[53]	[54]	[55]	[56]	Proposed
App to make home prototype	×	×	×	×	×	×	×	×	✓
Device status data logging	×	×	×	×	✓	×	×	×	✓
Real-time sensors data display	✓	×	✓	✓	✓	✓	×	×	✓
Use of micro-processor (Raspberry Pi)	✓	×	✓	×	×	✓	×	×	✓
Internal network in case of gateway failure	×	×	×	×	×	×	×	×	✓
Sensors recent state recovery	✓	×	✓	×	×	×	×	×	✓
Light and fan intensity control using pulse wave modulation	×	×	×	×	×	×	×	×	✓
Climate control	×	×	×	×	×	×	×	×	✓
Smart lightening control	×	×	×	✓	×	×	×	×	✓
Use of blockchain security	×	×	×	×	×	×	×	×	✓
Predictive model based on usage of appliances and sensor data	×	×	×	×	×	×	×	×	✓
Use of ordinary electrical appliances	✓	×	✓	×	×	×	×	×	✓

7 Analysis of Risk Associated with IoT Smart Home

In the future, IoT-based smart home automation will be an area of utmost importance. With the ever-increasing number of smart devices in the homes including TVs and energy management systems, the need of an hour is to protect the households against possible security and safety threats and the impact on their lives. Reference [57] classified hazards into five broad categories including communication, information, and human-related hazards. Insufficient accountability of the internal gateway and the lack of system event logs for traceability are among the biggest software risks. The most serious negative impact is linked to poor authentication in the application programming interface (API). Unauthorized changes to system operations via mobile applications, providing access to end user's system resources without the appropriate authentication, have the greatest risk value. Unauthorized tampering or alteration of physical detectors are examples of hardware dangers. The inadequate distinction of user account credentials is classified as an information risk. Communication concerns include the server's possible deletion. Human-related hazards are associated with the use of weak or readily guessable passwords, as well as end users' susceptibility to fraud. Design phase security measures implementation in smart home system development is needed to address the privacy concerns related to such systems. A complete approach that addresses both security and privacy issues becomes critical. The next concern is how to build such a model, including the crucial aspects required to maintain security and privacy. The following stages are advised by this study to be included as minimum requirements in the model:

- During the transfer of personal data in the context of smart homes, identification and classification are performed.
- The key dangers and challenges to privacy and security are assessed and described.
- The objective is to discover and apply proactive, investigative, and reactive strategies to mitigate risks and enhance security measures.
- Developing a comprehensive plan to effectively manage information within smart homes while ensuring the protection of privacy.

Additional work is required to develop a framework to categorize the data collected, warehoused, updated, and shared by smart homes. This involves more work on developing a user-generated approach for managing data in smart homes, alongside integrating it with the digital ecosystems with which they interact. However, it is critical to recognize that the suggested approach is limited and requires improved privacy and security measures, particularly in cyber-network situations.

8 Conclusion

This research presents a comprehensive overview of a smart home automation project and its features. The main goal is to develop a user-friendly and customizable system that allows users to make informed decisions about the status of their appliances at home. The proposed system has two modes: admin and user. These modes are designed to perform specific tasks. The admin mode allows users to create a layout of their homes for automation tasks, while the user mode allows users to operate specific home appliances through a GUI. Users can manage the state of each device based on their previous usage patterns and data. A voting ensemble learning model based on random forest and convolutional neural networks (CNNs) aids in decision-making for detecting the "ON" and "OFF" status of household appliances. Additionally, the proposed framework leverages blockchain technology to authenticate IoT devices. The need for intelligent and flexible decision-making in home automation cannot be overstated in today's world. Risk analysis highlights the importance of incorporating security and privacy safeguards into the design of smart home systems. This home automation project also offers a simple, versatile, reliable, and affordable solution. Deep learning models will be used in the future to improve the system's decision-making efficiency. Two future research directions for the current proposed work are: (1) Integrate edge computing into the smart home architecture to process data closer to the source, reducing latency and bandwidth usage while improving real-time decision-making. (2) Evaluate the environmental impact of IoT-based smart homes, including energy consumption and electronic waste, and explore sustainable design options.

Acknowledgement: This research was funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2024R333), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Funding Statement: This research was funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2024R333), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Author Contributions: Study conception and design: Muhammad Umer, Nazik Alturki, Oumaima Saidani, and Amal Alshardan; data collection: Shtwai Alsubai, Raed Alharthi, and Reemah M. Alhebshi; analysis and interpretation of results: Muhammad Umer, Ali Kashif Bashir, and Shtwai Alsubai; draft manuscript preparation: Muhammad Umer, Ali Kashif Bashir, Nazik Alturki, Oumaima Saidani, and Amal Alshardan; funding: Raed Alharthi. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All the data related to the development of mobile applications and artificial intelligence techniques can be found upon suitable request from the corresponding authors or from the following link MUmerSabir/HomeAutomation (github.com).

Conflicts of Interest: The authors declare that they have no conflicts of interest.

References

1. Zeinab, K. A. M., Elmustafa, S. A. A. (2017). Internet of Things applications, challenges and related future technologies. *World Scientific News, 2(67),* 126–148.
2. Kang, B., Park, S., Lee, T., Park, S. (2015). IoT-based monitoring system using tri-level context making model for smart home services. *2015 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, IEEE.
3. Darianian, M., Michael, M. P. (2008). Smart home mobile RFID-based Internet-of-Things systems and services. *2008 International Conference on Advanced Computer Theory and Engineering*, Phuket, Thailand, IEEE.
4. Wang, I., Smith, J., Ruiz, J. (2019). Exploring virtual agents for augmented reality. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Glasgow, Scotland, UK.
5. Lobaccaro, G., Carlucci, S., Löfström, E. (2016). A review of systems and technologies for smart homes and smart grids. *Energies, 9(5),* 348.
6. Abdulrahman, T., Isiwekpeni, O., Surajudeen-Bakinde, N., Otuoze, A. O. (2016). Design, specification and implementation of a distributed home automation system. *Procedia Computer Science, 94,* 473–478.
7. Gill, K., Yang, S. H., Yao, F., Lu, X. (2009). A zigbee-based home automation system. *IEEE Transactions on Consumer Electronics, 55(2),* 422–430.
8. Al-Ali, A. R., Al-Rousan, M. (2004). Java-based home automation system. *IEEE Transactions on Consumer Electronics, 50(2),* 498–504.
9. Coskun, I., Ardam, H. (1998). A remote controller for home and office appliances by telephone. *IEEE Transactions on Consumer Electronics, 44(4),* 1291–1297.
10. Baudel, T., Beaudouin-Lafon, M. (1993). CHARADE: Remote control of objects using free-hand gestures. *Communications of the ACM, 36(7),* 28–35.
11. Kumar, P., Pati, U. C. (2016). IoT based monitoring and control of appliances for smart home. *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, India, IEEE.
12. Tharaniya soundhari, M., Brilly Sangeetha, S. (2015). Intelligent interface based speech recognition for home automation using android application. *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, India, IEEE.
13. Javale, D., Mohsin, M., Nandanwar, S., Shingate, M. (2013). Home automation and security system using Android ADK. *International Journal of Electronics Communication and Computer Technology (IJECCT), 3(2),* 382–385.
14. Piyare, R., Tazil, M. (2011). Bluetooth based home automation system using cell phone. *2011 IEEE 15th International Symposium on Consumer Electronics (ISCE)*, Singapore, IEEE.
15. Sriskanthan, N., Tan, F., Karande, A. (2002). Bluetooth based home automation system. *Microprocessors and Microsystems, 26(6),* 281–289.
16. Farshidi, S., Jansen, S., España, S., Verkleij, J. (2020). Decision support for blockchain platform selection: Three industry case studies. *IEEE Transactions on Engineering Management, 67(4),* 1109–1128.
17. Geneiatakis, D., Soupionis, Y., Steri, G., Kounelis, I., Neisse, R. et al. (2020). Blockchain performance analysis for supporting cross-border E-government services. *IEEE Transactions on Engineering Management, 67(4),* 1310–1322.
18. Ding, Q., Li, J. (2022). AnoGLA: An efficient scheme to improve network anomaly detection. *Journal of Information Security and Applications, 66,* 103149.
19. Yamin, M. M., Ullah, M., Ullah, H., Katt, B. (2021). Weaponized AI for cyber attacks. *Journal of Information Security and Applications, 57,* 102722.
20. Akleylek, S., Soysald, M. (2022). A new lattice-based authentication scheme for IoT. *Journal of Information Security and Applications, 64,* 103053.

21. Rana, A., Sharma, S., Nisar, K., Ibrahim, A. A. A., Dhawan, S. et al. (2022). The rise of Blockchain Internet of Things (BIoT): Secured, device-to-device architecture and simulation scenarios. *Applied Sciences, 12(15),* 7694.
22. Dave, M., Rastogi, V., Miglani, M., Saharan, P., Goyal, N. (2022). Smart fog-based video surveillance with privacy preservation based on blockchain. *Wireless Personal Communications, 124,* 1677–1694.
23. Meng, W., Lopez, J., Xu, S., Su, C., Lu, R. (2021). IEEE access special section editorial: Internet-of-Things attacks and defenses: Recent advances and challenges. *IEEE Access, 9,* 108846–108850.
24. Yeoh, W. Z., Teh, J. S., Chen, J. (2022). Automated enumeration of block cipher differentials: An optimized branch-and-bound GPU framework. *Journal of Information Security and Applications, 65,* 103087.
25. Wang, W., Xu, H., Alazab, M., Gadekallu, T. R., Han, Z. et al. (2021). Blockchain-based reliable and efficient certificateless signature for IIoT devices. *IEEE Transactions on Informatics, 18,* 7059–7067.
26. Abdulraheem, A. S., Salih, A. A., Abdulla, A. I., Sadeeq, M., Salim, N. et al. (2020). Home automation system based on IoT. *Technology Reports of Kansai University, 62(5),* 10788–10790.
27. Hoque, M. A., Davidson, C. (2019). Design and implementation of an IoT-based smart home security system. *International Journal of Networked and Distributed Computing, 7(2),* 85–92.
28. Satapathy, L. M., Bastia, S. K., Mohanty, N. (2018). Arduino based home automation using Internet of Things (IoT). *International Journal of Pure and Applied Mathematics, 118(17),* 769–778.
29. Pirbhulal, S., Zhang, H., Alahi, E., M., E., Ghayvat, H. et al. (2017). A novel secure IoT-based smart home automation system using a wireless sensor network. *Sensors, 17(1),* 69.
30. Chong, G., Zhihao, L., Yifeng, Y. (2011). The research and implement of smart home system based on Internet of Things. *2011 International Conference on Electronics, Communications and Control (ICECC)*, Ningbo, China, IEEE.
31. Haq, Z. U., Khan, G. F., Hussain, T. (2013). A comprehensive analysis of XML and JSON web technologies. *New Developments in Circuits, Systems, Signal Processing, Communications and Computers,* 102–109.
32. Hang, L., Kim, D. H. (2018). Design and implementation of intelligent fire notification service using IP camera in smart home. *International Journal of Control and Automation, 11(1),* 131–142.
33. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review,* 21260.
34. Christidis, K., Devetsikiotis, M. (2016). Blockchains and smart contracts for the Internet of Things. *IEEE Access, 4,* 2292–2303.
35. Breiman, L. (1996). Bagging predictors. *Machine Learning, 24(2),* 123–140.
36. Wright, R. E. (1995). Logistic regression. In: Grimm, L. G., Yarnold, P. R. (Eds.), *Reading and understanding multivariate statistics*, pp. 217–244. American Psychological Association.
37. Gardner, W. A. (1984). Learning characteristics of stochastic-gradient-descent algorithms: A general study, analysis, and critique. *Signal Processing, 6(2),* 113–133.
38. Breiman, L., Friedman, J., Olshen, R., Stone, C. (1984). Classification and regression trees. *Statistics/Probability Series*.
39. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics, 29(5),* 1189–1232.
40. Sharaff, A., Gupta, H. (2019). Extra-tree classifier with metaheuristics approach for email classification. In: *Advances in computer communication and computational sciences*, pp. 189–197, Springer.
41. Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena, 404,* 132306.
42. Yamashita, R., Nishio, M., Do, R. K. G., Togashi, K. (2018). Convolutional neural networks: An overview and application in radiology. *Insights into Imaging, 9(4),* 611–629.
43. Maksimović, M., Vujović, V., Davidović, N., Milošević, V., Perišić, B. (2014). Raspberry Pi as Internet of Things hardware: Performances and constraints. *Design Issues, 3(8),* 1–6.

44. Leccese, F., Cagnetti, M., Trinca, D. (2014). A smart city application: A fully controlled street lighting isle based on Raspberry-Pi card, a zigbee sensor network and wimax. *Sensors, 14(12),* 24408–24424.

45. Wei, Z., Qin, S., Jia, D., Yang, Y. (2010). Research and design of cloud architecture for smart home. *2010 IEEE International Conference on Software Engineering and Service Sciences*, Beijing, China, IEEE.

46. Brock, J. D., Bruce, R. F., Cameron, M. E. (2013). Changing the world with a Raspberry Pi. *Journal of Computing Sciences in Colleges, 29(2),* 151–153.

47. Quadri, S. A. I., Sathish, P. (2017). IoT based home automation and surveillance system. *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, IEEE.

48. Esmaeel, H. R. (2015). Apply android studio (SDK) tools. *International Journal of Advanced Research in Computer Science and Software Engineering, 5(5*).

49. Patchava, V., K., H., B., P. R. (2015). A smart home automation technique with raspberry Pi using IoT. *International Conference on Smart Sensors and Systems (IC-SSS)*, Bangalore, India, IEEE.

50. Jabbar, W. A., Alsibai, M. H., Amran, N. S. S., Mahayadin, S. K. (2018). Design and implementation of IoT-based automation system for smart home. *2018 International Symposium on Networks, Computers and Communications (ISNCC)*, Rome, Italy, IEEE.

51. Hadwan, H. H., Reddy, Y. (2016). Smart home control by using Raspberry Pi and arduino UNO. *International Journal of Advanced Research in Computer and Communication Engineering, 5(4),* 283–288.

52. Mahamud, M. S., Zishan, M. S. R., Ahmad, S. I., Rahman, A. R., Hasan, M. et al. (2019). Domicile-an IoT based smart home automation system. *2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, Dhaka, Bangladesh, IEEE.

53. Jabbar, W. A., Kian, T. K., Ramli, R. M., Zubir, S. N., Zamrizaman, N. S. et al. (2019). Design and fabrication of smart home with Internet of Things enabled automation system. *IEEE Access, 7,* 144059–144074.

54. Dey, S., Roy, A., Das, S. (2016). Home automation using Internet of Thing. *2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, NY, USA, IEEE.

55. Vishwakarma, S. K., Upadhyaya, P., Kumari, B., Mishra, A. K. (2019). Smart energy efficient home automation system using IoT. *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, Ghaziabad, India, IEEE.

56. Singh, H. K., Verma, S., Pal, S., Pandey, K. (2019). A step towards home automation using IoT. *2019 Twelfth International Conference on Contemporary Computing (IC3)*, Noida, India, IEEE.

57. Jacobsson, A., Boldt, M., Carlsson, B. (2016). A risk analysis of a smart home automation system. *Future Generation Computer Systems, 56,* 719–733.