Discover Internet of Things

RESEARCH

Open Access

An edge-enabled multimodal Smart Home Energy Management System using deep learning

Thinagaran Perumal[1], Yao Lu[1], Monica Ravishankar[2], Abhishek Sharma[3], Vinaytosh Mishra[4,5*] and Thompson Stephan[5]

*Correspondence:
Vinaytosh Mishra
vinaytosh@gmail.com
[1]Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Seri Kembangan, Malaysia
[2]Department of Computer Science and Engineering, GITAM School of Technology (GST), GITAM University, Bangalore, India
[3]Department of Electronics and Communication Engineering, LNMIIT, Jaipur, India
[4]Datta Meghe Institute of Higher Education and Research, Wardha, Maharashtra, India
[5]Thumbay College of Management and AI in Healthcare, Gulf Medical University, Ajman, UAE

Abstract
The rapid proliferation of Internet of Things (IoT) technologies has significantly transformed smart home environments. Heterogeneity of various IoT applications generally leads to interoperability requirements that needs to be fulfilled along with ensuring federated device configuration. In this paper, we propose a deep learning-based Smart Home Energy Management System (SHEMS) that integrates multimodal sensor data collected from diverse sources to resolve the management of heterogeneous IoT devices in smart homes. Here, all computations are done over the edge, and online servers are used for managing the whole system remotely. A prototype of the proposed system is built, and a comparative analysis is done for predicting the energy consumption of the smart devices using EMA, LSTM and ARIMA models over different Linux based SoCs which are used as local server. The system uses EMA for predicting the next day's power consumption, ARIMA for short term and LSTM for long term power consumption prediction. MQTT protocol is used as a performance metric for evaluating the reliability, speed and robustness of the proposed model. Experimental results demonstrate that the proposed system not only ensures robust and efficient energy prediction, but also facilitates scalable and secure smart home automation. The integration of edge computing with multimodal learning makes it a promising solution for future green and intelligent living environments.

Highlights

- Using different prediction methods for different time spans improved forecast accuracy.
- Local processing kept monitoring and control available even when remote services were disrupted.
- Lightweight communication supported reliable smart home operation with modest hardware.

Keywords Internet of Things, Smart Home Energy Management System, Multimodal, EMA, LSTM, ARIMA, MQTT protocol

1 Introduction

Recent advancements in wireless communication and IoT technologies, enabled the development of several consumer electronics devices that evolve around IoT-enabled home automation systems [1–5]. IoT refers to the interconnection of everyday objects such as sensors and embedded systems through the Internet, allowing them to communicate and exchange information seamlessly [6]. In smart home environments, IoT technologies aim to enhance immersion and pervasiveness by enabling smooth access to and interaction among a wide range of household devices [7]. This growing connectivity is expected to support the development of various applications that rely on the large volume of data generated by smart devices, while also offering improved services and user experiences for residents. Figure 1 depicts the general smart home environment which portrays the interaction of the home occupants with their home automation system and regulate the settings accordingly for efficient functioning. In general, the smart home consists of appliances, devices, and sensors that record data related to their respective tasks, day to today activities of the home occupants and the utility consumption of the devices. Each of these sensors captures a specific aspect of the home environment, and their outputs represent distinct data modalities. Integrating these heterogeneous sources results in multimodal data, which provides a more holistic and context-aware view of household activities and energy usage patterns. Unlike unimodal systems that rely on a single type of sensor signal, multimodal approaches can exploit the complementary nature of different data streams to enhance learning accuracy and robustness.

However, efficiently processing and analyzing this multimodal data from different sensors in real time remains a significant challenge. Edge type computation with machine learning seems to be a better possibility in dealing with the massive multimodal data. By shifting computation closer to the data source, edge computing enables low-latency processing, reduces the reliance on cloud infrastructure, and enhances data privacy by minimizing transmission of sensitive information. In the context of SHEMS, edge-enabled solutions allow for rapid, localized decision-making based on real-time sensor inputs [8]. Also, they will be able to get a detailed analysis of consumer usage and pattern for using different appliances via personal phone or computer. Figure 2 shows the

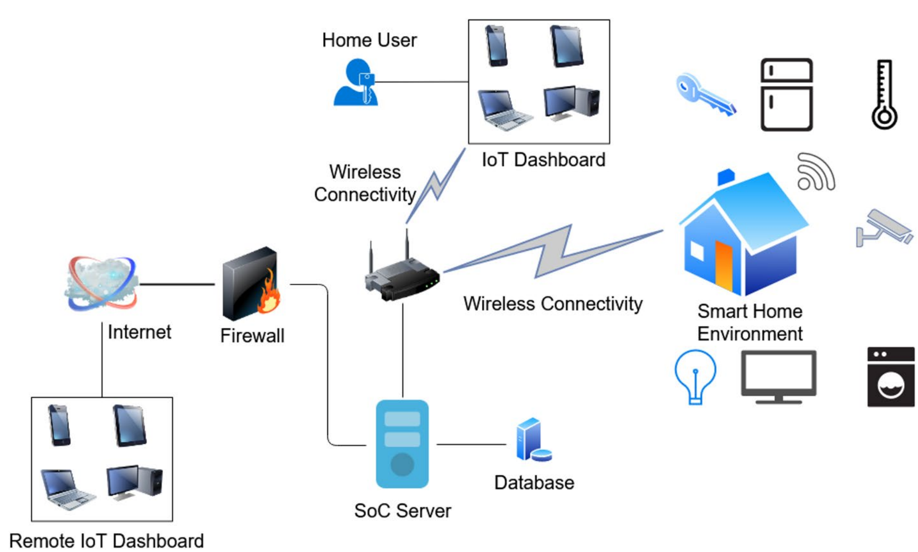

Fig. 1 Smart home environment

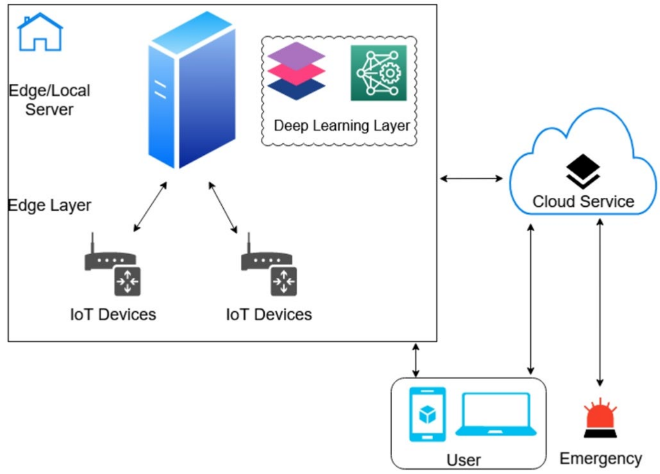

Fig. 2 Smart home energy management system architecture

high-level architecture of the proposed SHEMS, which is easily scalable relies heavily on the local server therefore making the system more robust and secure. Here the deep learning layer which is used for power consumption prediction forms a part of the local server rather than the cloud.

Rest of the paper is organized as follows. Section 2 discusses the related work followed by the system architecture in Sect. 3. Software architecture and its implementation is discussed in Sects. 4 and 5 respectively. Evaluation and testing phase is presented in Sect. 6 with final conclusions drawn in Sect. 7.

2 Related works

Energy consumption prediction aid in efficient resource management. More and more researches explore the combination of IoT infrastructure and Machine Learning (ML) algorithms to achieve energy consumption monitoring. Long Short-Term Memory (LSTM) has become a common method for energy consumption prediction in smart homes due to its excellent time series modeling capabilities [9]. In [10], Malik et al. compared various time series models such as Auto Regressive Integrated Moving Average (ARIMA) and Seasonal Auto Regressive Integrated Moving Average (SARIMA). ARIMA captures the timing and accuracy of time series data, SARIMA introduces external factors to enhance the accuracy of prediction. LSTM performs well in predicting energy consumption due to its ability to capture complex temporal dependencies. Lu et al. [11] predicted energy consumption with Temporal Kolmogorov-Arnold Transformer (TKAT), an attention-based model that combines the strengths of Kolmogorov-Arnold Networks (KANs) and Transformer architectures. TKAT draws on the Temporal Fusion Transformer (TFT) structure and combines the attention mechanism to capture both short-term and long-term dependencies. It simplifies temporal relationship modeling through the KAN structure and combines energy prediction with XAI to provide a more accurate and understandable prediction system for smart home energy management. The accuracy rate is 5.6% higher than the baseline model. The application of optimization algorithms in smart home energy management mainly focuses on equipment scheduling and energy efficiency optimization [12]. The application of optimization algorithms in smart home energy management mainly focuses on equipment scheduling and energy efficiency optimization. Enhanced Northern Goshawk Optimization (ENGO) introduces velocity ratio to dynamically adjust the search speed in different individuals and different stages. Youssef et al. optimizes energy usage strategies with ENGO [13], allowing users to reduce electricity bills without sacrificing comfort. However, ENGO cannot relocate rapidly when encountering a dynamic environment. The self-adaptive capability of Improved Bald Eagle Search (IBES) solved this problem well. The Gaussian mutation mechanism in IBES dynamically adjusts the search strategy according to the feedback during the search process, thereby maintaining search diversity and adaptability in a changing environment. Youssef et al. proposed IBES algorithm for dispatching and complex management to reduce electricity costs and peak load [14]. For the non-stationary and nonlinear problems existing in smart home energy consumption data, Jrhilifa et al. introduced Variational Mode Decomposition (VMD) [15]. It decomposes the original power time series into several Intrinsic Mode Function (IMF) with different frequencies. Each Intrinsic Mode Function (IMF) captures a specific oscillatory component of the original signal, effectively separating noise and preserving essential information. These IMFs are individually processed by Bidirectional Gated Recurrent Unit (BI-GRU) networks, and their respective prediction outputs are aggregated to generate the final power consumption forecast. This model was applied to electricity forecasting and achieved an MSE of 0.0038. Federated Learning (FL) is a distributed learning method that allows multiple devices to train on local data without uploading the data to a central server. This has important application value in the field of smart homes [12]. Moraes et al. [16] proposed a federated learning-based energy consumption prediction model. FL combined with CNN-LSTM, multiple devices can jointly train the model while retaining data privacy. The MAE of this method is verified to be 0.125 under low computing resources, and it can still run efficiently under low computing power. Deep Reinforcement Learning (DRL) is a method that combines reinforcement learning with deep neural networks. It can continuously optimize strategies by interacting with the environment in an uncertain environment. Aldahmashi et al. [17] proposed Proximal Policy Optimization (PPO), which combined deep learning with the proximal policy optimization framework to improve the stability of policy updates. Researchers models SHEMS as a Markov decision process, PPO collects state, action, and reward data by interacting with the environment, uses these experiences to estimate the advantage function, and introduces a clipping mechanism to stabilize the update strategy. Experimental results show that this method reduces electricity costs by 31.5% and improves the power factor from 0.44 to 0.9. Nowadays, SHEMS has gradually evolved from single device control to comprehensive optimization driven by user behavior modeling and multimodal data. Dasappa et al. [18] proposed a multi-sensor data fusion framework Energy Management for Smart Spaces (EMSS), which realizes human behavior modeling and personalized energy-saving recommendations based on "micro-moments" by integrating multi-source sensor data under the edge-cloud architecture. The system not only introduces high-level, medium-level, and low-level data fusion strategies, but also designs a synthetic data generator to support energy-saving scenario simulation and algorithm verification under multiple building types, and provide users with personalized energy-saving suggestions. But the limitations of this study are that it does not take into account user privacy and security and lacks extensive validation on real user data. In this paper, we propose a novel secured platform that leverages multimodal sensing data for accurate energy consumption prediction of IoT devices in smart home environments.

3 System architecture

Adhering to the aforementioned related works, the proposed system's architecture is as follows:

3.1 Home automation

- Sensor data collected from the individual sensors and sent to the local server (SoC), are forwarded to the centralized server.
- The SoC processes the data before sending it to the cloud. It retains the backup of the sensor data in the database and generate relevant results from raw data.
- Platform independent local and remote device monitoring and control system is required.
- Communication takes place using secure and lightweight protocols.
- The application allows clients to view energy consumption reports and the status of devices.

3.2 Home security

- Access restricted only the authorized person to enter the house using RFID verification or their smart devices, followed by face verification.
- For communication between devices and the server, internally and externally, data packets are encrypted.
- Clients are alerted in case of an emergency and assisted to contact an emergency service if needed.

3.3 IoT device and sensors

IoT devices and sensors are configured in the home environment setting. Sensors for temperature, humidity, light, current, voltage, PIR, LDR, RFID, camera, and ultrasonic are integrated using different MCUs and are used for capturing real-time indoor data. This environmental data is transmitted to SoC (local server) for processing, which is later sent to the system configured within a built-in repository for further processing. The system is complemented with a Python-based dashboard (PyQt) to monitor these indoor data in real-time [19]. Indoor environmental data gathered by the sensors are embedded on the chip and forwarded to the communication modules.

In this way, the proposed system is used for both monitoring as well as controlling the devices of the building. Besides this, cameras are only used for face verification and not to monitor activities inside the house as it may lead to privacy concerns. In addition to these sensors, several relay modules are used for controlling the home appliances. A single current sensor is used to determine energy consumption and generate more detailed results using bespoke algorithm, which is tailored over the deployed dashboard.

Table 1 Specification of micro-controller

Components	Description
Memory	32 Mb Flash
CPU clock	80 MHz
Digital pins	13
Analog pins	1
Energy	3.3 V
Communication	UART, SPI, I2C, and Wi-Fi

Table 2 Specification of SoC

	Processor	RAM	Clock
SoC 1	Quad core	1 GB	1.4 GHz
SoC 2	Dual core	520 MB	650 MHz

3.4 MCU system

The core component of the proposed system is an MCU with 32 Mb flash memory. They are used to control several IoT devices and send real-time data of different sensors to SoC through wireless communication. Table 1 shows MCU specifications.

3.5 Local server

In a proactive home automation and security system architecture, Linux based SoC is responsible for providing a gateway to all the sensors and micro-controllers. This helps to gain access to the online servers for automation functionality and process the raw data collected from sensors, thereby aids in optimal resource utilization. Users will be able to interact with SHEMS via online and offline mode since all the functionalities are implemented locally over SoC. Table 2 gives better insight for different SoC that are used to deploy the SHEMS. A detailed comparative analysis in terms of speed and accuracy is done in the later section of this paper.

4 Software architecture

The proposed system has four major modules which are as follows:

4.1 Data collection module

As discussed in the previous section, different sensors are used to monitor and capture the multimodal indoor environmental data. Raw data is first processed and transmitted to the online server to reduce delays and packet drop. Microcontrollers continuously read sensor data and transmit the readings to the communication module using MQTT protocol. Along with the published message, House ID, User ID, Device ID, Sensor ID, and Sensor value are also published, which aids to trace and monitor them in case of delays. The SoC publishes the control values for different actuators using the same method, as stated above.

4.2 Communication module

1) MQTT

MQTT protocol is used for communication between the devices and servers. It is light-weight, fast, and data can be sent using three QoS levels. For different application, different QoS level is used to send data. MQTT server is created on SoC. Microcontrollers subscribe to the respective rooms they are controlling. They publish the sensor data and parallelly read the data published by the SoC and work accordingly. SoC sends the data to the online server where the users can control and monitor the devices. Users can also perform the same task locally by connecting to the same network as devices.

2) Storage

The data is stored in local server as well as online server. Since the values are present in both the servers, it helps the system to revert back to its previous state, in case of system failures.

4.3 Prediction module

This module is responsible for predicting energy consumption using data from the current and voltage sensors. It is implemented using LSTM and ARIMA techniques to calculate the energy consummation for a week and the whole year. This module is also responsible for pre-processing the raw sensor data.

4.4 User application module

Two modules are developed. One is deployed for an online server, while the other for the local server. Both are platform independent. Qt designer is used for designing the software and is further implemented on Python; thus, it can be run on any device that supports Python. For those devices which supports C++, Qt also enables them to deploy the application in C++. For online platform, it is deployed using standard web development languages. An API key is required to authenticate the connection. Access to data like energy consumption is provided to the administrator to calculate the bill and notify users. Figure 3 shows the complete sequence diagram, which covers all the modules described above.

5 Implementation

A prototype is designed, built, and tested in the lab to validate the architecture of the proposed system. For predicting the power consumption of an individual household, UCI dataset [20] is used. This dataset was gathered in a house located in Sceaux (7 km

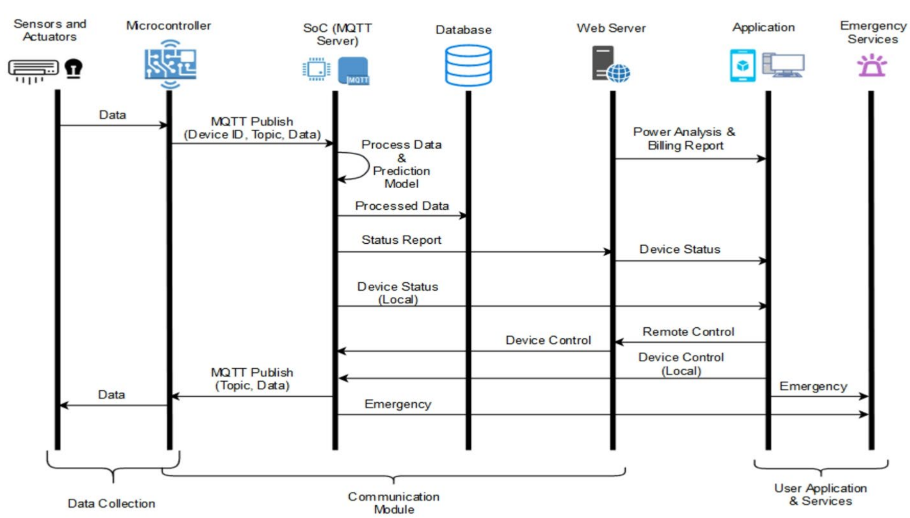

Fig. 3 Sequence diagram

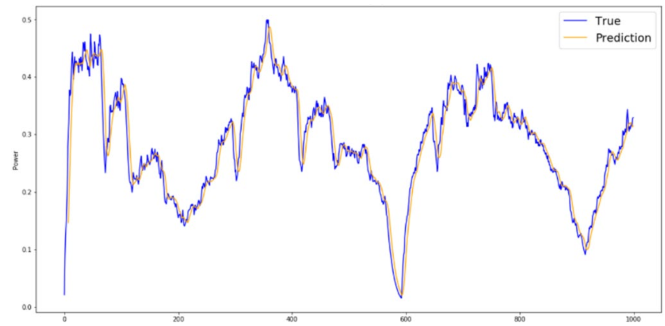

Fig. 4 Standard averaging

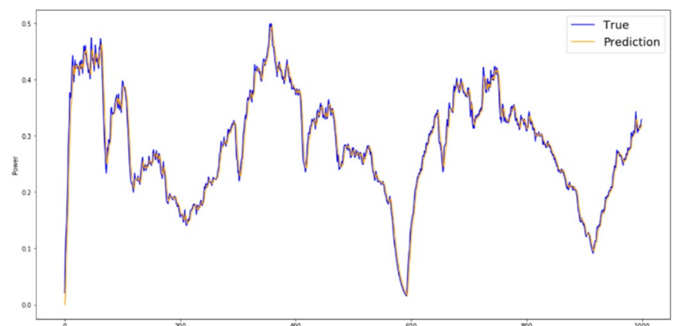

Fig. 5 Exponential moving averaging

from Paris, France) between December 2006 and November 2010 (47 months). It contains power consumption data with 9 attributes and respective dates with sampling rate of 1 s.

5.1 Next-day energy consumption

We have gathered the energy consumption data for each day and then used it for predicting the next day's energy consumption. Standard averaging is computed by taking mean of all the previous values, and output can be seen in Fig. 4.

Exponential Moving Average (EMA) is calculated using (1)

$$\mathrm{X_{t+1}} = \mathrm{EMA_t} = \gamma * \mathrm{EMA_{t-1}} + \ (1 + \gamma) * \mathrm{X_t} \tag{1}$$

where the initial value, $EMA_0 = 0$ and the contribution of new value is decided by the value of γ. The result of EMA is portrayed in Fig. 5.

The simulation results of Figs. 4 and 5 reveal that, EMA predicts the next day's energy consumption with fairly a good accuracy. But when it comes to predicting the energy consumption for multiple days, the values remain constant. To address this issue, standard averaging was used. But, since the experiments reveal that Mean square error (MSE) of standard averaging is 0.0006, while that for EMA is 0.00024, EMA is found to be optimal for predicting the next day's energy consumption.

5.1.1 General trend

Since ARIMA is used for the short-term forecasting, the same data pre-processing is used as described earlier. The data is divided into training dataset and test dataset. The training dataset is trained for 159 weeks and the test dataset for 46 weeks. After

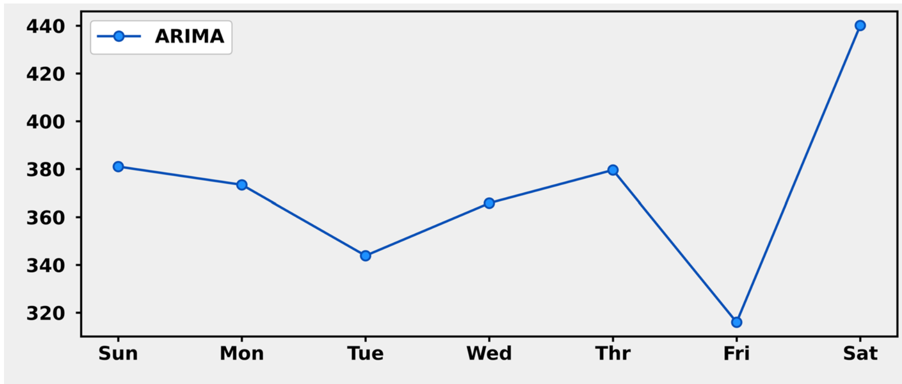

Fig. 6 RMSE for 46 weeks of test data

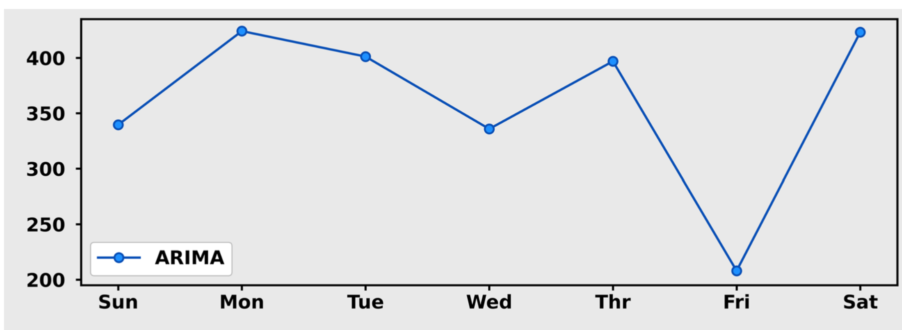

Fig. 7 RMSE for ten weeks of test data

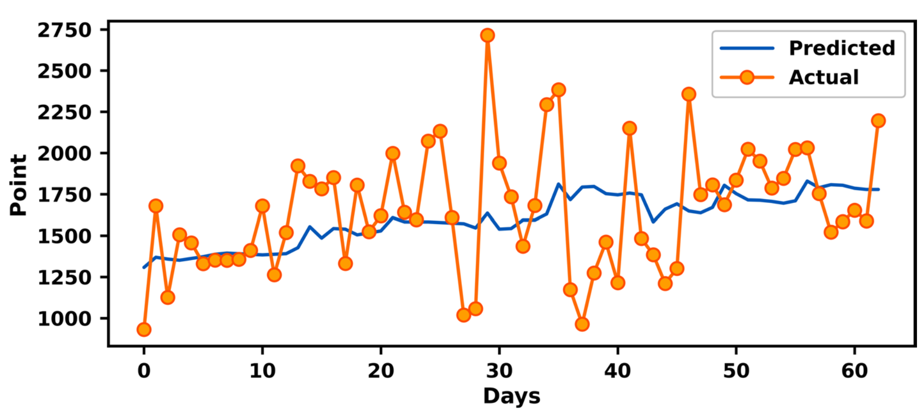

Fig. 8 Actual vs. Predicted using ARIMA

training the model, the predicted model gave RMSE (Root Mean Square Error) of about 373.169 as shown in Fig. 6, and when the test dataset was computed for nine weeks, RMSE was about 368.239 as shown in Fig. 7. When LSTM was deployed for the same task, it produced RMSE of about 377.311, which higher when compared with ARIMA. Figure 8 shows the actual and predicted active energy consumption. It can be inferred from the graph, that ARIMA model does not give accurate predictions wherever peaks are present.

In order to predict the energy consumption for the whole year, train data was considered for three years, while test data was considered for a year. The data was also scaled

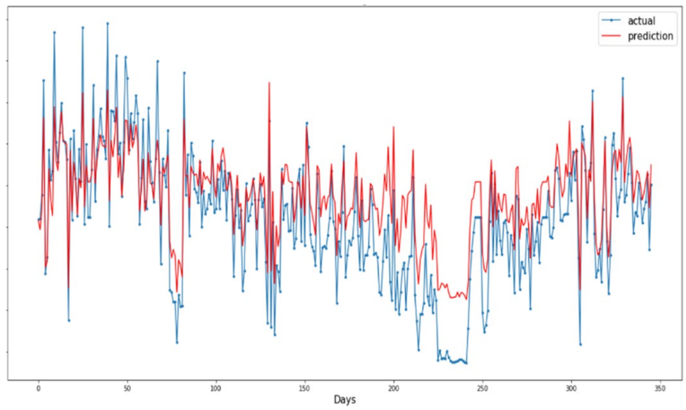

Fig. 9 Prediction using LSTM for a year

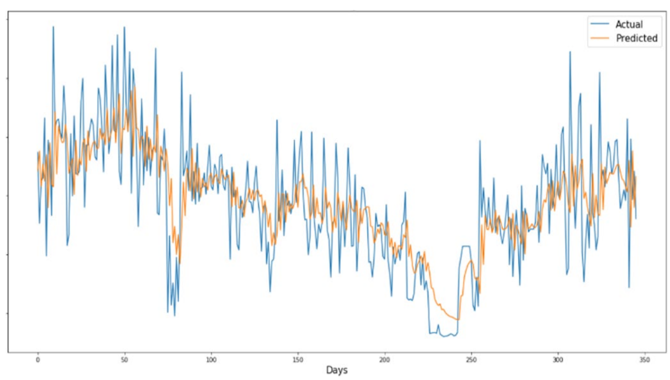

Fig. 10 Prediction using ARIMA for a year

Fig. 11 Model Loss in LSTM as we increase the training dataset

between 0 and 1. Prediction using LSTM had RMSE of 0.0457, while that of ARIMA had RMSE of 0.0771, as shown in Figs. 9 and 10. Though this proves that LSTM model predicts the peaks more accurately than ARIMA, the prediction of per hour data using LSTM shows loss with the increase in training data (Fig. 11).

5.2 User application and emergency services

All devices can be controlled using online and local server user interface. Users must log in to their accounts to control and monitor their devices. Since the computations are done on SoC, UI present in SoC has a feature to predict and generate graphs for energy consumption. Apart from this, it has other basic functionalities that are used to display the following: house occupants, emergency status and contact details, temperature and humidity. This information will be sent to the emergency service in case of any disaster, enabling them to retaliate and act. For instance, in case of gas leakage, UI also has an additional feature to enable the windows to open automatically and caution the users and neighbors accordingly.

Figure 12 shows the main window of the smart Home Automation and Security System (HASS) application. It is designed using Qt designer and integrated with Python using PyQt. It is lightweight and has all the features listed above. The UI on edge is kept minimal as SoC used for implementation have limited resources. Moreover, executing both prediction model as well as face detection model parallelly while running a high-end application can cause overheating of the board, eventually damaging it and affecting its performance. Figure 13 shows the energy consumption trend for a single household, which is further compared against multiple households (Fig. 14). Remote application for controlling the devices is portrayed in Fig. 15

6 Evaluation and testing

A set of criteria metrics were developed to evaluate the performance with respect to scalability, speed, and security.

6.1 MQTT

To evaluate the latency, throughput, and packet drops, MQTT protocol was deployed for network analysis. For relative assessment, consecutive packets in the range of 10, 100, 1000, 10,000, were sent to the local server with reliability of the following QoS levels: QoS 0, QoS 1 and QoS 2. The packet loss for different QoS levels is tabulated in Table 3. It is observed that mostly a very small number of packets are dropped using QoS-2, thus, it is used only during critical operations like access to enter the home. QoS-1 is used in

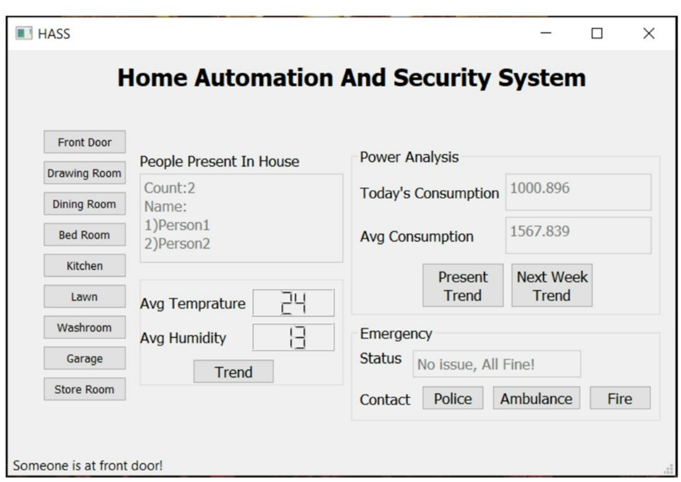

Fig. 12 Platform Independent application

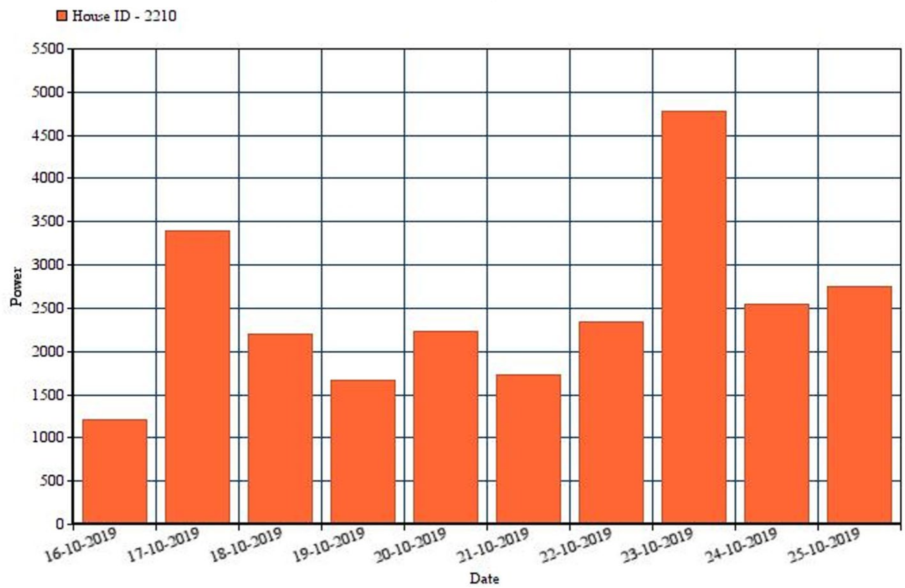

Fig. 13 Daily energy trend report for single household

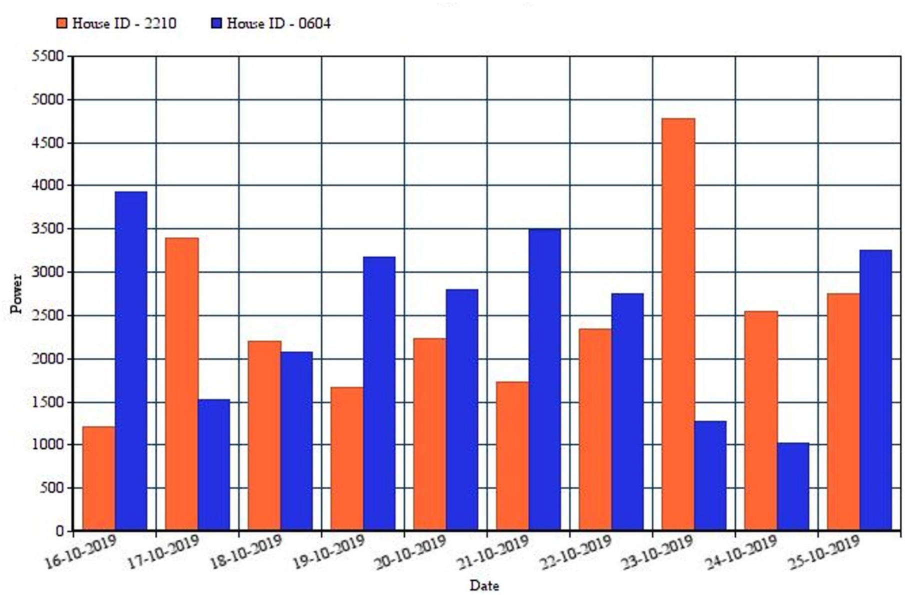

Fig. 14 Energy trend comparison of different households

situations like sending emergency requests along with the acknowledgment of receipt. QoS-2, in this case, will lead to delay in the transmission of multiple data, while QoS-0, cannot assure the successful receipt of the request. For sending sensor data, QoS-0 is deployed.

The proposed model is further improved by connecting all the microcontrollers to SoC in order to reduce the number of connected devices. SoC combines the data and then publishes it on the server. Moreover, it keeps the connection active for a certain period for sending multiple data. This reduces the network overload, thus saving up to 242 bytes per transmission. The improved model is also robust to online server crashes,

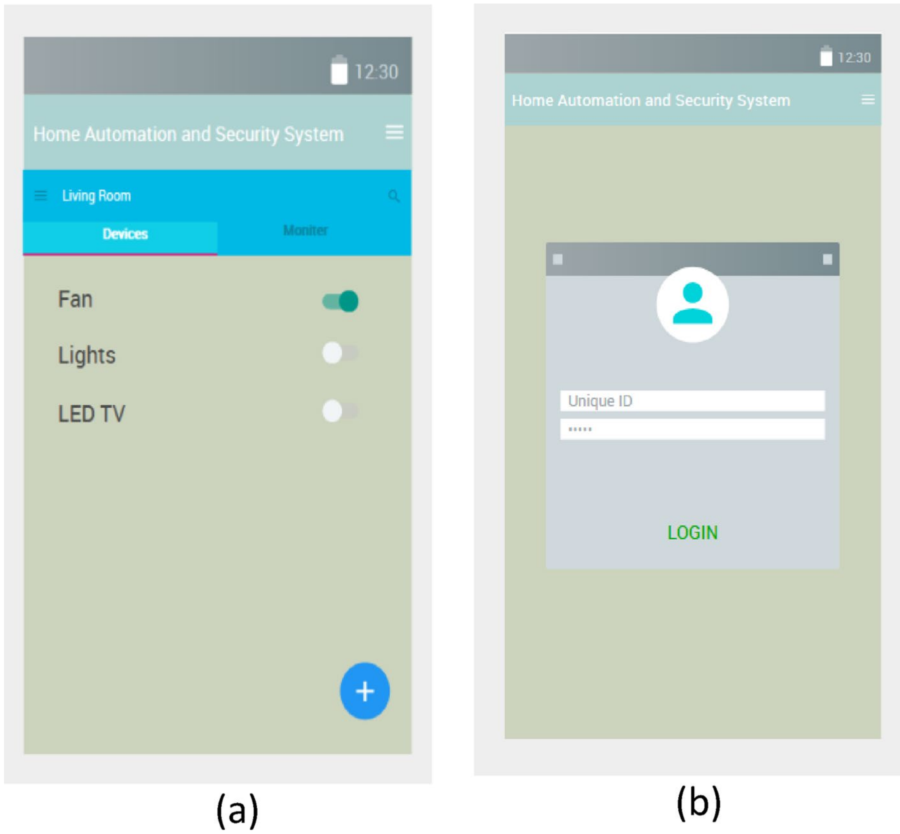

Fig. 15 **a** Remote control of devices **b** Remote Login for controlling devices

Table 3 Packets dropped for different QOS level

	Packets			
QoS	**10**	**100**	**1000**	**10,000**
0	0	0	0	0
1	0	0	0	0.016%
2	0	0	0.125%	0.0025%

by ensuring normal functioning of the local server. In case the user is not able to control devices from a remote location, they will be able to get the complete control once they are connected to the local network. The sensor data will be saved on the local server, and once the server is back to operation, it can sync in with the sensor data stored in the local server.

6.2 Prediction model

The prediction model is one of the core models that is implemented onboard. This model aids in reducing the network overload thus ensuring optimal resource management.

Figures 16 and 17 shows the comparison between the CPU and RAM usage, respectively, for face detection in SoC 1 and SoC 2. Comparison between the CPU and RAM usage for energy consumption using ARIMA model in SoC 1 and SoC 2 is depicted in Figs. 18 and 19. This infers that SoC 1 shows better results than SoC 2, as the former has a better processor and RAM. While the implementation on SoC 2 is done using it as SoC with no gate level coding, each process significantly consumes more time. For

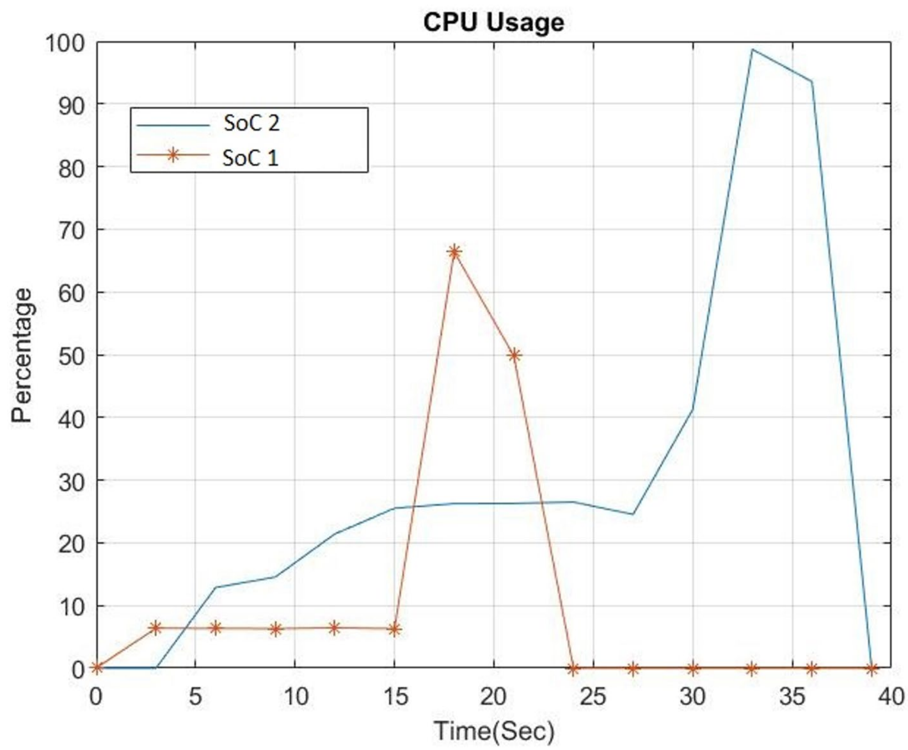

Fig. 16 CPU performance comparison for different SoC for face detection

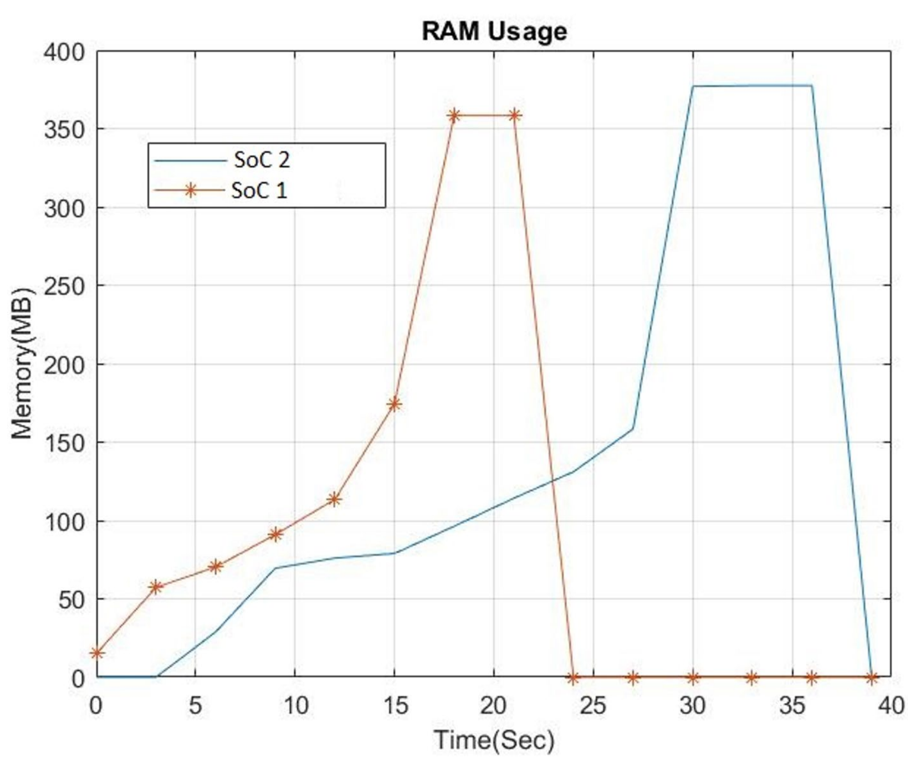

Fig. 17 RAM performance comparison for different SoC for face detection

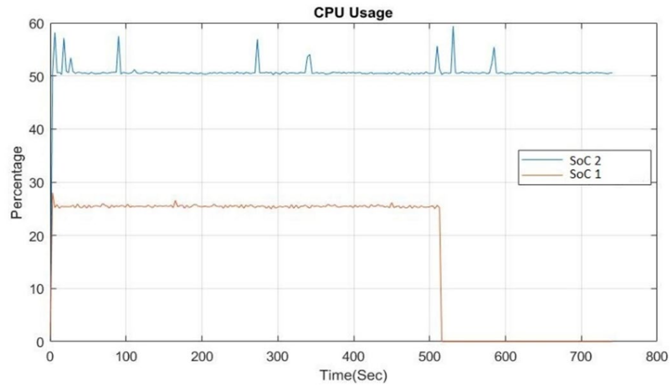

Fig. 18 CPU performance comparison for different SoC for ARIMA

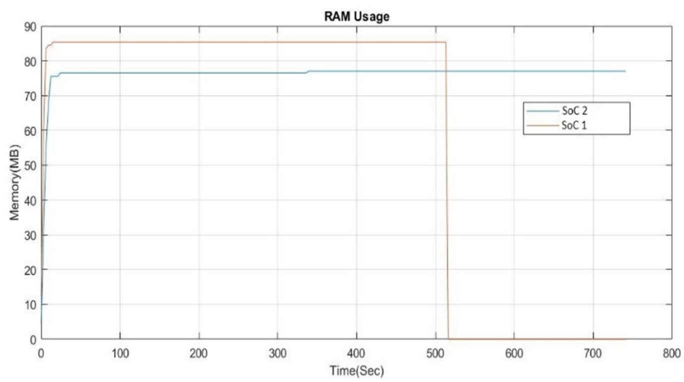

Fig. 19 RAM performance comparison for different SoC for ARIMA

Table 4 Total time taken for face recognition

	Total time (including Lib import time) (sec)	Encoding (sec)
SoC 1	6.4874	0.001611
SoC 2	9.5354	0.00262
Laptop	1.94472	0.0009942

Table 5 Total time taken for energy consumption prediction

	Total time (including Lib import time) (sec)	Encoding (sec)
SoC 1	511.66875	511.54790
SoC 2	735.89700	735.77322
Laptop	53.5244612	53.37189

some tasks when the overlay was created and integrated with Python, the performance level improved drastically. Table 4 shows the total time taken for face recognition, while Table 5 shows the time taken by the boards for predicting the energy consumption. In this way, the proposed system has also resolved the issue of predicting power consumption for short duration accurately by using LSTM based approach for long term prediction.

7 Conclusion

This paper presents a deep learning-based Smart Home Energy Management System (SHEMS), which is optimized so that it could be deployed over a local SoC-based server. The system integrates various environmental and device-level sensors to collect multimodal data, allowing for energy consumption prediction for the smart home environment using EMA, LSTM and ARIMA over different Linux based SoCs which are used as local server. The proposed system uses EMA for predicting the next day's power consumption, ARIMA for short term and LSTM for long term power consumption prediction. Experiments reveal that Exponential Moving Average (EMA) produced an optimized output with MSE of 0.00024. Moreover, this algorithm uses fewer resources and less time to compute as compared to other deep learning algorithms that were implemented, and at the same time, fails for predicting the power consumption for multiple days. Whereas, LSTM, which was used for predicting energy consumption for a year, had better accuracy with an RMSE value of 0.0457. This concludes that choosing an algorithm based on the time duration gives better overall results. Thus, the proposed system is efficient, secure, and responsive, making it well-suited for real-world smart home environments where edge computing and multimodal sensing are essential.

In the future, we plan to extend this system by integrating user activity recognition to further personalize energy control strategies. Additionally, incorporating deep reinforcement learning for dynamic scheduling and exploring privacy-preserving techniques will help improve adaptability, scalability, and data security in multi-user smart home scenarios.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1007/s43926-026-00345-3.

Supplementary Material 1

Supplementary Material 2

Supplementary Material 3

Acknowledgements
The authors express their sincere gratitude to Universiti Putra Malaysia, GITAM University, LNMIIT Jaipur, Datta Meghe Institute of Higher Education & Research, and Gulf Medical University, Ajman, UAE, for their continuous support and encouragement throughout this research.

Author contributions
TP conceived the system architecture and supervised the technical framework.YL contributed to model development and multimodal data processing.MR handled software implementation and prototype integration.AS contributed to hardware configuration and SoC-based deployment.VM performed manuscript revision, validation, and contributed to writing, editing, and supervising experimental evaluation.TS assisted with algorithm optimization and performance analysis.All authors reviewed and approved the final manuscript.

Funding
No external funding was received for conducting this study.

Data availability
The datasets used and/or analyzed during the current study available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests

The authors declare no competing interests.

Received: 17 November 2025 / Accepted: 24 April 2026

Published online: 26 May 2026

References

1. Jain N, Sharma S, Thakur V, Nutakki M, Mandava S. Prediction and analysis of household energy consumption integrated with renewable energy sources using machine learning algorithms in energy management. Int J Renew Energy Res (IJRER). 2024;14(2):354–62.
2. El Ghazi M, Aknin N. Optimizing Deep LSTM model through hyperparameter tuning for sensor-based human activity recognition in smart home. Informatica. 2024. https://doi.org/10.31449/inf.v47i10.5268
3. Bkheet SA, Agbinya JI, Khamis GSM. Advanced Deep Learning Approach for Smart Home Appliance Identification Using Recurrent Neural Networks with LSTM. IoT. 2024;5(4):835–51.
4. Gupta BB, Gaurav A, Attar RW, Arya V, Alhomoud A, Chui KT. LSTM based neural network model for anomaly event detection in care-independent smart homes. CMES-Computer Model Eng Sci. 2024. 140(3): 2689.
5. Abdel-Aty AH, Nisar KS, Alharbi WR, Owyed S, Alsharif MH. Boosting wind turbine performance with advanced smart power prediction: Employing a hybrid ARMA-LSTM technique. Alexandria Eng J. 2024;96:58–71.
6. Alasmari R, Alhogail AA. Protecting smart-home IoT devices from MQTT attacks: An empirical study of ML-based IDS. IEEE Access. 2024;12:25993–6004.
7. Ahmad HB, Asaad RR, Almufti SM, Hani AA, Sallow AB, Zeebaree SR. Smart home energy saving with big data and machine learning. Jurnal Ilmiah Ilmu Terapan Universitas Jambi. 2024;8(1):11–20.
8. Yedilkhan D, Smakova S. Machine Learning Approaches for Smart Home Device Recognition from Network Traffic. Procedia Comput Sci. 2024;231:709–14.
9. Lu Y, Vijayananth V, Perumal T. Smart home energy prediction framework using temporal Kolmogorov-Arnold transformer. Energy Build. 2025;335:115529.
10. Rock LY, Tajudeen FP, Chung YW. Usage and impact of the internet-of-things-based smart home technology: a quality-of-life perspective. Univ Access Inf Soc. 2024;23(1):345–64.
11. Nasir M, Muhammad K, Ullah A, Ahmad J, Baik SW, Sajjad M. Enabling automation and edge intelligence over resource constraint IoT devices for smart home. Neurocomputing. 2022;491:494–506.
12. Xu R, Khan S, Jin W, Khan AN, Khan QW, Lim S, Kim DH. A decentralized federated learning based interoperable and heterogeneity aware predictive optimization method for energy and comfort in smart homes environment. Appl Soft Comput. 2024;161:111689.
13. Youssef H, Kamel S, Hassan MH, Yu J, Safaraliev M. A smart home energy management approach incorporating an enhanced northern goshawk optimizer to enhance user comfort, minimize costs, and promote efficient energy consumption. Int J Hydrog Energy. 2024;49:644–58.
14. Youssef H, Kamel S, Hassan MH, Nasrat L, Jurado F. An improved bald eagle search optimization algorithm for optimal home energy management systems. Soft Comput. 2024;28(2):1367–90.
15. Jrhilifa I, Ouadi H, Jilbab A, Mounir N. Forecasting smart home electricity consumption using vmd-bi-gru. Energ Effi. 2024;17(4):35.
16. de Moraes Sarmento EM, Ribeiro IF, Marciano PRN, Neris YG, de Oliveira Rocha HR, Mota VFS, da Silva Villaça R. Forecasting energy power consumption using federated learning in edge computing devices. Internet Things. 2024;25:101050.
17. Aldahmashi J, Ma X. Real-time energy management in smart homes through deep reinforcement learning. IEEE Access. 2024.
18. Dasappa NS, Somu N. Multi-sensor data fusion framework for energy optimization in smart homes. Renew Sustain Energy Rev. 2024;193:114235.
19. Face Recognition pypi.org. 2020. https://pypi.org/project/face-recognition/. Accessed 02 Feb 2025.
20. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption. Accessed: 02 Feb 2025.

Publisher's Note