Sigma J Eng Nat Sci, Vol. 42, No. 5, pp. 1628−1645, October, 2024

Sigma Journal of Engineering and Natural Sciences
Web page info: https://sigma.yildiz.edu.tr
DOI: 10.14744/sigma.2023.00106

Research Article

Modeling and implementation of demand-side energy management system

Abdulkadir GÖZÜOĞLU[1], Okan ÖZGÖNENEL[2,*], Cenk GEZEGİN[2]

[1]*Department of Electrical and Energy, Tokat Gaziosmanpasa University, Tokat, 60250, Türkiye*
[2]*Department of Electrical and Electronics Engineering, Ondokuz Mayis University, Samsun, 55139, Türkiye*

ARTICLE INFO

Article history
Received: 23 February 2023
Revised: 07 June 2023
Accepted: 05 July 2023

Keywords:
CNN-LSTM Neural Network; Database; Deep Learning; Demand-Side Energy Management; Esp-01, Esp8266, Esp32; Future Smart Homes, And Smart Grids; Iot Network; Local Home Automation Server; Microcontroller; Monitor And Control; Smart Controller Board, Load Profiles; Wi-Fi Communication

ABSTRACT

In recent years, Internet of Things (IoT) applications have become across-the-board and are used by most smart device users. Wired Communication, Bluetooth, radio frequency (RF), RS485/Modbus, and zonal intercommunication global standard (ZigBee) can be used as IoT communication methods. The low delay times and ability to control homes from outside the building via the Internet are the main reasons wireless fidelity (Wi-Fi) communication is preferred. Commercially produced devices generally use their unique interfaces. The devices do not allow integration to form an intelligent home automation and demand-side energy management system. In addition, the high cost of most commercial products creates barriers for users.

In this study, a local home automation server (LHAS) was created subject to low cost. Smart devices connected to the server through a Wi-Fi network were designed and implemented. The primary purpose of the design is to create an IoT network to form an LHAS. The IoT network will learn the energy consumption behavior of users for future Smart Grids. The designed intelligent devices can provide all the necessary measurements and control of houses. The open-source software Home Assistant (Hassio) was used to create the LHAS. Espressif systems (ESP) series microcontrollers (μCs) were chosen to design intelligent devices. ESP-01, NodeMCU, and ESP-32, the most widely used ESP models, were preferred. A convolutional neural network (CNN)/long short-term memory (LSTM) neural network was designed, and analysis was performed to learn the consumption behavior of residential users.

Cite this article as: Gözüoğlu A, Özgönenel O, Gezegin C. Modeling and implementation of demand-side energy management system. Sigma J Eng Nat Sci 2024;42(5):1628−1645.

INTRODUCTION

The Internet of Things (IoT) network controls and monitors devices over the Internet [1]. IoT devices and communication networks aim to bring together independent devices created for human use and control them over the Internet [2]. Combining all devices in the communication network, providing a communication environment between the devices, and transferring useful information constitutes intelligent home systems [3].

***Corresponding author.**
*E-mail address: okanoz@omu.edu.tr

This paper was recommended for publication in revised form by Editor-in-Chief Ahmet Selim Dalkilic

Published by Yıldız Technical University Press, İstanbul, Turkey

In this study, smart devices were designed and implemented based on low-cost scenarios using Wi-Fi communication. The demand-side energy management system was created for real-time environmental monitoring and control. In addition, the system includes load planning and energy management applications for future smart grids.

The main contributions of this study are summarized as follows:

1. An advanced and intelligent IoT network architecture was designed and implemented for future smart homes. The main idea of the application is based on low-cost concerns.
2. An LHAS was created in this study. The preferred LHAS is Hassio. It is an open-source IoT environment. The LHAS is capable of Wi-Fi and message queuing telemetry transport (MQTT) communication methods. In addition, the server was arranged to perform fuzzy logic and deep learning applications.
3. As the brain of the application, Raspberry Pi was chosen for the implementation of Hassio. In contrast, the NVIDIA Jetson Nano developer kit was used for deep learning analysis. The designed intelligent modules for home appliances are based on ESP μCs.
4. The designed LHAS controls and monitors the whole house with the help of sensor measurements. The measured data are saved to the database in LHAS. The LHAS presents users with an advanced fuzzy logic-based control method.
5. The saved data especially alternating current (AC) measurements are essential for deep learning analysis. The data was used for extracting the consumption behavior of consumers. The consumption behavior information is essential to form load profiles of users and for load demand analysis in future smart grids.

The rest of this article is organized as follows. The related works are discussed in Section II. Section III introduces the methodology of designed LHAS, intelligent modules to form an IoT network, and gives information about the study's communication protocol, control method, and deep learning applications. Section IV gives exhaustive information on our solution and practical applications. The evaluation results are reported in Section V, and Section VI concludes the article.

RELATED WORKS

Numerous home automation applications have been reviewed, and various publications have been presented.

IOT Network for Smart Homes

In future smart homes, connecting all devices and sensors that collect information about the environment to the IoT network is essential.

Afifah et al. [4] presented combining sensor and mobile interface technology. Gebhardt et al. [5] created a Raspberry Pi-based intelligent home system. Jain et al. [6] proposed a study based on Raspberry Pi. The device was controlled due to the transmission of text messages via e-mail. The disadvantage of this method is that the microprocessors used are relatively costly, and text messages must be transmitted according to a specific rule [7].

Jabbar et al. [8] created an IoT network to form low-cost smart-home applications using NodeMCU was implemented in this study. Kane et al. [9] proposed a network schema for smart homes using LoRa 2.4 GHz. Franco et al. [10] emphasized an IoT-based load monitoring and recognition of activity systems for smart homes. Allifah and Zualkernan [11]carried out a smart home intelligent devices' secure ty ranking level. Illy et al. [12] created an IoT network for smart homes focusing on intrusion detection and prevention systems with machine learning applications. Ulloa-Vásquez et al. [13] proposed an intelligent network for smart home appliance consumption with high-resolution measurement.

Intelligent Module Design for Smart Homes

In recent years, ESP-based μCs have been widely used because of their low cost, ease of implementation, and flexibility in adapting to IoT networks. ESP μCs support Wi-Fi communication and allow firmware updates to communicate with different home automation servers. Tahir et al. [14] carried out a study using ESP8266 and ZigBee focused on managing the energy consumption values of intelligent buildings and the operation estimation of air conditioners. Přeučil and Novotný [15] proposed power-saving methods for Wi-Fi environment sensors using ESP-based μCs and ZigBee/MQTT communication. Akkurt [16] emphasized the indices of performance for predicting the heating load of smart buildings. Beki et al. [17] carried out an ESP32-based intelligent home application. Madhu and Vyjayanthi [18] proposed a NodeMCU-based intelligent home application that uses Android applications and Google Assistant for monitoring and control. Singh et al. [19] proposed an IoT-enabled control system and designed an intelligent system for smart homes. Midul et al. [20] designed and implemented an IoT-based smart meter for residential load by using ESP series μCs.

Fuzzy Logic Control for Smart Homes

Today, advanced control methods are preferred over classical ones in μC- or processor-based studies, and fuzzy Logic is one of them [21]. Krishna et al. [22] presented a study on applying the fuzzy logic control method in intelligent houses with various rules. Sevil et al. [23] proposed a fuzzy logic control system in smart homes for managing air conditioners. Ain et al. [21] used fuzzy logic control systems, and the controller's processing and application of home devices and sensor information are considered. Zhang et al. [24] proposed a study using the fuzzy logic control method in intelligent home applications. Jabeur et al. [25] designed a fuzzy logic-based controller to control lighting systems in smart homes with energy-saving functionality.

Paramathma et al. [26] developed an approach based on fuzzy logic as consumer-side management for smart homes. Roy et al. [27] proposed an energy scheduling system based on the fuzzy logic controller for a home energy management system.

Deep Learning in Home Energy Management System

There have also been studies on deep learning related to determining consumption behaviors [28]. The IoT network can apply advanced control methods within the building, determine and predict users' consumption behavior and create the load profiles of consumers. The process of determining consumption behavior can be realized using deep learning applications. Alhussein et al. [29] carried out short-term energy consumption values of houses that were estimated using CNN-LSTM deep learning. Zhou et al. [30] conducted a study to estimate electrical energy consumption using K-shape clustering and CNN-LSTM deep learning methods. Yan et al. [31] proposed a Hybrid CNN-LSTM neural network model to predict individual house energy consumption. Yu et al. [32] reviewed the deep reinforcement learning method for intelligent home energy management systems. Han et al. [33] created an IoT network and used short-term forecasting methods to apply deep learning to energy management systems for smart homes. Kodama et al. [34] proposed an intelligent algorithm for home energy management systems to predict energy consumption using a deep reinforcement learning method. Gao et al. [35] carried out an iteration optimization-based IoT network with learning functionality. Lu et al. [36] proposed a study to show the demand response of intelligent modules by using a deep reinforcement learning method to form an intelligent energy management system.

MATERIALS AND METHODS

This study designed and implemented an open-source IoT communication network and local home automation server, where intelligent devices provide data exchange. A fuzzy-logic controller was designed to monitor and control the processes of the devices in the building. The flow diagram of the proposed technique is shown in Figure 1.

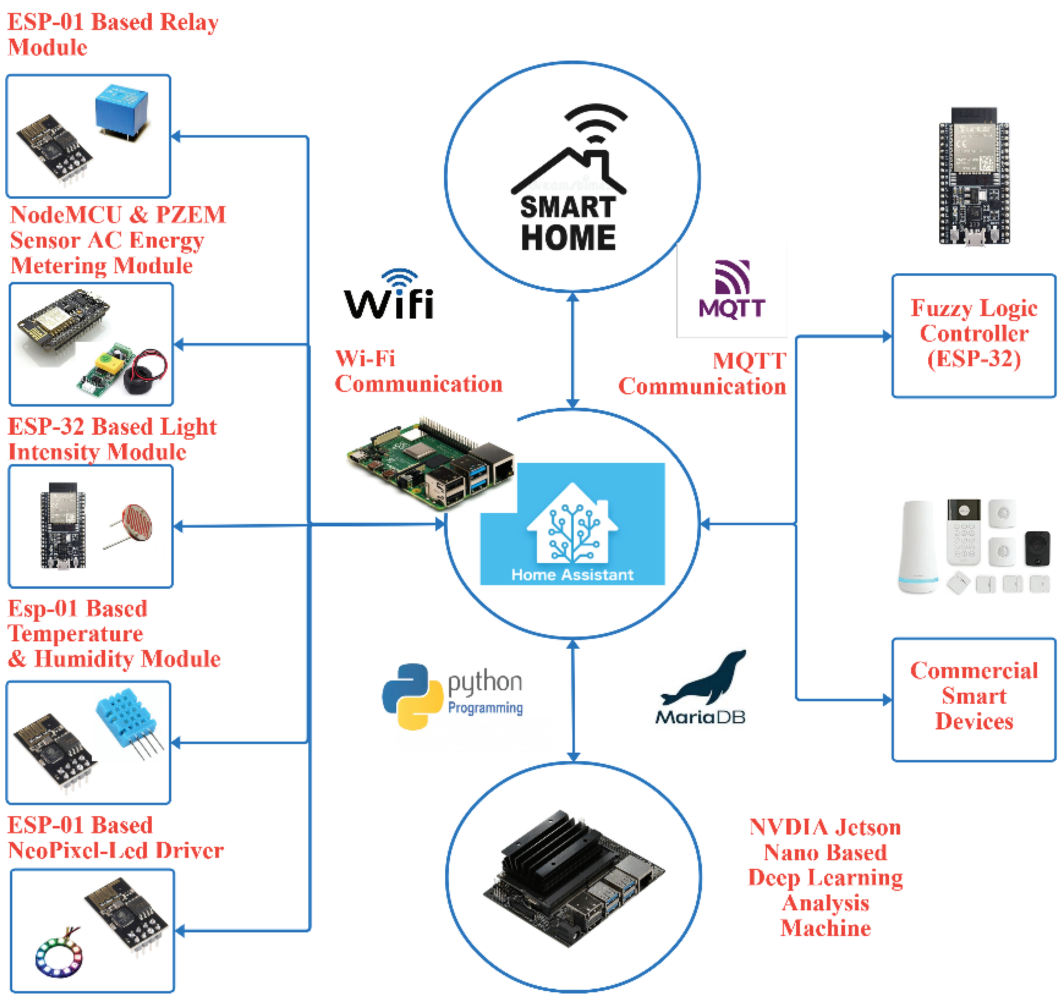


Figure 1. The flow diagram of the proposed technique.

ESP series μCs can provide communication compatibility with the LHAS. The MQTT communication protocol was used to adapt devices that cannot be integrated with the IoT network. The CNN-LSTM deep learning application was used to determine consumers' energy consumption behavior and create load profiles. A reconfigurable user interface was implemented in the automated system.

The proposed solution in this application can be summarized as follows:

The study aims to convert an ordinary house into an intelligent one with advanced control methods. First, the LHAS is created to form an IoT network. Hassio is chosen as the LHAS environment. Hassio performs whole management operations in the system. The communication integration with intelligent modules, designed in this study, and commercial ones are arranged by Hassio.

ESP-based innovative modules are designed. The intelligent modules are set to on-and-off control and measure temperature, humidity, ambient luminosity, and all AC measurement of the house. The module design can be reconfigured according to consumer needs.

The fuzzy logic controller is designed and simulated in MATLAB, a programming and numeric computing platform, and implemented using an ESP-32 kit and MQTT communication. A fuzzy logic controller increases the control resolution of the appliances.

A deep learning algorithm was formulated and implemented to obtain consumers' energy consumption behavior. Jetson Nano developer kit helped to perform deep learning tasks. The algorithm was prepared in Python. The deep learning algorithm used current, voltage, and power data during the learning process.

Local Home Automation Server

In innovative home applications, a home automation server is used for data transfer between devices, compatibility with different devices, and database operation. The designed intelligent devices in this study, communicate with the IoT network using Wi-Fi communication protocols. All the measurement and control signals reach the relevant device via the server, and the necessary control operations are performed. The LHAS is designed to provide user interfaces. In this study, the Home Assistant (Hassio) [37] was used because it is an open source and open to development. The server was installed on a Raspberry Pi minicomputer. These quiet, low-power computers provide benefits in terms of energy consumption. The designed IoT network does not require a connection to the Internet; only a router is sufficient. An internet connection is required when a remote connection is expected.

Cybersecurity is a topic that has been discussed in intelligent home applications since 2006 [38]. Innovative home device manufacturers have conducted various studies on virtual devices for the home, such as surveillance cameras, electricity meters, refrigerators, and door locks, which can be controlled via the IoT network [39]. In a study on cyber security, different connection methods and their grading are presented [11].

In the application designed in this study, an internet connection is not required to establish the IoT network. The router is sufficient to provide the necessary connection. Reaching LHAS from outside the local network and establishing a secure network for monitoring and control operations is necessary.

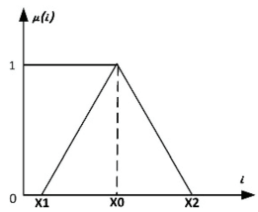
In our study, four different scenarios were tested in order to be able to connect to LHAs securely from outside the local network. The scenarios we have implemented are listed below:

- Home Assistant Cloud
- Using Static IP
- Online DNS Service
- DNS service created over the modem

Home Assistant Cloud is a particular service Hassio offers and requires a monthly payment. *Using Static IP* is one of the secure methods and requires a monthly payment. However, it may not be supported by some service providers. *Online DNS Services* can be created using applications such as DuckDNS in Hassio. In this method, which does not require a monthly payment, the security level is low because some ports are open on the router modem. *DNS service created over the modem* is provided by advanced modem routers such as KeenDNS. This application is secure and does not require monthly payments. Most modem manufacturers can provide the specified method in the future.

Fuzzy Logic Controller

The fuzzy-logic control system is a rule-based control method based on words and sentences [40]. The control structure consists of four parts. These can be listed as the input membership functions, verbal rules, decision algorithms, and output membership functions. The number of membership functions is determined by the measurement signals used as inputs and the number of verbal rules [24]. In this study, 35 rules were determined: 25 rules were determined for heating-cooling systems, five for lighting control, and five for energy consumption. As shown in Equation (1), triangular membership functions are used.



$$\mu_i = \begin{cases} 0, & i \le x1 \\ \dfrac{i - x1}{x0 - x1}, & x1 < i < x0 \\ \dfrac{x2 - i}{x2 - x0}, & x0 < i < x2 \\ 0, & i > x2 \end{cases} \tag{1}$$

Fuzzification was applied to the input values using triangular membership functions. The fuzzified output was obtained using input values and rules. The scalar values were obtained by applying defuzzification to the output values. These values were converted into a single output value using the center mean method [24]. Single-output values were obtained using Equation (2).

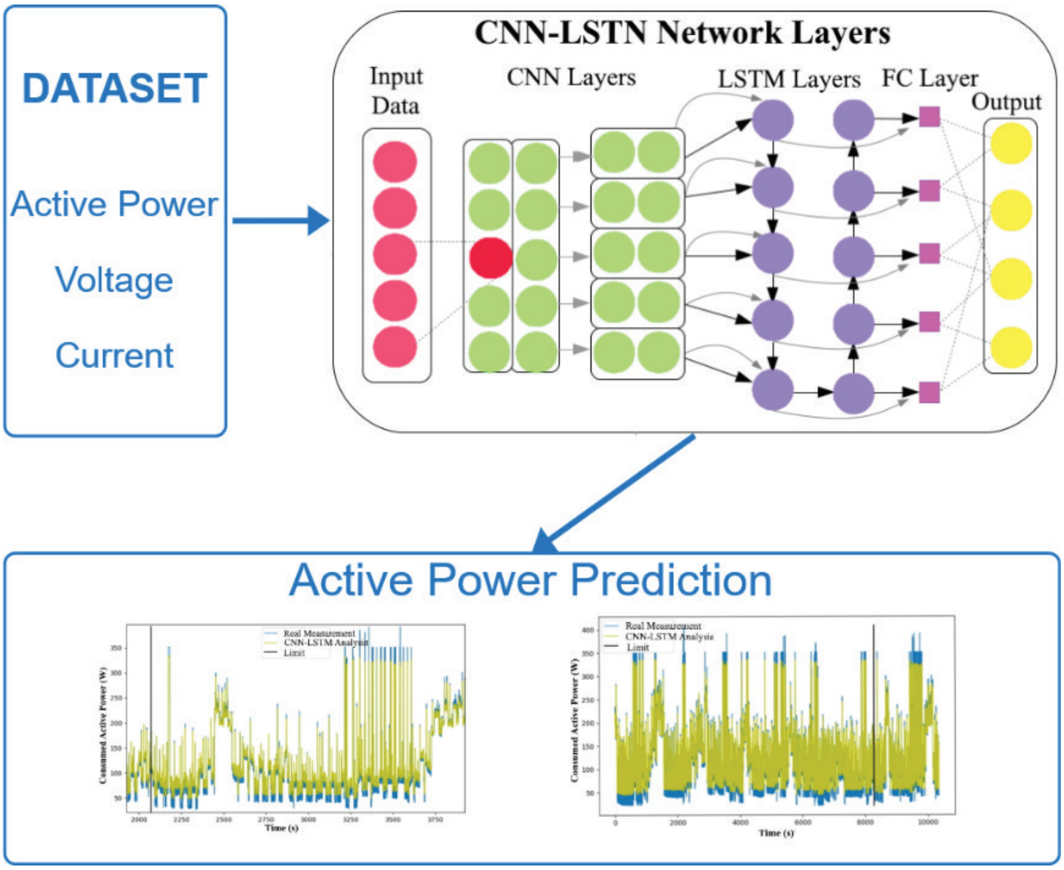


Figure 2. CNN-LSTM AI layers.

$$D = \frac{\sum_{i=1}^{k}[o_i \mu_i]}{\sum_{i=1}^{k}[\mu_i]} \tag{2}$$

In Equation (2), D denotes the defuzzification result, o output variable, k number of single result value, μ membership function value, and i index value.

Deep-learning Algorithm

In this study, a deep learning analysis was conducted using the CNN-LSTM method to determine electricity consumers' consumption behavior and create load profiles for different user types.

The layers of the CNN-LSTM model are shown in Figure 2. The CNN deep learning model was used to determine the stable periodic usage behaviors over time. The LSTM model, in contrast, is based on learning non-regular consumption patterns that occur instantly.

Convolutional Neural Network (CNN)

Convolutional Neural Network consists of layers that perform the learning process. The number of layers increases the learning level and inversely extends the processing time [41]. The discrete-time two-dimensional convolution equation is given by Equation (3).

$$C(i,j) = \sum_{x=-a}^{a} \sum_{y=-a}^{a} A(x,y) * w(i+x, j+y) \tag{3}$$

Here, A is a (2a+1)x(2a+1) matrix. Matrix C(i,j) represents the analysis result matrix. In addition, w(i,j) is a matrix with the input values.

In this study, CNN provided positive results in learning the daily repetitive and periodic consumption energy usage values. The CNN network used in the analysis was chosen for the five layers. The number of layers can be changed using the software.

Long Short-Term Memory (LSTM)

The consumption values of electricity users were recorded in a database with timestamps. During the analysis of time-dependent data in artificial neural networks, positive results are obtained using Recurrent Neural Networks (RNN) [42, 43]. However, because the amount of energy consumption data is considerable, the RNN is insufficient.

Long Short-Term Memory (LSTM) works more efficiently for deep learning analysis of big time-series data [44]. The gate functions and state transfer processes used for LSTM analysis are as follows:

$$f_t = \sigma(w_f * [a_{t-1}, x_t] + b_f \quad (4)$$

$$i_t = \sigma(w_i * [a_{t-1}, x_t] + b_i \quad (5)$$

$$o_t = \sigma(w_o * [a_{t-1}, x_t] + b_o \quad (6)$$

$$\tilde{c_t} = \tanh(w_c * [a_{t-1}, x_t] + b_c) \quad (7)$$

$$a_t = f_t * c_{t-1} + i_t * \tilde{c_t} \quad (8)$$

$$a_t = \tanh(c_t) \quad (9)$$

Here, f denotes the forget gate output, i the gate output, o the output gate output, at denotes the hidden layer at time t, xt denotes the input value at t, w is the connection weight parameter dd, b is the offset parameter value, and ct is the intermediate variable value.

The LSTM deep learning application was used to determine the consumption behavior of users that occurs instantly, either expressly or non-periodically. While CNN analysis focuses on learning consumption patterns that are repeated daily, weekly, monthly, and seasonal times, LSTM is used to learn instantaneous consumption patterns.

MQTT Communication Protocol

MQTT communication protocols are widely used, particularly in TCP/IP-based Wi-Fi applications. In this communication method, for connection and data transmission, unlike communication methods such as RF, Bluetooth, and inter-integrated controller (I2C), there is no requirement

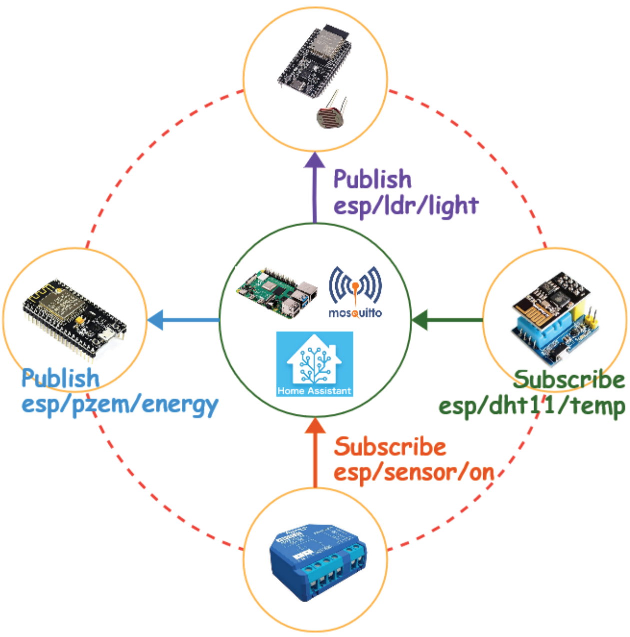


Figure 3. MQTT communication flow chart.

for a specific device name or IP address. Communication is provided by a software interface called the MQTT Broker. As shown in Figure 3, server and devices can receive and send data over a specific "Topic" using the Publish/Subscribe method.

Server and devices can subscribe to specific "Topic" and broadcast the data. MQTT Broker interfaces regulate two-way communication traffic without saving data [45].

ESP Series Microcontroller-Based Smart Device Design

ESP-based microcontrollers are preferred because they support Wi-Fi and Bluetooth communication protocols, integrate different servers, and are relatively inexpensive.

In this study, various modules have been designed for supplying Wi-Fi-based smart devices needed by smart homes with low-cost concerns. While these modules collect the necessary measurement information for the LHAS, the home devices that need to be controlled by the server are carrying out the switching operations.

The design is performed by ESP-01, NodeMCU, and ESP-32 development kits, which are widely used ESP μCs. The ESP-01 and NodeMCU kits are based on ESP8266 μC. The ESP-32 kit includes an ESP32-Wroom-32d μC.

The ESP-01 kit contains two general-purpose input-output (GPIO) pins. The kit supports transmission control protocol/internet protocol (TCP/IP), 802.11 b/g/n and Wi-Fi Direct (P2P) communication protocols. The kit, which has 32kB of flash memory, does not include an analog input unit [46].

The NodeMCU kit contains 17 GPIO pins internally but only one 10-bit analog input. The board also has general-purpose light emitting diode (LED), reset, and flash buttons [47].

The ESP-32 kit was based on an ESP32-Wroom-32D microcontroller. There were 16 analog-to-digital converters (ADCs) of 10 bits and 32 GPIO pins. The kit provides an advantage for long-line code uploads owing to its 520kB SRAM and 4 MB flash memory [48].

In this study, five modules were designed to allow all the appliances to communicate with the server in smart homes.

ESP-01 Based Relay Module

The relay module was designed using an ESP-01 kit, as shown in Figure 4. All appliances used in the building that works with the on-and-off working principle can be made intelligent and controlled via a server with the help of the designed module.

ESP-01 Based Temperature & Humidity Module

In smart homes, the LHAS needs measurement information of the environment to perform operations according to specific rules. As shown in Figure 5, a temperature and humidity measurement module based on the ESP-01 and DHT11 or DHT22 sensors was designed. DHT11 and DHT22 are low-cost and small-size sensors. The specific features and error rates of the sensors are shown in Table 1.

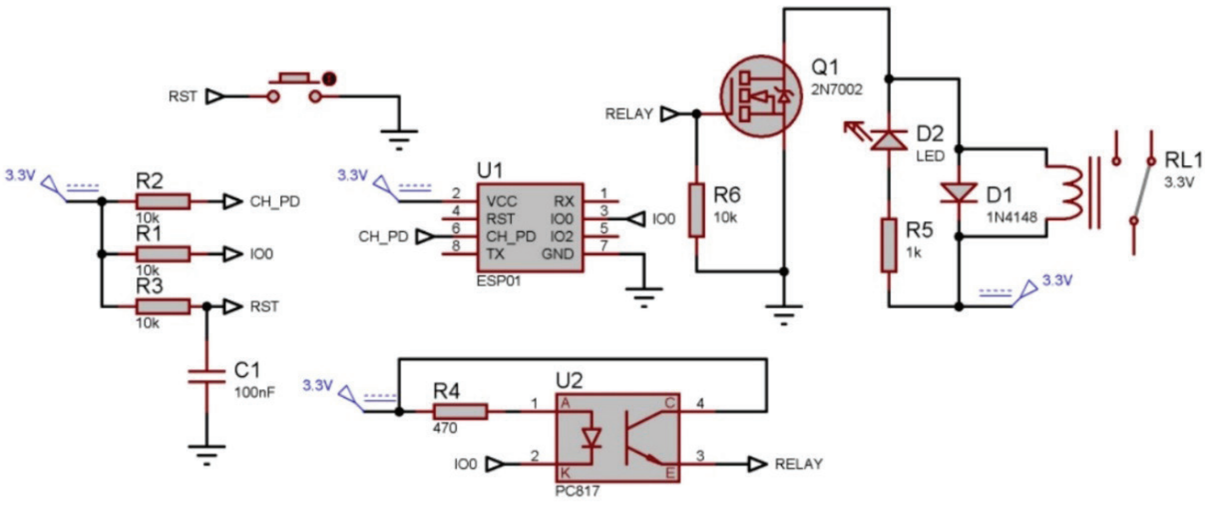


Figure 4. Relay module design.

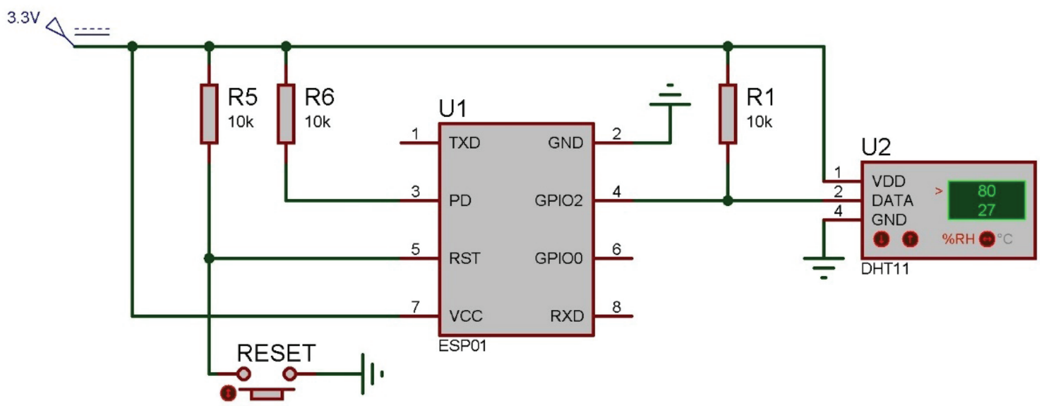


Figure 5. Temparature & humidity module design.

Table 1. The specific features of dht11 and dht22 sensors

The Features	DHT11	DHT22
Temperature measurement range	0-50^{0}C/±2^{0}C	40-125^{0}C/±0.5^{0}C
Humidity measurement range	20-80%/±5%	0-100%/±2-5%
Sampling rate	1Hz	0.5Hz
Dimensions	15.5mm*12mm*5.5mm	15.1mm*25mm*7.7mm
Operating voltage	3-5V	3-5V
Max. current	2.5 mA	2.5mA

Using the module in more than one place inside the home provides positive results in calculating the output value of the fuzzy logic controller.

ESP-32 Based Luminosity Module

The ambient luminosity information of the house should also be obtained for the automation processes of the LHAS. As seen in Figure 6, an ESP-32-based module was designed. The ambient luminosity is measured by the light-dependent resistor (LDR) connected to an ADC pin of ESP-32.

One analog input was sufficient to measure the luminous flux. Therefore, it is possible to use the NodeMCU kit.

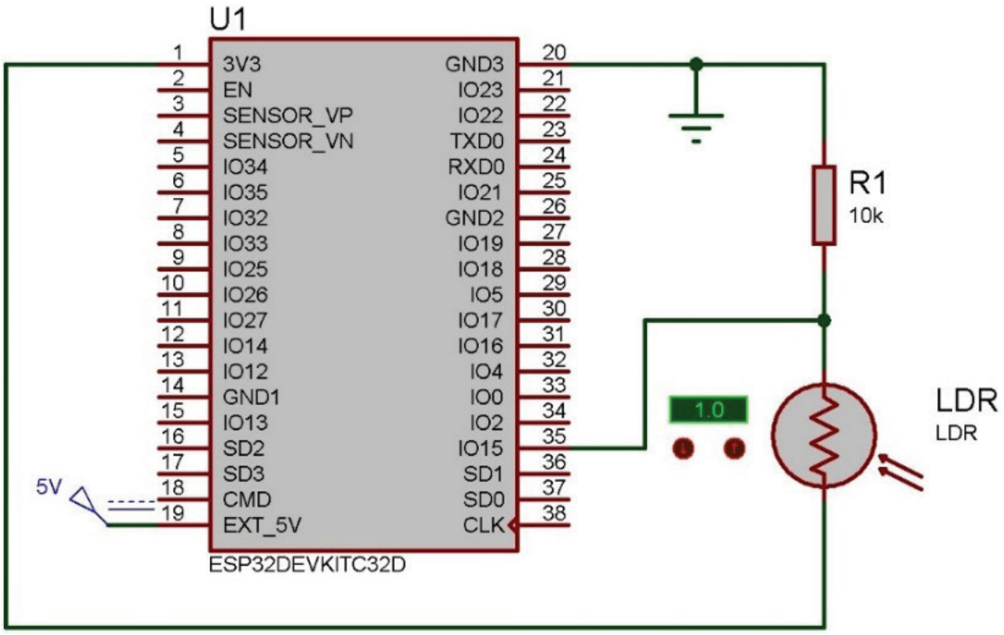


Figure 6. Luminosity module design.

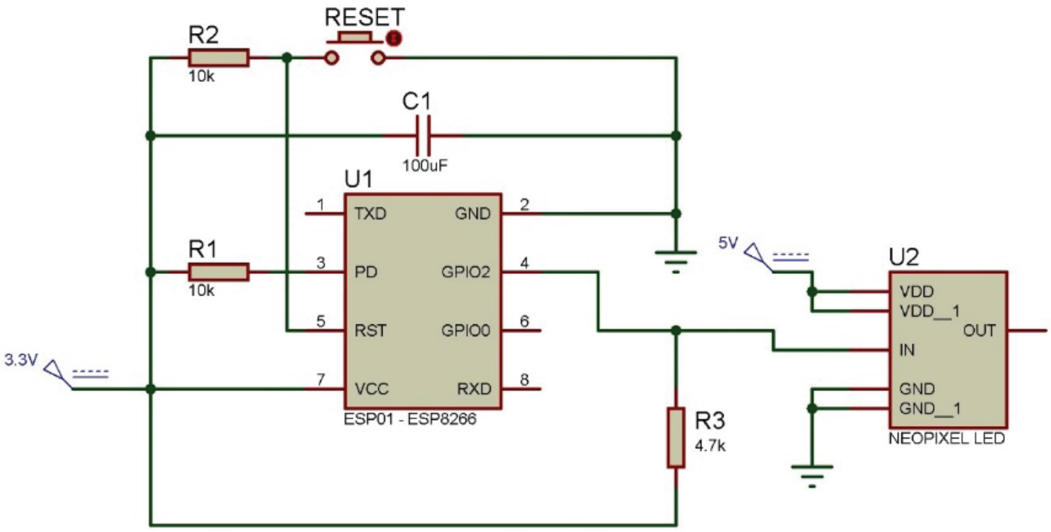


Figure 7. NeoPixel led driver module design.

However, this module was also designed as a fuzzy logic controller. Fuzzy logic functions and rules were embedded in the ESP-32 kit.

ESP-01 Based Neopixel Led Driver Module

Recently, the use of LEDs in lighting systems has become widespread. LEDs are preferred because of their low power consumption, heat-dissipation potential, and long life. Red, green, blue (RGB) LEDs can change color tones, and simple LEDs work as a single color. However, NeoPixel LEDs offer more advanced color effects to users. Owing to the WS2811 driver included in this type of LED, the desired LED on the strip can be operated using a data cable in different colors and formats.

In this study, as shown in Figure 7, a module was designed in which NeoPixel LEDs can be connected to the LHAS and controlled by the user interface.

NODEMCU and PZEM Sensor Based AC Energy Measurement Module

An AC energy measurement module was designed, similar to the relay module. The smart device comprises a NodeMCU, a Pzem AC measurement sensor, and a relay module.

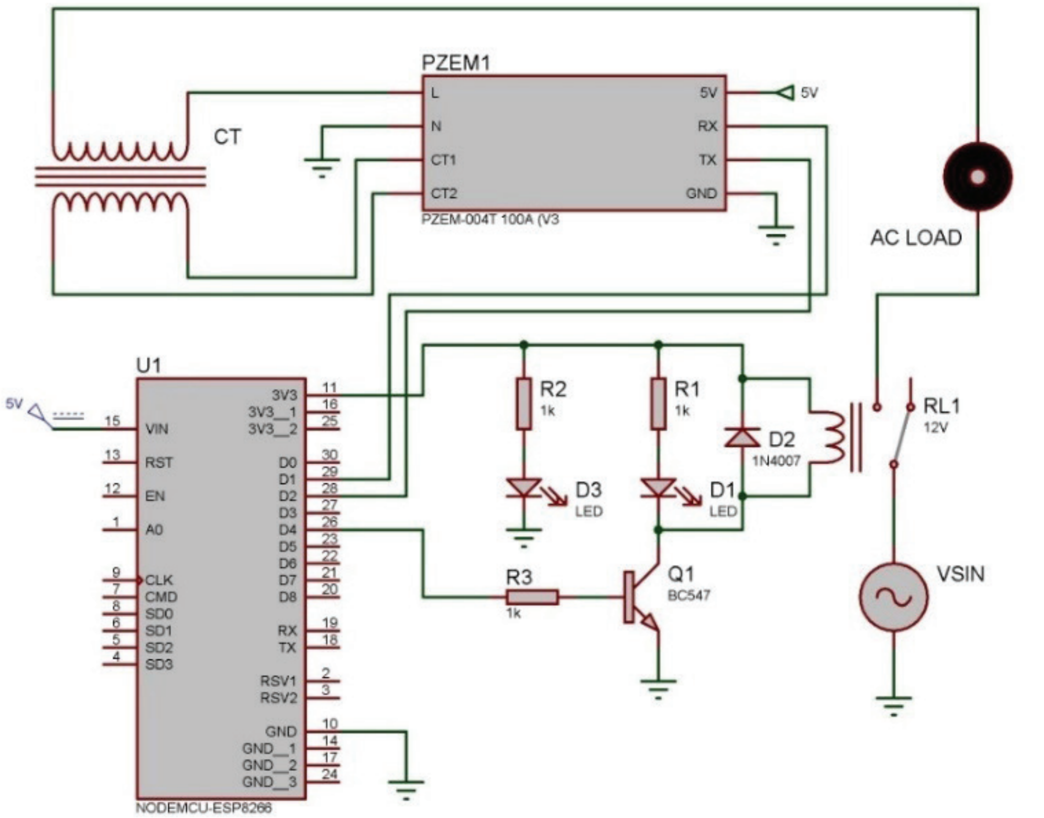


Figure 8. AC energy measurement module design.

Pzem 004t v3.0 is a low-cost, alternating current (AC) voltage, current power, and energy measurement sensor. It communicates using a transistor-transistor logic (TTL) interface over a Modbus-remote terminal unit (RTU)-like communication protocol. Pzem 004t sensor is operating under 80-260V AC, 100A (max: 23kW) conditions. The sampling resolution of the sensor is 16-bit. AC voltage, current, active power, power factor, frequency, and energy consumption measurements can be obtained with requested time intervals sensitively [49]. The designed circuit diagram is shown in Figure 8.

AC measurement is essential to monitor the energy consumption amount by users. In contrast, energy consumption values are used by deep learning applications for further analysis. It is crucial to determine users' consumption behavior and calculate the load profiles of different consumer types.

Experimental Applications

In this study, a LHAS, intelligent devices, different communication protocols integrations, and CNN/LSTM deep learning applications implemented by creating an experimental setup. The general hardware structure of the management system is shown in Figure 9.

Implementation of Local Home Automation Server

Hassio is used as the LHAS. Sample user interface views of Hassio can be seen in Figure 10. The Raspberry Pi 4 model B was chosen as the minicomputer on which the server can operate.

Because Hassio is an open environment for development, it allows the integration of ESP-based kits and different commercial smart devices.

The house's automation/scenario processes, database applications, and user interface design were performed on the Hassio. The designed server/user interface is accessed via a specific Internet Protocol (IP) using a Personal Computer (PC) and web browser. Secure control/monitoring from outside the local network is also tested.

ESP-32 Based Fuzzy Logic Controller

Unlike classical control methods, fuzzy-logic controllers can make higher-resolution decisions. Therefore, this study used fuzzy logic as the primary automation controller.

The rules of the controller were determined, and the membership functions were tested in MATLAB fuzzy logic toolbox. Some sample views of fuzzy logic design in MATLAB are shown in Figure 11. The designed controller was created in C language and uploaded to the ESP-32 kit.

Figure 9. The hardware part of the energy management system. From left to right; LHAS, deep learning, fuzzy logic, smart plugs, temperature, humidity measurement, neopixel driver.

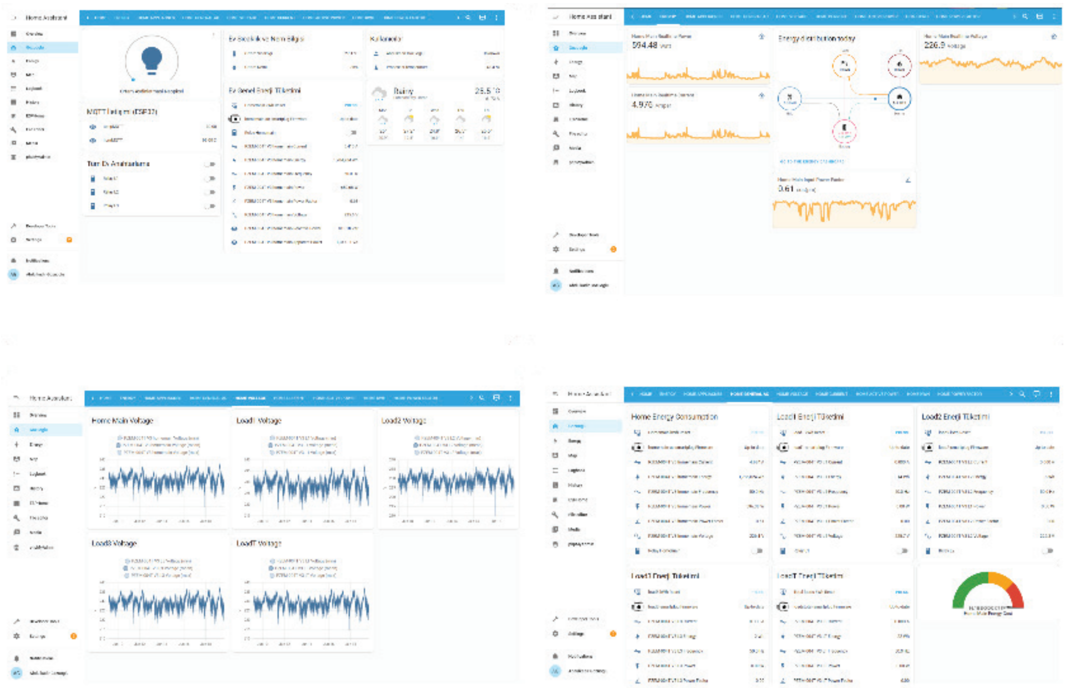


Figure 10. Some sample views show the user interface of LHAS, from left to right; the Main window, energy monitor window, voltage monitor window, and whole smart plug measurements.

The ESP-32 kit collects the input data and processes the rules and functions required for the fuzzy-logic controller, turning the verbal rules into a digital output. The LDR is connected to the ADC pin of ESP-32, so the measurement is obtained directly. The other temperature, humidity, and AC energy measurements are received via MQTT communication.

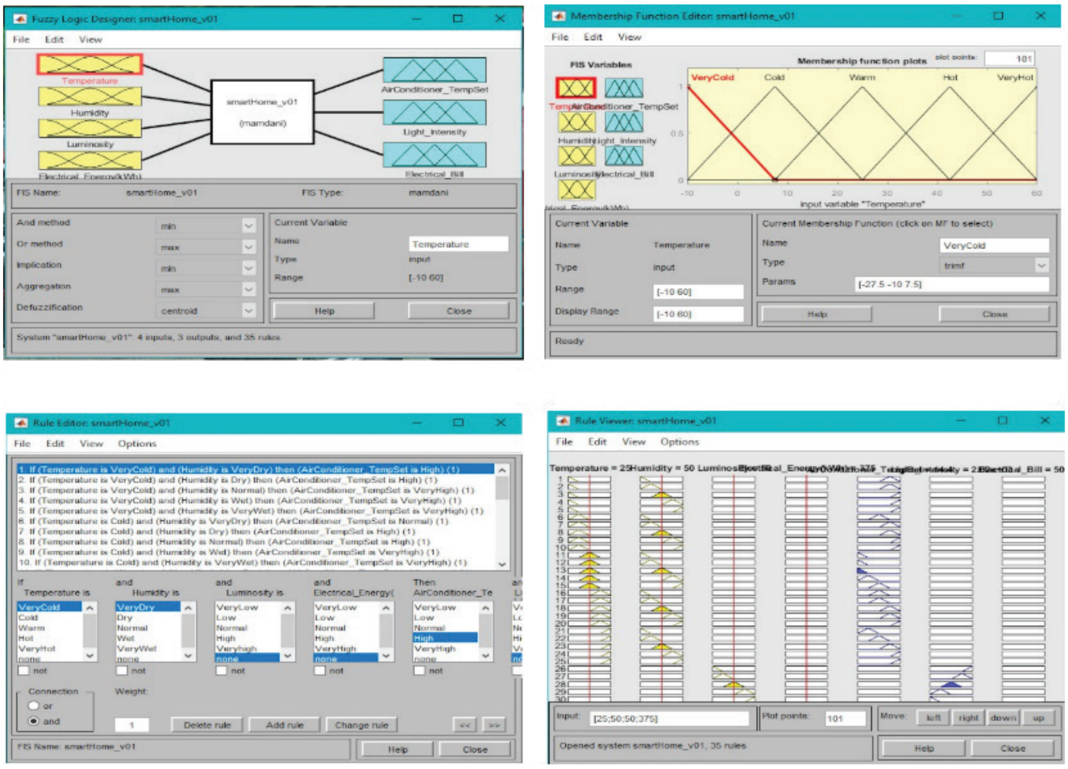


Figure 11. Sample views of fuzzy logic controller design in MATLAB fuzzy logic toolbox. From left to right; main controller window, membership function window, rules creation window, and rules graphical representation window.

Integration Between ESP Based Modules and Server

The factory firmware allows the µCs to be connected and coded in various software environments. In our study, firmware updates were required to ensure the integration of the ESP kits with the Hassio. The ESP-Home Flasher [50] application was used for the firmware updates. The firmware update process is done by uploading "Binary" files prepared in the Hassio to the µCs. The uploading can be performed via serial communication or a Wi-Fi network. The Binary files are created with the help of a library called ESP-Home application. The ESP-Home is an application that has been installed in Hassio. As a result of the firmware update, the ESP-based modules communicate with Hassio and perform automated processes.

MQTT Communication Between Smart Modules & Server

Because the intelligent modules designed in this study are ESP-based, communication with the server can be achieved using the appropriate Binary file with the help of firmware update. The process can be repeated for commercially produced modules based on ESP, and integration with the server can be realized. However, in cases where modules that do not allow firmware updates are used or the software in the µC is too complex to be written on the ESP Home application, integration with the server can be performed using MQTT communication.

In this study, the temperature, humidity, and energy measurement values, which are the input information of the fuzzy-logic controller, were obtained by the ESP-32 over the MQTT Broker. Fuzzy Logic, controller output values are also sent to the server using MQTT Broker.

Database Implementation

The MariaDB database was used on LHAS The database records all measurement and control information of the users. This data can be used to report or obtain retrospective information. The data in the database were also used in our study's CNN/LSTM deep-learning processes. The specified learning processes were performed to obtain the consumption behaviors of consumers.

Determination of Consumption Behavior and Load Profiles

Determining the energy demand and realizing energy production, transmission, and distribution planning in future smart grids are essential. Specifying user consumption behaviors and load profiles using demand-side energy management systems significantly benefits grid planning processes. This study applied CNN/LSTM deep learning to perform the energy consumption learning process. The CNN is designed as five layers and LSTM as six. The software is prepared in Python. The specific setting parameters of the CNN/LSTM model performed in Python are shown in Table 2.

Table 2. The proposed CNN/LSTM model's layer configuration

Layer	Configuration	Activation	Others
Convolution #1	Filter num.=64, Size of kernel=3, Strides=1	Relu	Loss=MSE
Convolution #2	Filter num.=128, Size of kernel =3, Strides=1	Relu	Optimizer=adam
Convolution #3	Filter num.=256, Size of kernel =3, Strides=1	Relu	Epochs=600
Convolution #4	Filter num.=128, Size of kernel =3, Strides=1	Relu	Batch size=16
Convolution #5	Filter num.=64, Size of kernel =3, Strides=1	Relu	
LSTM #1	Num. = 64	Tanh	
LSTM #2	Num. = 128	Tanh	
LSTM #3	Num. = 256	Tanh	
LSTM #4	Num. = 128	Tanh	
LSTM #5	Num. = 64	Tanh	
LSTM #6	Num. = n_feats	Tanh	
Dropout	Dropout(0.1)		

Deep learning analysis is performed according to the stored data. The data is saved into the database within one minute. Approximately 10,000 (1x60x24x7=10080) data form one week of data. After one week of data recording, the learning process started and was repeated at this interval. The current, voltage, and active power values were used as the input data in the learning process.

CNN performs learning operations efficiently on long-time series-type data. It is used to identify and learn the repetitive consumption patterns of users. The LSTM method performs the learning processes of consumption methods that do not have a certain period and suddenly appear out of the repetitive consumption behavior.

This study used *the Nvidia Jetson Nano Developer Kit* as the deep-learning hardware. Deep learning can also be performed using the Raspberry Pi. Raspberry Pi boards are insufficient for long data deep learning applications. For this reason, Jetson Nano, which has a 128-Core Maxwell GPU, is relatively low-cost and produced for artificial intelligence applications, is preferred [51]. Jetson Nano kit continuously performs deep learning processes for 10000 data (weekly) by using the consumption data saved in the database on the server.

RESULTS AND DISCUSSION

In this study, a demand-side energy management system was designed for future smart grids, and its real-time application was performed. The primary goal we plan to create in the energy management system is to determine users' consumption behavior and load profiles while applying an advanced control system on the demand side. Hardware and software applications are based on open-source, low-cost considerations.

The hardware of the proposed system is applied to a real house, as seen in Figure 12. The measurement data are obtained from a real house. The data is saved to the database. Wi-Fi and MQTT communication were tested, and positive results were obtained, as seen in Figure 13.

The fuzzy logic controller was tested. The real-time calculation results from ESP-32-based module are compared with MATLAB simulation results. The fuzzy logic output signals for temperature, humidity, luminosity, and AC energy can be seen in Figure 14. The fuzzy logic controller output signals are met with approval.

The deep learning analysis was performed in Jetson Nano Developer Kit. The Jetson Nano comes with an Ubuntu 18.04 version. Ubuntu is a free, open-source, and popular Linux-based operating system. Three separate sketches were prepared for deep learning analysis. The first sketch creates a file with a comma-separated values (CSV) extension. The current, voltage, and power data are saved to the file. The second sketch is the main one. The CNN/LSTM deep learning model is prepared for deep learning analysis. The third sketch is used for a graphical representation of the results.

The actual measurement data, prediction data obtained as a result of deep learning, and the data limit values used in the learning process are shown in Figure 15 and Figure 16 for 4000 and 10000 data.

The error analysis was performed with three types of data. These are verified data, training data, and all data.

Figure 12. Installing AC smart plug in a real house. Measure the energy and analyze the consumption behavior.

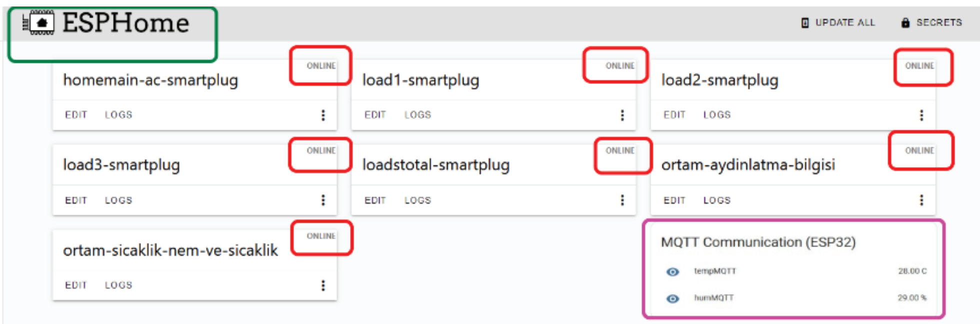


Figure 13. Hassio user interface Wi-Fi and MQTT communication test.

The verified data is 80% of the total data the deep learning analysis uses for further analysis. The training data is the last 20% of the whole data. All data is a combination of verified and training data. Table 3 shows the error rates in the error rate tests mean squared error (MSE), root mean square error (RMSE), and mean absolute error (MAE) performed based on the deep learning results—It is seen that the oscillation between the original data and the learned data is between 6% and 8%.

The overall cost of the energy management system created in our study was compared with that of the products used commercially, as shown in Table 4.

The list includes essential software and devices that should be used in a smart home. The table contains only some of the devices and equipment used in the study. Primary devices are added to give an idea of the total cost of the proposed system. It is seen that commercial products cost approximately three times more.

In this study, we proposed an advanced intelligent demand-side energy management system with automated learning functionality. The preferred equipment and software are relatively low-cost. The integration methods between LHAS and intelligent modules were tested, and positive results were obtained. The related works and recent studies on this subject have focused on typical parts of the system.

The studies implemented on IoT networks for smart homes focus on various communication methods. The studies give information about RF, ZigBee, LoRa, Wi-Fi, and Bluetooth communications [4-13]. In other applications, ESP-based intelligent modules have been designed for various purposes inside smart homes [14-20]. A fuzzy logic controller was designed and implemented for intelligent homes. The applied studies are prepared to control house heating, cooling, and energy consumption appliances [21-27]. Deep learning applications are

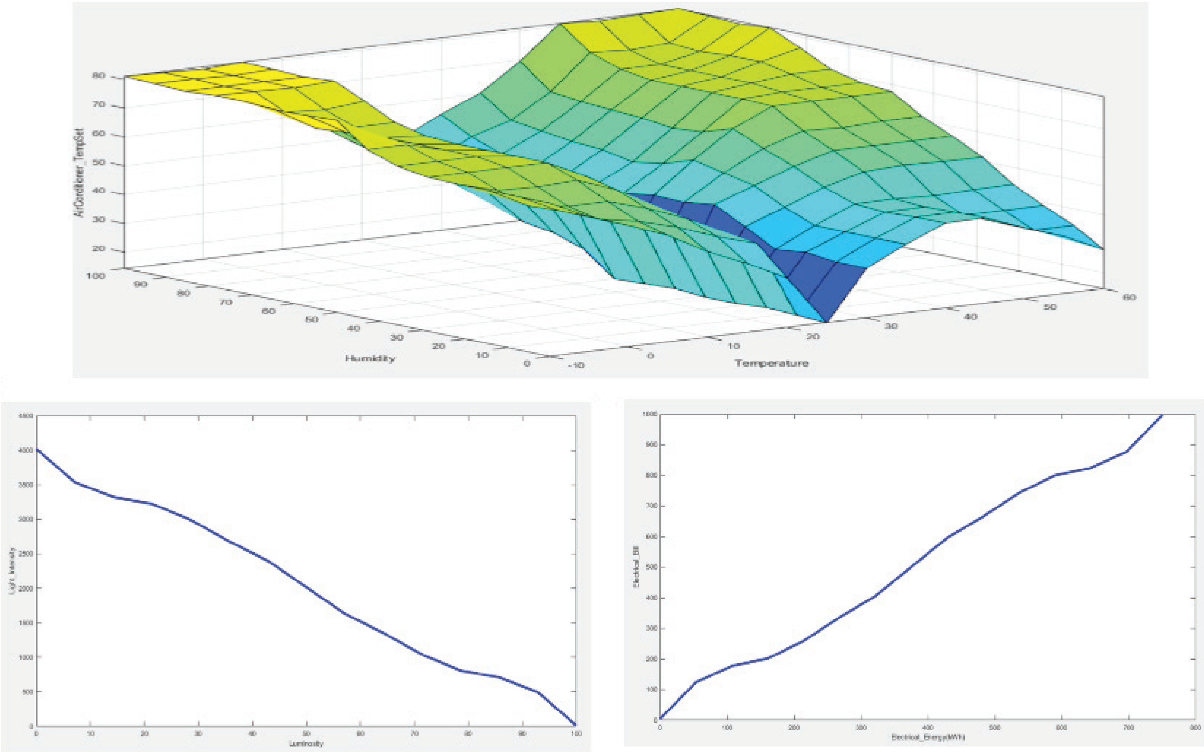


Figure 14. The output signals of the fuzzy logic controller. From left to right; temperature, humidity, luminosity, and AC energy.

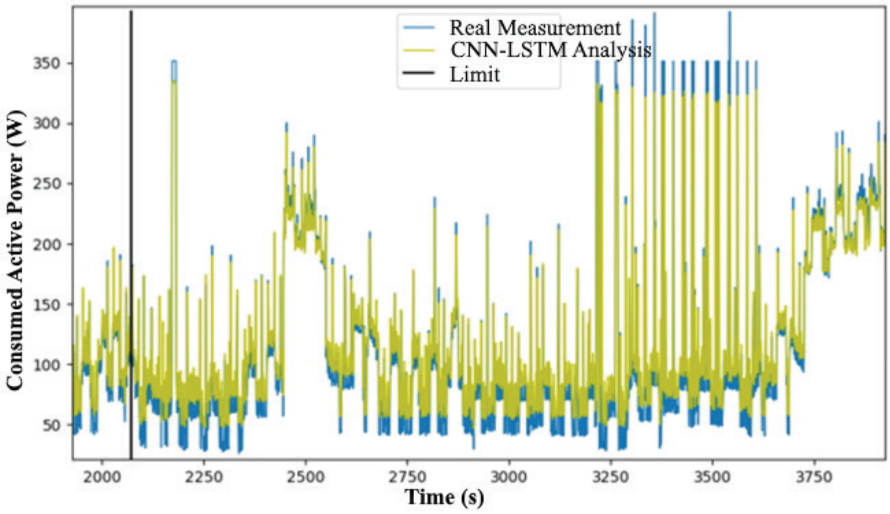


Figure 15. CNN/LSTM analysis data recordings for 4000 data.

also implemented in intelligent home studies. The proposed designs focus on energy consumption estimations [28-36].

Commercially available intelligent modules on the market communicate using their own IoT networks. Commercial devices do not allow communication and integration with standard IoT networks. Integration is necessary to use different brands of modules used in the house together. It will be beneficial for the smart homes of the future to allow commercial brands to use their own IoT networks and integrate with standard IoT networks with a method such as MQTT or Wi-Fi

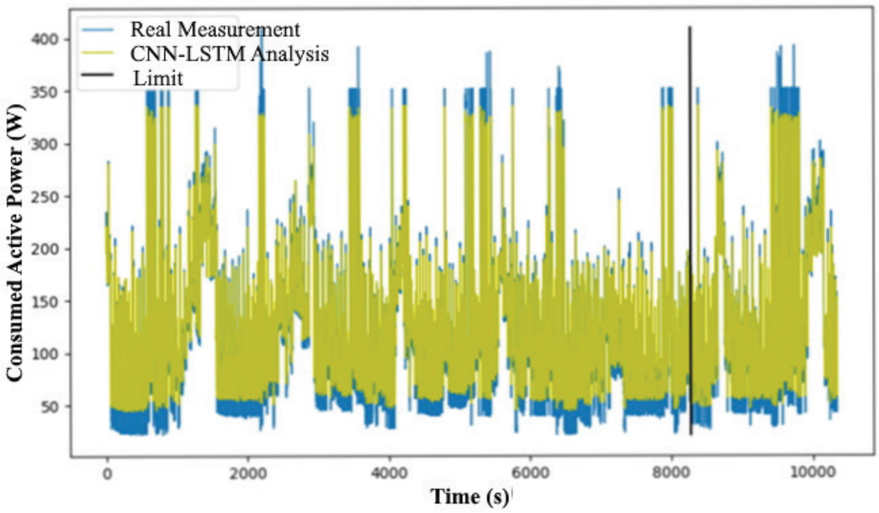


Figure 16. Comparison of measurement and actual values as a result of CNN/LSTM analysis by using full 10000 data.

Table 3. Deep learning error rate analysis

Data type & Error rate	MSE (%)	RMSE (%)	MAE (%)
Verified data	6.074	7.793	6.247
Training data	6.448	8.030	6.477
All data	6.303	7.939	6.418

MSE: Mean squared error; RMSE: Root mean square error; MAE: Mean absolute error.

Table 4. The smart modules cost analysis comparison

Commercial Smart Modules	~Cost ($)	The Modules in This Study	~Cost ($)
Smart single channel switch	$21.67	ESP-01 Based Relay Module	$2.78
Wi-Fi temperature and humidity sensor	$27.50	ESP-01 Based Temperature & Humidity Module	$4.17
Smart illumination sensor	$19.44	NodeMCU Based Light Intensity Module	$5.00
WS2811 NeoPixel driver with Wi-Fi control	$19.44	ESP-01 Based NeoPixel Led Driver Module	$3.33
Smart Energy Measurement Module	$27.78	NodeMCU & PZEM Sensor Based AC Energy Metering Module	$13.89
Gateway Module Smart Home Automation	$333.3	Raspberry Pi Based LHAS Hardware	$138.89
Main Processor for AI Analysis	$776.6	Jetson Nano Development Kit	$490,2
Total	**$1.225**	**Total**	**$168.06**

In our study, Hassio was used as LHAS, and an IoT network was created. Low-cost innovative modules have been designed and created in order to make home appliances smart. A fuzzy logic controller is designed to increase the control resolution in the house. The MQTT communication environment has been prepared to integrate non-ESP-based microcontrollers and commercial intelligent modules. A database was established in LHAS, and all measurements were recorded. Using the data saved in the database and the Jetson Nano developer kit, the user's consumption behavior was learned through the CNN/LSTM method. All these processes are arranged to be managed from a single center within Hassio. In addition, all devices and software used have been determined with low-cost considerations.

CONCLUSION

Currently, energy efficiency and energy savings are among the most important issues. More efficient use of electrical energy depends on making the right plans and knowing information about the network's demand side. Depending on the provided user type, determining the load profiles result in a correct analysis of the load flows in the high, medium, and low voltages (HV, MV, and LV) networks.

This analysis proposes more accurate planning for required energy by smart grids. Considering these stages, the demand-side management system provides intelligent solutions for individual users. In addition, it provides data needed by the networks and enables future smart grids to operate more efficiently.

Future research will be oriented toward extending the system to different consumer types and MV network loads. The current study is performed predominantly in residential loads. In contrast, applying the method to commercial, industrial LV consumers and MV loads will help smart girds better, while energy demand analysis application.

ACKNOWLEDGEMENTS

This work was financially supported by OMU BAP (project number: PYO. MUH.1904.21.018). The authors would like to thank for this support.

AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

DATA AVAILABILITY STATEMENT

The authors confirm that the data that supports the findings of this study are available within the article. Raw data that support the finding of this study are available from the corresponding author, upon reasonable request.

CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ETHICS

There are no ethical issues with the publication of this manuscript.

REFERENCES

[1] Garg H, Dave M. Securing IoT Devices and Securely Connecting the Dots Using REST API and Middleware. IEEE IoT-SIU 2019:1−6. [CrossRef]
[2] Ponnusamy V, Sharma B. Investigation on IoT Intrusion Detection in Wireless Environment. IEEE ICCOINS 2021:7−13. [CrossRef]
[3] Vishwakarma SK, Upadhyaya P, Kumari B, Mishra AK. Smart Energy Efficient Home Automation System Using IoT. IEEE IoT-SIU 2019:1−4. [CrossRef]
[4] Afifah K, Fuada S, Putra RVW, Adiono T, Fathany MY. Design of low power mobile application for smart home. IEEE ISESD 2016:127−131. [CrossRef]
[5] Gebhardt J, Massoth M, Weber S, Wiens T. Ubiquitous Smart Home Controlling Raspberry Embedded System. UBICOMM. 2014.
[6] Jain S, Vaibhav A, Goyal L. Raspberry Pi based interactive home automation system through E-mail. IEEE ICROIT 2014:277−280. [CrossRef]
[7] Ding J, Wang Y. A WiFi-based smart home fall detection system using recurrent neural network. IEEE Trans Consum Electron 2020;66:308−317. [CrossRef]
[8] Jabbar WA, Kian TK, Ramli RM, Zubir SN, Zamrizaman NSM, Balfaqih M, et al. Design and Fabrication of Smart Home With Internet of Things Enabled Automation System. IEEE Access 2019;7:144059−144074. [CrossRef]
[9] Kane L, Liu V, McKague M, Walker GR. Network architecture and authentication Scheme for LoRa 2.4 ghz smart homes. IEEE Access 2022;10:93212−93230. [CrossRef]
[10] Franco P, Martínez JM, Kim YC, Ahmed MA. IoT based approach for load monitoring and activity recognition in smart homes. IEEE Access 2021;9:45325−45339. [CrossRef]
[11] Allifah NM, Zualkernan IA. Ranking security of IoT-based smart home consumer Devices. IEEE Access 2022;10:18352−18369. [CrossRef]
[12] Illy P, Kaddoum G, Kaur K, Garg S. ML-Based IDPS enhancement with complementary features for home IoT networks. IEEE Trans Netw Serv Manag 2022;19:772−783. [CrossRef]
[13] Ulloa-Vásquez F, García-Santander L, Carrizo D, Heredia-Figueroa V. Intelligent electrical pattern recognition of appliances consumption for home energy management using high resolution measurement. IEEE Lat Am Trans 2022;20:326−334. [CrossRef]
[14] Tahir M, Mohammed AS, Marwaha S, Vij M, Saeed Y, Atif M, et al. Wi-Fi Aided home energy management system and AC prediction through temperature and humidity sensors. IEEE ICCR 2022:1−9. [CrossRef]
[15] Přeučil T, Novotný M. Evaluation of power saving methods for low-power WiFi environment sensors. IEEE MECO 2022:1−5. [CrossRef]
[16] Akkurt GG. Performance indices of soft computing models to predict the heat load of buildings in terms of architectural indicators. J Therm Eng 2017;3:1358−1374. [CrossRef]
[17] Beki C, Sahari N, Ismail N. Internet of things based smart home system Using ESP32 microcontroller. Prog Eng Appl Technol 2020;1:267−273.
[18] Madhu GM, Vyjayanthi C. Implementation of cost effective smart home controller with android application using Node MCU and internet of things (IOT). IEEE ICEPE 2018:1−5. [CrossRef]

[19] Singh K, Jain A, Thiyagarajan P. Design and implementation of integrated control system for IoT enabled home automation. IEEE ICAC3N 2022:1433−1437. [CrossRef]
[20] Midul MAHS, Pranta SH, Biddut ASI, Siam SI, Hazari MR, Mannan MA. Design and implementation of IoT-based smart energy meter to augment residential energy consumption. IEEE ICREST 2023:106−110. [CrossRef]
[21] Ain QU, Iqbal S, Mukhtar H. Improving quality of experience using fuzzy controller for Smart homes. IEEE Access 2022;10:11892−118908. [CrossRef]
[22] Krishna PN, Gupta SR, Shankaranarayanan PV, Sidharth S, Sirphi M. Fuzzy logic based smart home energy management system. IEEE ICCCNT 2018:1−5. [CrossRef]
[23] Sevil M, Elalmıs N, Görgün H, Aydin N. Control of air conditioning with fuzzy logic controller design for smart home systems. Sigma J Eng Nat Sci 2015;33:439−463.
[24] Zhang L, Leung H, Chan KCC. Information fusion based smart home control system and its application. IEEE Trans Consum Electron 2008;54:1157−11565. [CrossRef]
[25] Jabeur R, Ouaaline N, Lakrim A. A fuzzy logic controller controls a smart lighting system for energy savings. IEEE IRSEC 2021:1−6. [CrossRef]
[26] Paramathma MK, Devaraj D, Selvi VAI, Karuppasamypandian M. Development of fuzzy logic based approach for consumer side management in smart home. IEEE ICACITE 2021:1056−1061. [CrossRef]
[27] Roy C, Binodon, Karmaker A, Ghosh HR. A naive bayes classifier based energy scheduling for home energy management system (HEMS). IEEE ICECE 2022:212−215. [CrossRef]
[28] Lu R, Jiang Z, Wu H, Ding Y, Wang D, Zhang HT. Reward Shaping-Based Actor-Critic Deep Reinforcement Learning for Residential Energy Management. IEEE Trans Ind Inform 2023;19:2662−2673. [CrossRef]
[29] Alhussein M, Aurangzeb K, Haider SI. Hybrid CNN-LSTM model for short-term individual household load Forecasting. IEEE Access 2020;8:180544−180557. [CrossRef]
[30] Zhou Y, Ren B, Xue X, Chen L. Building energy consumption forecasting based on K-Shape clustering and CNN-LSTM. IEEE ICPET 2022:1147−1152. [CrossRef]
[31] Yan K, Li W, Ji Z, Qi M, Du Y. A Hybrid LSTM neural network for energy consumption forecasting of individual households. IEEE Access 2019;7:157633−157642. [CrossRef]
[32] Yu L, Qin S, Zhang M, Shen C, Jiang T, Guan X. A review of deep reinforcement learning for smart building energy management. IEEE Internet Things J 2021;8:12046−12063. [CrossRef]
[33] Han T, Muhammad K, Hussain T, Lloret J, Baik SW. An efficient deep learning framework for intelligent energy management in IoT networks. IEEE Internet Things J 2021;8:3170−3179. [CrossRef]
[34] Kodama N, Harada T, Miyazaki K. Home energy management algorithm based on deep reinforcement learning using multistep prediction. IEEE Access 2021;9:153108−153115. [CrossRef]
[35] Gao Y, Li S, Xiao Y, Dong W, Fairbank M, Lu B. An iterative optimization and learning-based iot system for energy management of connected buildings. IEEE Internet Things J 2022;9:21246−21259. [CrossRef]
[36] Lu R, Bai R, Luo Z, Jiang J, Sun M, Zhang HT. Deep reinforcement learning-based demand response for smart facilities energy management. IEEE Trans Ind Electron 2022;69:8554−8565. [CrossRef]
[37] Home-Assistant [cited 2023 May]. Hassio, Open Source Home Automation Server. Available from: https://www.home-assistant.io/
[38] Pishva D, Takeda K. Product-based security model for smart home appliances. IEEE Aerosp Electron Syst Mag 2006;23:32−41. [CrossRef]
[39] Kumar D, Shen K, Case B, Garg D, Alperovich G, Kuznetsov D, et al. All things considered: An analysis of iot devices on home networks. USENIX Secur Symp 2019:1169−1185.
[40] Jantzen J. Foundations of fuzzy control. A practical approach. 2nd ed. Chichester, UK: Wiley; 2013. [CrossRef]
[41] Huang H, LV Y. Short-term Tie-line Power Prediction Based on CNN-LSTM. IEEE EI2 2020:4118−4122. [CrossRef]
[42] Gaur K, Singh SK. CNN-Bi-LSTM Based Household Energy Consumption Prediction. IEEE ICPSC 2021:233−237. [CrossRef]
[43] Ren L, Dong J, Wang X, Meng Z, Zhao L, Deen MJ. A data-driven auto-CNN-LSTM prediction model for lithium-ion battery remaining useful life. IEEE Trans Ind Inform 2020;17:3478−3487. [CrossRef]
[44] Wang Y, Du X, Lu Z, Duan Q, Wu J. Improved LSTM-based time-series anomaly detection in rail transit operation environments. IEEE Trans Ind Inform 2022;18:9027−9036. [CrossRef]
[45] MQTT-Broker [Internet]. [cited 2023 May]. The Standard for IoT Messaging. Available from: https://mqtt.org/
[46] Espressif-ESP01 [Internet]. [cited 2023 May]. ESP8266 ESP-01 Wifi Module. Available from: https://www.espressif.com/en/support/documents/technical-documents?keys=&field_type_tid%5B%5D=14&field_download_document_type_tid%5B%5D=510
[47] Espressif-NodeMCU [Internet]. [cited 2023 May]. ESP8266 ESP-12E NodeMCU Wifi Module. Available from: https://www.espressif.com/en/

support/documents/technical-documents?keys=&-field_type_tid%5B%5D=14&field_download_document_type_tid%5B%5D=510

[48] Espressif-ESP32 [Internet]. [cited 2023 May]. ESP32-Wroom-32D devkit. Available from: https://www.espressif.com/en/support/documents/technical-documents?keys=&field_type_tid%5B%5D=266

[49] PZEM-004T [Internet]. [cited 2023 Jun 6]. PZEM-004T V3.0 User Manual. Available from: https://innovatorsguru.com/wp-content/uploads/2019/06/PZEM-004T-V3.0-Datasheet-User-Manual.pdf

[50] ESPHomeFlasher [Internet]. [cited 2023 May]. ESP Home Flasher. Available from: https://github.com/esphome/esphome-flasher

[51] Jetson-Nano [Internet]. [cited 2023 May]. Nvidia Jetson Nano Developer Kit. Available from: https://developer.nvidia.com/embedded/jetson-nano-developer-kit