

Predictive Modeling of Energy Consumption in U.S. Smart Homes Using IoT Sensor Data and Climate Variables

Jairo Gonzales-Coronel¹, Rommel Auqui-Gamboa¹, Renzo Lopez-Jimenez¹, Javier Gamboa-Cruzado², Angel Nuñez Meza³, Alex Salazar-Marzal², Dulio Oseña Gago⁴, Jorge Nolasco-Valenzuela⁵, Liliana Chanona-Hernández^{6,*}

¹ Universidad Nacional Federico Villarreal,
Peru

² Universidad Nacional Mayor de San Marcos,
Peru

³ Universidad Nacional Daniel Alcides Carrión,
Peru

⁴ Universidad Nacional de Cañete,
Peru

⁵ Universidad San Ignacio de Loyola,
Peru

⁶ Instituto Politécnico Nacional, ESIMEZ,
Mexico

{2021018211, 2022016425, 2022022424}@unfv.edu.pe,
{jgamboac, alexmelecio.salazar}@unmsm.edu.pe, anunezm@undac.edu.pe,
dosed@undc.edu.pe, jorge.nolasco@usil.pe, lchanonah@ipn.mx

Abstract. This research addresses the challenge of monitoring and optimizing energy consumption in smart homes, due to the high volume of data generated by IoT devices and its relationship with climate variables. The central problem lies in the absence of an IoT system capable of processing, integrating, and transforming large volumes of data into useful information for energy decision-making. The objective was to design, implement, and evaluate an intelligent system based on IoT, AWS, DuckDB, Python, and Power BI for monitoring, analyzing, and visualizing energy consumption in smart homes. For this purpose, the agile Scrum methodology was used, due to its iterative and incremental approach. A dataset of 503,910 records was used, from which a 5% sample equivalent to 25,195 observations was selected. The developed architecture integrated IoT sensor data, AWS storage, data cleaning and loading through Python, queries with DuckDB, and interactive visualization in Power BI. The results showed improvements in query time, report generation, reduction of incorrect data, and elimination of missing data in the experimental group after the implementation

of the system. In conclusion, the proposed architecture proved to be efficient and reliable for supporting timely decisions aimed at energy efficiency in smart homes.

Keywords. Database, smart home, energy consumption, IoT, energy efficiency; power BI, internet of things.

1 Introduction

The expansion of smart homes and IoT devices generates a massive volume of energy consumption data and climate variables. The challenge lies in the inefficiency of traditional methods for processing and analyzing this information, which hinders effective optimization and decision-making for energy efficiency. This requires the creation of a robust system that ensures data quality and response speed. Various studies have shown that IoT

architectures, both in edge and cloud environments, make it possible to collect, store, process, and query energy data efficiently. Along these lines, HEMS Cloud-IoT solutions, IoT API optimization, and query models for IoT databases have been implemented, showing improvements in storage, control, latency, and operational performance [1, 10, 13].

Likewise, the authors [2, 6, 29] have addressed energy optimization in smart homes through hybrid approaches, edge-cloud frameworks, and LSTM-based models aimed at reducing costs, decreasing dependence on the grid, and maintaining user comfort. These studies confirm that the combination of IoT, artificial intelligence, and contextual variables makes it possible to generate actionable recommendations for residential energy management.

Complementarily, recommendation systems applied to prosumers, solar energy, and electrical self-sufficiency have shown important advances in consumption optimization and energy trading. In particular, LLMs, machine learning models, LightGBM, SHAP, and heat maps have been used to identify key decision factors and improve the accuracy of energy recommendations [3, 5, 58].

On the other hand, the authors [4, 45] have highlighted that IoT solutions also contribute to environmental sustainability by reducing unnecessary data transmission, decreasing energy consumption, and supporting more efficient urban systems. These contributions link energy efficiency with emission reduction and sustainability in smart environments.

Regarding data, several studies have developed or used energy and IoT datasets to analyze consumption patterns, evaluate algorithms, and study residential behaviors. Studies based on smart meters, high-frequency electricity datasets, and ZigBee traffic show the importance of having structured, reliable, and representative data to validate models in smart homes [7, 9, 37].

In addition, user interaction with intelligent systems has been analyzed from the perspectives of adoption, perception, and visualization of energy information. The authors [8, 12, 35] show that user acceptance, visual

clarity, and the design of dashboards or mobile applications are critical elements for transforming complex data into understandable and useful information for decision-making.

Regarding predictive modeling, there is a growing trend toward advanced machine learning and deep learning models for forecasting energy consumption. The reviewed studies have used federated learning, XGBoost, LSTM, ConvLSTM, and CNN-LSTM, achieving relevant improvements in metrics such as MAE, MAPE, RMSE, R^2 , and predictive accuracy [11, 41, 56].

Similarly, data quality constitutes a fundamental axis in IoT systems. The authors [14, 31, 33] have proposed imputation and compensation methods for missing values using XGBoost, LSTM, hybrid models, weighted averages, and stacking, demonstrating that the proper treatment of incomplete data improves the reliability of subsequent analyses.

Continuing along this line, other studies have examined anomalies, duplicates, missing data, outliers, and sensor failures in smart home and physical sensor contexts. These works show that the detection and correction of data quality problems are essential to ensure reliable energy analyses and avoid bias in the interpretation of results [15, 16, 23].

In the field of security, the authors [17, 27, 32] have identified relevant threats in IoT and smart home ecosystems, including privacy intrusion, unauthorized access, cyberattacks, and network anomalies. Likewise, they have proposed IDS frameworks, ensemble learning, rule induction, and hyperparameter optimization to strengthen threat detection and improve the reliability of intelligent systems.

Complementarily, Padilla-Gomez and colleagues [68] show that the expansion of IoT devices increases the risks associated with data confidentiality, integrity, availability, authentication, and authorization. This contribution is relevant for energy monitoring systems in smart homes, because data transmission from sensors and cloud platforms requires protection mechanisms that ensure system reliability.

On the other hand, efficiency in the provisioning, composition, and computational offloading of IoT services has been addressed

through ZTP solutions, multi-objective optimization, and offloading strategies.

These studies show that service automation, response time reduction, and computational offloading are essential factors for improving the scalability and performance of smart homes [18, 30, 66].

Regarding smart home architectures, the authors [19, 21, 65] have described the evolution of SHS toward multifunctional, automated, and IoT-based models. Their contributions integrate sensors, microcontrollers, wireless protocols, control services, and big data architectures to improve the management of domestic scenarios and the automatic response of the system.

Likewise, the integration of IoT, artificial intelligence, and data source search has been addressed through AIoT approaches, bibliometric studies, and frameworks for tracking and indexing IoT resources. The authors [20, 26, 67] show that this convergence makes it possible to expand analytical capacity, automation, data discovery, and decision-making in intelligent systems. Along similar lines, Andrade-Mogollon and collaborators [69] show that the integration of IoT, artificial intelligence, and data analytics makes it possible to monitor critical variables in real time, optimize resources, and strengthen decision-making in intelligent environments. Although their application focuses on precision agriculture, their findings are transferable to the smart home context, where the continuous capture of energy and climate data makes it possible to improve operational efficiency and sustainability.

Regarding human activity recognition in smart homes, the authors [22, 24, 28] have reviewed and applied algorithms based on IoT sensors, 1D-CNN, LSTM, RNN, and GRU to classify daily activities. These studies demonstrate that activity analysis makes it possible to better understand residential behavior and enrich predictive models associated with energy consumption.

Relatedly, more advanced approaches have been proposed for activity recognition using Transformers, semi-supervised learning, semantic reasoning, SimCLR, and attention mechanisms. The authors [34, 36, 54] show that these techniques improve accuracy in scenarios with complex sensory sequences, limited labels, or partially annotated data.

In addition, the availability of multimodal datasets has strengthened algorithm evaluation in domestic environments. The authors [38, 39, 59] have developed databases with audio, vibration, environmental variables, sensors, accelerometers, and RGB-D data, providing relevant resources to validate automated activity recognition systems in smart homes.

From the perspective of data evaluation and reliability, some studies have analyzed errors, measurement quality, and statistical relationships in different applied contexts. The authors [40, 42, 48] show that rigorous error measurement, label evaluation, and statistical analysis of data quality make it possible to strengthen the validity of empirical results. Similarly, the authors [43, 46, 49] have proposed models aimed at improving computational efficiency and accuracy in activity recognition, fall detection, and kitchen activity tasks. These works show that reducing computational cost and fusing sensors make it possible to obtain more efficient systems without compromising accuracy.

In this same direction, graph neural networks, explainable models, and bidirectional recurrent architectures have been applied to activity recognition and occupancy detection. The authors [50, 51, 52] show that attention, hierarchical pooling, and visual explanations improve both model performance and user trust in intelligent environments. Likewise, the authors [53, 55] have explored multi-label classification and multimodal recognition approaches using deep networks and self-attention mechanisms. These studies are relevant because they make it possible to address scenarios involving multiple residents, multiple activities, and the fusion of heterogeneous signals within the home.

In the specific field of energy prediction, the authors [57, 60, 61] have analyzed the relationship between IoT sensors, variable correlation, previous consumption, machine learning models, and forecasting methods for smart grids. Their results suggest that non-parametric, hybrid, and sensor-based models are suitable for capturing nonlinear patterns and seasonal variations in energy consumption.

Finally, recent literature on missing data confirms that imputation is preferable to deletion when it is necessary to preserve useful

information in sensory datasets. The authors [62, 63, 64] have reviewed deterministic, probabilistic, machine learning, deep learning, GAN-based, and spatial recovery methods, highlighting that method selection should depend on the pattern, mechanism, and rate of missing data.

Finally, from a methodological point of view, the authors [44, 48] show that experimental designs and indicator analysis make it possible to rigorously evaluate the impact of a technological solution. This evidence supports the need to compare groups, measure times, data quality, and operational performance to validate the real effect of an IoT architecture applied to energy monitoring.

Although the literature reports advances in IoT architectures, predictive models, and energy management in smart homes, gaps remain in the complete integration of the data flow: cleaning, normalization, cloud storage, efficient querying, and interactive visualization. In addition, several studies prioritize advanced models but do not always propose accessible, reproducible, and low-cost frameworks based on tools such as Python, DuckDB, AWS, and Power BI. A limited articulation among climate variables, appliance-level consumption, and understandable dashboards for the end user is also observed.

Likewise, many works do not detail systematic data cleaning and validation processes, even though IoT datasets may contain errors, duplicates, missing values, or inconsistent records. This research responds to these gaps through an integrated architecture for monitoring and analyzing energy consumption in smart homes. The solution combines Python for data cleaning, DuckDB for efficient queries, AWS for cloud storage, and Power BI for interactive visualization. The study provides an accessible and replicable framework that improves operational efficiency, reduces errors and missing data, and transforms energy and climate data into useful information for decision-making. In addition, it makes it possible to identify the devices with the highest consumption and support more efficient and sustainable residential management.

The main purpose of this paper is to design, implement, and validate a monitoring and management system for energy consumption in

smart homes through an integrated framework based on IoT, AWS, DuckDB, Python, and Power BI, aimed at optimizing the processing, querying, and visualization of energy and climate data. The paper is organized as follows: Section 3 describes the research method, the Scrum methodology, and the evaluated indicators; Section 4 presents the case study and sprint-based development; Section 5 presents the results and discussion on operational efficiency, data quality, and analytical visualization; finally, Section 6 presents the conclusions, limitations, and future lines of research.

2 Theoretical Background

2.1 Internet of Things

Al-Sammarraie and Jawad [20] argue that the Internet of Things (IoT) interconnects devices, physical objects, electronic equipment, animals, and people through unique identifiers, enabling the autonomous exchange of data without direct human intervention. Bouchabou and other authors [22] indicate that smart homes have been driven by the advancement of IoT and the reduction in sensor costs, facilitating health services, daily assistance, energy management, and security. Majib and coauthors [25] point out that the Internet of Things (IoT) makes it possible to connect smart devices to the physical environment and the Internet, facilitating monitoring, automation, and remote control by users or other systems.

Nirupam, together with other researchers [29], states that IoT technologies integrate artificial intelligence, big data analytics, sensors, and smart meters to collect environmental data, appliance consumption, and user habits, making them essential for energy management in smart homes.

Fengyang and collaborators [30] point out that service composition in the Internet of Things has gained relevance for developing intelligent systems in areas such as industrial automation, traffic management, and smart homes. Yan [26] states that AIoT integrates the Internet of Things (IoT) and Artificial Intelligence (AI), becoming a

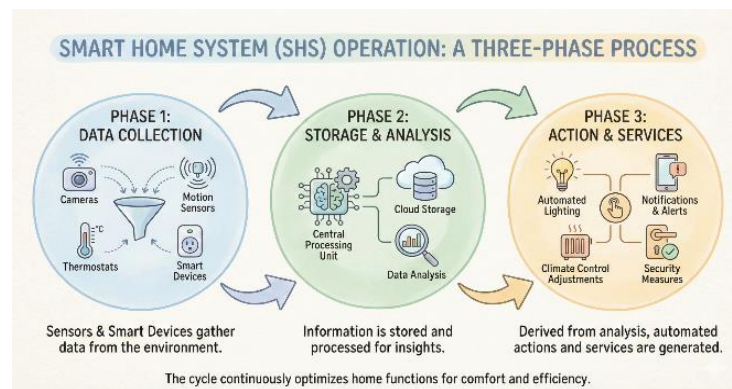


Fig. 1. The 3 Phases for the operation of an SHS

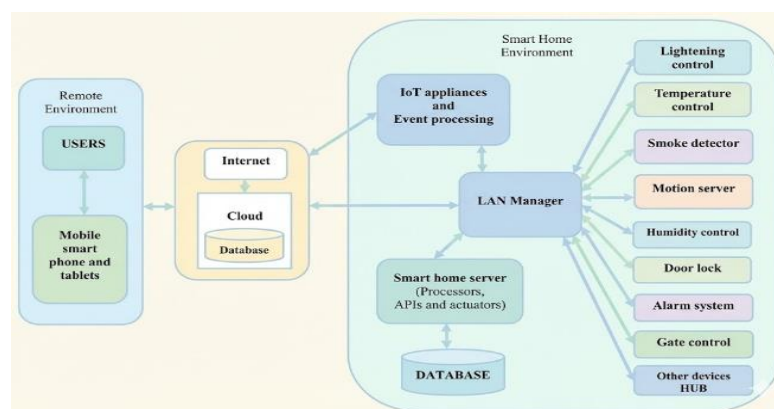


Fig. 2. Typical architecture of smart home modeling

key technology due to its potential impact on social and economic development.

2.2 Smart Homes

The definition of a Smart Home System (SHS) has been addressed by Chakraborty and collaborators [19], who states that these systems enable continuous communication between household devices and the user, facilitating the monitoring and remote control of various domestic functions from any location.

An SHS integrates sensors, cameras, microphones, and other connected devices to collect information from the environment, such as motion, temperature, humidity, smoke or gas presence, as well as the status of smart appliances. Thanks to the connectivity provided by IoT, these devices can exchange information

in real time and respond in a coordinated manner, contributing to improved comfort, security, and home efficiency (see Figure 1). Singh and colleagues [24] argue that the concept of a smart home is based on the integration of various services through a common communication system, which makes it possible to provide greater independence and comfort to residents. This is achieved through interconnected networked devices capable of forming an interactive and functional domestic environment.

2.2.1 Typical Architecture of Smart Homes

Smart homes are not limited to automation or remote device control; rather, they are characterized by the integration of advanced technologies aimed at optimizing energy efficiency, security, and user comfort. According to Djumanazarov and other authors [21], these

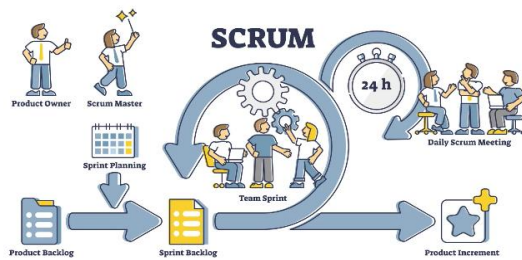


Fig. 3. SCRUM methodology development process

systems articulate sensors, actuators, wireless networks, and artificial intelligence to identify behavioral patterns, anticipate needs, and configure adaptive and functional domestic environments (see Figure 2). Likewise, interoperability among devices, through standards such as ZigBee, Z-Wave, or Wi-Fi, favors the formation of an integrated ecosystem that improves home management. Overall, smart homes reflect a transition toward more sustainable, connected, and user-centered housing.

2.2.2 Challenges and Limitations

According to Buil-Gil and their research group [27], the most recurrent threats in smart homes are mainly related to privacy breaches, unauthorized access, and improper exposure of sensitive data. Likewise, the authors identify other relevant risks, such as hacking, malware, and DoS/DDoS attacks targeting IoT devices, which can compromise the security of surveillance systems, smart appliances, and home networks.

2.3 Predictive Modeling with IoT Sensors for Energy Consumption

Energy consumption in smart homes is relevant due to its impact on household economics, environmental sustainability, and the efficient use of energy resources. In this context, predictive modeling based on IoT sensor data and climate variables makes it possible to anticipate demand, identify consumption patterns, and support decisions aimed at savings, efficiency, and the reduction of environmental impact. IoT sensors transform residential energy management by shifting from a reactive approach to a predictive and adaptive one. These devices collect data on

temperature, humidity, lighting, occupancy, appliances, and air conditioning.

When integrated with external climate variables, they make it possible to better represent the factors that influence energy consumption. This articulation is supported by the capture, transmission, storage, and preprocessing of data from IoT sensors, together with predictive algorithms such as regression, random forest, gradient boosting, neural networks, and time series. These models estimate future consumption based on historical patterns, environmental conditions, and residential habits. Overall, predictive modeling with IoT sensor data and climate variables promotes advanced energy management aimed at monitoring, anticipating, and optimizing consumption through automated decisions according to the residential context.

3 Research Method

The research adopts an applied approach, aimed at the development and evaluation of an intelligent system based on IoT sensors for the predictive modeling of energy consumption in smart homes. For this purpose, Scrum is integrated with an experimental design in order to analyze the predictive accuracy and usefulness of the system in residential energy management. The methodological process includes the operationalization of variables related to energy consumption, IoT sensor data, and climate variables, as well as the application of predictive models to estimate household energy demand. For the analysis, a dataset corresponding to smart homes located in the United States is used, which delimits the empirical context of the study.

3.1 SCRUM Solution Development Methodology

Scrum is an agile project management and development methodology based on iterative and incremental cycles. Its application makes it possible to organize work in small and multidisciplinary teams, prioritizing collaboration, continuous feedback, and the progressive delivery of functionalities.

Table 1. Operationalization of the dependent variable

Indicator	Index	Unit of measurement	Unit of observation
Report preparation time	[0–2000]	Seconds (s)	Dataset / System log
% Incorrect data	[0–30]	Percentage (%)	Dataset / Validation script
% Missing data	[0–15]	Percentage (%)	Dataset / Counting script
User satisfaction	Completely satisfied / Very satisfied / Neutral / Slightly satisfied / Not satisfied	Ordinal scale	End user / Survey
Query response time	[0–500]	Milliseconds (ms)	SQL engine (DuckDB) / Excel (simulated)

Gamboa-Cruzado and colleagues [44] describe Scrum as an agile methodology centered on collaborative work and continuous improvement, with particular relevance for software development projects. In this research, Scrum is used to guide the development of the proposed solution, allowing each system increment to be progressively validated and aligned with user requirements. This methodological approach is consistent with Mina-Rios and collaborators [70], who developed an IoT-based intelligent system through an applied methodology, experimental design, and Scrum, integrating data processing, cloud storage, and analytical visualization. In addition, it facilitates adaptation to changes, improves control of the development process, and contributes to obtaining a functional, verifiable, and results-oriented product (see Fig. 3):

- **Planning:** The objectives, scope, tasks, responsibilities, and estimated project timelines are defined. This phase establishes the organizational basis of the development

process and makes it possible to identify strategies to mitigate potential risks.

- **Implementation:** This corresponds to the execution phase, in which the requirements are transformed into functional results through previously selected tools and techniques. In technological projects, it includes data processing, component development, and continuous monitoring of progress.
- **Review and retrospective:** The quality of the implemented work and the achievement of the objectives are evaluated, identifying errors, improvements, and lessons learned to optimize the development process.
- **Release:** This corresponds to the final delivery of the developed results, products, or systems, ensuring that they are understandable, accessible, and consistent with the initial requirements.
- **Justification for the use of Scrum:** Its selection is justified because the project requires progressive validations, continuous testing, and dynamic adjustments in the management of data, cloud infrastructure, and dashboards. In addition, it makes it possible to review progress periodically and adapt the plan according to the needs of the system.

3.2 Applied Research Methodology

To evaluate the impact of the intelligent system based on IoT sensors on the prediction of energy consumption in smart homes, an applied research methodology was adopted. This approach makes it possible to empirically analyze system performance through indicators associated with predictive accuracy, data processing efficiency, and the quality of the information generated.

3.2.1 Operationalization of Variables

The operationalization of variables defines the way in which the indicators associated with the dependent variable of the study will be measured. This procedure makes it possible to quantify, compare, and experimentally analyze the

Table 2. Sprint planning

Sprint	Sprint Description	Final Deliverable
Sprint 1: Project definition and dataset search	The research area is defined, the dataset is searched for, and an architecture is proposed.	Project scope document. Selected dataset and preliminary system architecture.
Sprint 2: Data cleaning and normalization	Dataset preprocessing is performed: cleaning, normalization, and removal of duplicate data.	Cleaned and normalized dataset. E-R model. Python script for the ETL process.
Sprint 3: Connection and data upload to AWS	Configuration of AWS S3, IAM, credentials, and upload of the processed dataset.	Configured S3 bucket. Stored dataset. AWS connection script.
Sprint 4: Power BI connection to AWS	Integration between Power BI and AWS S3 or AWS Athena. Data extraction from the cloud, creation of preliminary visualizations, and exploratory analysis.	Power BI and AWS connection.
Sprint 5: Dashboard creation	The team developed complete dashboards with filters, KPIs, correlations, and report export features.	Final dashboard in Power BI and complete system documentation.

performance of the proposed system (see Table 1). As can be observed, the dependent variable is operationalized through five indicators that make it possible to evaluate system performance from technical and usage-related dimensions. Report preparation time and query response time measure the system’s operational efficiency, considering the speed with which reports are generated and queries are processed. In turn, the percentage of incorrect data and the percentage of missing data make it possible to evaluate the quality of the information processed from the dataset and validation scripts. Finally, user satisfaction, measured using a Likert scale, incorporates a perceptual dimension regarding the usefulness and acceptance of the system by the end user. Taken together, these indicators allow for a comprehensive analysis of the efficiency, reliability, and usability of the proposed solution.

3.2.1 Research Design

The research is applied in nature and uses an experimental design aimed at evaluating, under controlled conditions, the impact of a predictive model on energy consumption in smart homes:

$$RGe \quad X \quad O1 \quad , \quad (1)$$

$$RGc \quad - \quad - \quad O2 \quad . \quad (2)$$

The experimental group (Ge) received the intervention of the predictive model based on IoT data and climate variables, whereas the control group (Gc) did not receive treatment. The random assignment of the groups strengthens the internal validity of the experiment. Subsequently, comparative measurements were carried out to determine the effect of the intervention on the defined indicators.

3.2.2 Population and Sample

The population consisted of all the records available in the dataset used, corresponding to smart homes in the United States, with a total of 503,910 records.

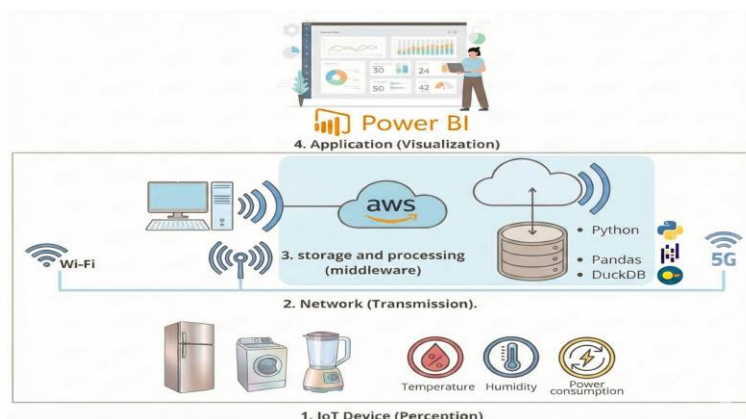
The sample was defined by considering 5% of the total number of records, equivalent to 25,195 observations. This selection made it possible to work with a representative volume of data for the experimental analysis, while maintaining adequate processing and evaluation conditions for the predictive model.

3.2.3 Data Collection Procedure

In this research, data collection was carried out through the selection and analysis of a public dataset, specifically “Augmented Smart Home

Table 3. Functional requirements

No.	Requirement	Description	Priority
FR1	Data reading	The system must allow the dataset data to be loaded into Python.	High
FR2	Data processing	The system will transform the data and enable queries on the stored data using DuckDB.	High
FR3	Data transfer to the cloud	The system must store the processed data in the cloud through Amazon Web Services.	High
FR4	Connection with Power BI	The dashboard must connect to AWS to extract the processed data.	High
FR5	Energy consumption visualization	The dashboard must display energy consumption by appliance and climate conditions.	High
FR6	Analysis filters	The system must allow data to be filtered by time periods and device type.	Medium
FR7	Dashboard generation	The dashboard must calculate indicators such as total, average, and maximum consumption.	Medium

**Fig. 4.** IoT system architecture

with Weather Information,” available on Kaggle and whose original source is found in Mendeley Data. The techniques used included indirect observation of records and database querying to obtain information on electricity consumption, household conditions, and climate variables.

4 Case Study

For the development of the solution based on a predictive model of energy consumption and climate variables, the Scrum methodology was applied. This approach made it possible to organize the implementation into iterative and incremental phases, facilitating the progressive

validation of system advances (see Table 2). Likewise, Scrum has proven to be an effective strategy for optimizing development and continuous improvement processes in technological and educational contexts, as reported by Gamboa-Cruzado and colleagues [44].

4.1 Sprint Planning

In this phase, the system architecture, functional requirements, and objectives of each sprint were defined. This process made it possible to organize development activities, prioritize tasks, and establish progressive deliverables for the implementation of the proposed solution.

Table 4. Non-functional requirements

No.	Requirement	Description	Priority
NFR1	Scalability	The system must allow the data volume to increase without significantly affecting performance.	High
NFR2	Response time	Queries must be executed within no more than 3 seconds using the dataset data.	High
NFR3	High availability	Cloud storage must guarantee continuous access to the data.	High
NFR4	Security	The cloud service must protect access through authentication and permission mechanisms.	High
NFR5	Intuitive interface	The dashboard must be easy for the end user to use, interpret, and understand.	Medium

```

import pandas as pd
import re
# 1. Load dataset
file_path = r"D:\HomeC_augmented.csv"
df = pd.read_csv(file_path)
# 2. Remove unnecessary index column
if "Unnamed: 0" in df.columns: df = df.drop(columns=["Unnamed: 0"])
# 3. Standardize column names (snake_case)
def clean_name(col):
    col = col.lower().strip()
    col = re.sub(r"[^a-z0-9]", "kw", col)
    col = re.sub(r"[^a-z0-9]", "_", col)
    col = re.sub(r"[^a-z0-9]", "_", col).strip('_')
    return col
df.columns = [clean_name(c) for c in df.columns]
# 4. Convert the time column and remove invalid timestamps
df["time"] = pd.to_datetime(df["time"], errors="coerce")
df = df.dropna(subset=["time"])
df = df.sort_values("time")
# 5. Remove duplicated records and duplicated timestamps
df = df.drop_duplicates()
df = df.drop_duplicates(subset=["time"])
# 6. Convert variables to numeric format when applicable
for col in df.columns:
    if col != "time": df[col] = pd.to_numeric(df[col], errors="coerce")
# Step 7: Save the cleaned dataset
clean_file_path = r"D:\IoT-Smarthome.csv"
df.to_csv(clean_file_path, index=False)
print("✅ Cleaned file saved locally:", clean_file_path)

```

Fig. 5. Python code for dataset cleaning

4.1.1 System Architecture

The system architecture integrates IoT devices for capturing electricity consumption data and environmental variables. The information is transmitted through a wireless network to a cloud storage and processing layer using AWS. Subsequently, the data are queried and processed using Python, Pandas, and DuckDB, and are finally visualized in Power BI through an interactive dashboard. This structure enables an organized data flow from perception to analytical visualization (see Fig. 4).

4.1.2 Functional Requirements

The functional requirements define the capabilities needed to process, store, analyze, and visualize the energy consumption data from the dataset. The flow includes data reading and transformation in Python, processing with DuckDB, storage in AWS, and final visualization through Power BI (see Table 3).

4.1.3 Non-Functional Requirements

The non-functional requirements establish the quality conditions needed to ensure that the system operates with adequate performance, availability, security, and ease of use. These criteria complement the main functions and guarantee an efficient experience for the end user (see Table 4).

4.2 Sprint

At this stage, the user stories derived from the functional requirements were identified. Each story represents a system functionality and serves as the basis for sprint planning and execution.

4.2.1 Sprint 1: Project Definition and Dataset Search

Project Definition: The research area corresponds to Smart Homes, due to its relevance for monitoring and optimizing energy consumption through connected IoT devices. This solution contributes to reducing costs, improving household efficiency, and promoting the sustainable use of energy.

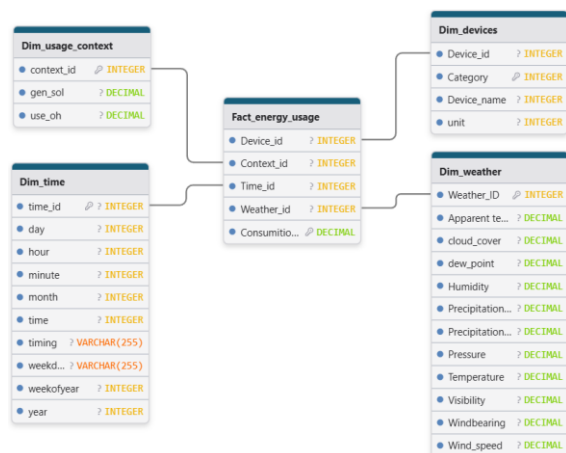


Fig. 6. Entity-relationship model

```
# =====
# 3.1. LOAD CLEAN DATA
# =====
df = pd.read_csv("IoT-Smarthome.csv")
df["time"] = pd.to_datetime(df["time"])

print(f"✅ Data loaded: {len(df):,} records")

# =====
# 3.2. TIME DIMENSION
# =====
dim_time = df[["time", "year", "month", "day", "weekday", "weekofyear", "hour", "minute", "timing"]
].drop_duplicates().copy()

dim_time["time_id"] = dim_time["time"].dt.strftime("%Y%m%d%H%M").astype(int)

dim_time = dim_time[["time_id", "time", "year", "month", "day", "weekday", "weekofyear", "hour", "minute", "timing"]]
dim_time.to_csv("dim_time.csv", index=False)
print(f"📄 dim_time.csv created ({len(dim_time):,} rows)")

# =====
# 3.5. DEVICES DIMENSION
# =====
device_cols = [
    "dishwasher", "home_office", "fridge", "wine_cellar", "garage_door", "barn", "well",
    "microwave", "living_room", "furnace", "kitchen",
    "car_charger_kw", "water_heater_kw", "air_conditioning_kw", "home_theater_kw",
    "outdoor_lights_kw", "microwave_kw", "laundry_kw", "pool_pump_kw"
]

dim_devices = pd.DataFrame({
    "device_id": range(1, len(device_cols) + 1),
    "device_name": device_cols,
    "unit": ["kW"] * len(device_cols),
    "category": ["main_circuit" if "_kw" not in c else "appliance" for c in device_cols]
})
dim_devices.to_csv("dim_devices.csv", index=False)
print(f"📄 dim_devices.csv created ({len(dim_devices):,} rows)")
```

Fig. 7. Python code for dataset normalization

The objective of the research is to design and implement an AWS-based centralized cloud system to analyze and visualize energy consumption, as well as to generate interactive reports in Power BI that support decision-making in energy efficiency.

Dataset Search: A search was conducted in specialized web sources to identify a dataset suitable for the study. Finally, “Augmented Smart Home with Weather Information” was selected because it contains records associated with energy consumption in smart homes and climate variables relevant to predictive modeling.

4.2.2 Sprint 2

In this sprint, dataset cleaning and normalization were performed using Python. The process included removing unnecessary columns, empty or duplicate records, as well as standardizing variable names and handling dates. In addition, libraries such as Pandas and DuckDB were used to facilitate efficient data processing and querying. As a result, a cleaned CSV file was generated, ready for the subsequent stages of analysis and visualization (see Fig. 5).

Data Normalization: After cleaning the dataset, data normalization was performed in order to structure the information more efficiently and consistently. For this purpose, an Entity-Relationship model was previously designed, which made it possible to define the main dimensions of the system, such as time, devices, climate, and usage context (see Fig. 6).

Subsequently, using Python, the data were reorganized into normalized tables, separating the variables according to their function within the model. This process made it possible to improve data integrity, reduce redundancies, and facilitate subsequent querying, processing, and predictive analysis (see Fig. 7).

4.2.3 Sprint 3: Connection and Data Upload to AWS:

Cloud Platform Selection: For the development of the research, Amazon Web Services (AWS) was selected as the cloud storage and processing platform.

Justification for Use: AWS enables secure, scalable, and highly available management of IoT data. In addition, it facilitates the centralization, querying, and analysis of large volumes of information.

Main Features: The platform provides access control through IAM, resilient storage with Amazon S3, and analytical services such as Athena.



Fig. 8. AWS bucket creation

```
Gonzalete@DESKTOP-HSS1KEQ MINGW64 ~/Desktop/Prueba IOT
$ python -m awscli configure
AWS Access Key ID [*****JMF7]:
AWS Secret Access Key [*****yCXh]:
Default region name [us-east-2]: us-east-2
Default output format [json]: json

credentials = session.get_credentials().get_frozen_credentials()
aws_access_key_id = credentials.access_key
aws_secret_access_key = credentials.secret_key
aws_session_token = credentials.token
```

Fig. 9. Python Code for Uploading the Normalized CSV Files to AWS

```
# ===== 4. UPLOAD TO AWS S3 =====
# Create boto3 session using default profile
session = boto3.Session(profile_name="default", region_name="us-east-2")
s3 = session.client("s3")

bucket = "datasetsmarthome"

# Dictionary of local files and target folders in S3
files_to_upload = {
    "dim_devices.csv": "dim_devices",
    "dim_time.csv": "dim_time",
    "dim_usage_context.csv": "dim_usage_context",
    "dim_weather.csv": "dim_weather",
    "fact_energy_usage.csv": "fact_energy_usage"
}

# Iterate and upload each file
for local_file, folder in files_to_upload.items():
    if folder:
        s3_key = f"{folder}/{local_file}" # folder + file
    else:
        s3_key = local_file # bucket root

    # Upload file
    s3.upload_file(local_file, bucket, s3_key)
    print(f"📁 File uploaded to s3://{bucket}/{s3_key}")
```

Fig. 10. Python code for uploading the normalized CSV files to AWS

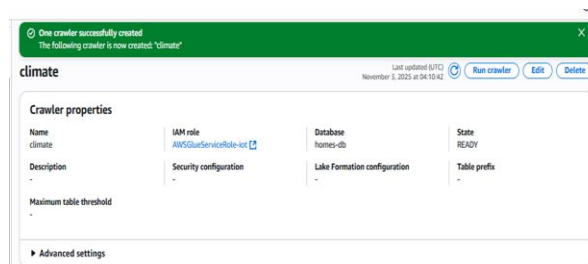


Fig. 11. AWS crawler creation

It also enables process automation, remote data access, and processing optimization for subsequent stages of analysis and visualization.

Connection and Data Upload to AWS: AWS Configuration: At this stage, an AWS S3 bucket was created by assigning a specific name and selecting the corresponding storage region. This configuration made it possible to enable a cloud repository for uploading, storing, and subsequently accessing the processed system data (see Fig. 8).

Connection with AWS: At this stage, the AWS credentials were configured within a Python virtual environment to establish the connection with Amazon S3. This configuration made it possible to authenticate access to the service, define the working region, and enable the automated upload of the normalized CSV files to the previously created bucket (see Fig. 9).

Data Upload to the Cloud: Once the connection with Amazon S3 was established, the normalized CSV files were uploaded using a Python script. This process made it possible to transfer the generated tables to the bucket configured in AWS, organizing them into folders according to their dimension or function within the data model (see Fig. 10).

4.2.4 Sprint 4: Power BI Connection to AWS

AWS Crawler Configuration: In this sprint, a crawler was configured in AWS Glue to identify and catalog the cleaned and normalized CSV files stored in Amazon S3. This process made it possible to create the corresponding database and make the data available for subsequent querying, integration, and visualization from Power BI (see Fig. 11).

ODBC Configuration: To establish the connection between AWS and Power BI, the Amazon Athena ODBC driver was configured. This driver acts as an intermediary between both platforms, enabling authentication of AWS credentials, definition of the region, catalog selection, and access to the schema where the processed data are located (see Fig. 12).

Power BI Configuration: After configuring the ODBC driver, the connection between Power BI and the database hosted on AWS was established. This connection made it possible to

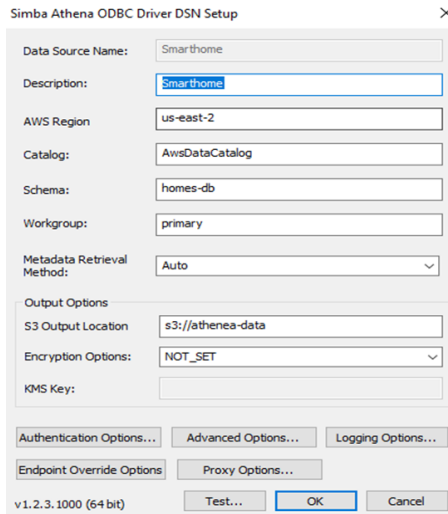


Fig. 12. ODBC driver configuration

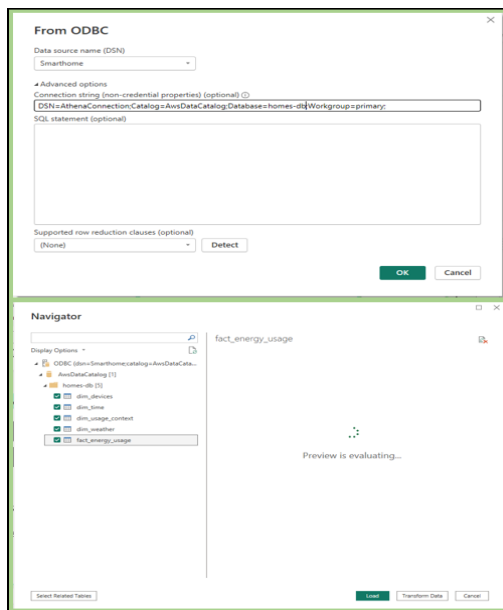


Fig. 13. Power BI connection to AWS through ODBC

access the processed tables, select the required data, and load them into Power BI for subsequent analysis and visualization through interactive dashboards (see Fig. 13).

4.2.5 Sprint 5: Dashboard Creation

In this sprint, interactive dashboards were developed to analyze energy consumption in



Fig. 14. Dashboard of overall electricity consumption and environmental data



Fig. 15. Dashboard of consumption by appliance and error analysis between actual and apparent temperature

smart homes. The panels make it possible to visualize overall electricity consumption, consumption by appliance, environmental variables, and model classification metrics, thereby facilitating the monitoring and analysis of energy information (see Fig. 14, Fig. 15, and Fig. 16). Likewise, versions adapted for mobile devices were designed in order to facilitate

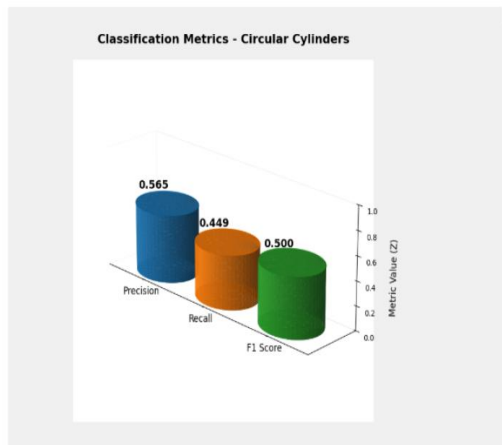


Fig. 16. Dashboard of classification metrics



Fig. 18. Mobile dashboard of consumption by appliance and analysis of actual and apparent temperature



Fig. 17. Mobile dashboard of overall electricity consumption and environmental data

access to the main indicators from more flexible query environments. These views make it possible to monitor overall consumption, environmental variables, and consumption by appliance from a mobile interface (see Fig. 17 and Fig. 18).

4.3 Deliverable Review

This phase made it possible to verify the fulfillment of the objectives of each sprint by evaluating whether the developed products met the established requirements and the quality



Fig. 19. Recall, Precision, F1-Score

criteria defined for the proposed solution (see Table 5).

4.4 Feedback and Comments

During the project review, consistent progress was observed in each sprint, highlighting the proper cleaning of the dataset, secure AWS configuration, and effective integration with Power BI.

It is recommended to optimize the ETL scripts through modularization in order to facilitate their maintenance, reuse, and scalability.

Table 5. Sprint definition

z	Aspects Reviewed	Review Result
Sprint 1: Project definition and search	Scope, dataset analysis, problem identification, preliminary architecture, and initial documentation.	The objectives were clearly defined; the dataset was validated, and the initial architecture was approved without observations.
Sprint 2: Data cleaning and normalization	Dataset quality, treatment of null and duplicate values, and normalization of columns and dates.	A clean and consistent dataset was obtained; the Python scripts work correctly.
Sprint 3: Connection and data upload to AWS	S3 configuration, IAM credentials, data upload, file integrity, and Python-AWS connection tests.	The connection was established securely; the dataset was correctly stored in S3, and the IAM credentials work properly.
Sprint 4: Power BI connection to AWS	Access from Power BI to AWS/Athena, extraction queries, initial visualizations, and basic filters.	Power BI was successfully connected to AWS; preliminary charts were generated, and the filters respond without errors.
Sprint 5: Final dashboards	Complete dashboards, KPIs, correlations, and advanced filters.	Functional, understandable, and suitable dashboards were delivered for the analysis of energy consumption.

Likewise, it is suggested to expand the technical documentation with more detailed flowcharts and visual evidence from each stage to strengthen development traceability.

Regarding data visualization, it is necessary to further deepen the interpretation of KPIs and correlations in order to improve the analysis of energy consumption.

Overall, the feedback confirms the appropriate management of cloud tools, a functional implementation, and continuous improvement in each project iteration.

5 Results and Discussion

5.1 Experimental Results

To evaluate the impact of the proposed system, five key indicators associated with energy consumption management in smart homes were measured: report preparation time, query response time, percentage of incorrect data, percentage of missing data, and user satisfaction.

For each indicator, 30 observations were collected before and after implementation, which made it possible to compare process performance under controlled experimental conditions (see Table 6).

5.1.1 Analysis of Precision, Recall, and F1- Score

To evaluate the performance of the predictive model, classification metrics were used to measure the system's ability to correctly distinguish positive and negative cases, considering both correct predictions and false positives and false negatives.

To evaluate the model's performance in detecting high temperatures, standard classification metrics were used: accuracy, precision, recall, and F1-score. These metrics make it possible to measure the model's ability to correctly identify records associated with high-temperature conditions.

Precision measures the accuracy in identifying positive instances:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

Recall indicates the system's ability to detect all positive cases:

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

F1-score combines precision and recall through their harmonic mean:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

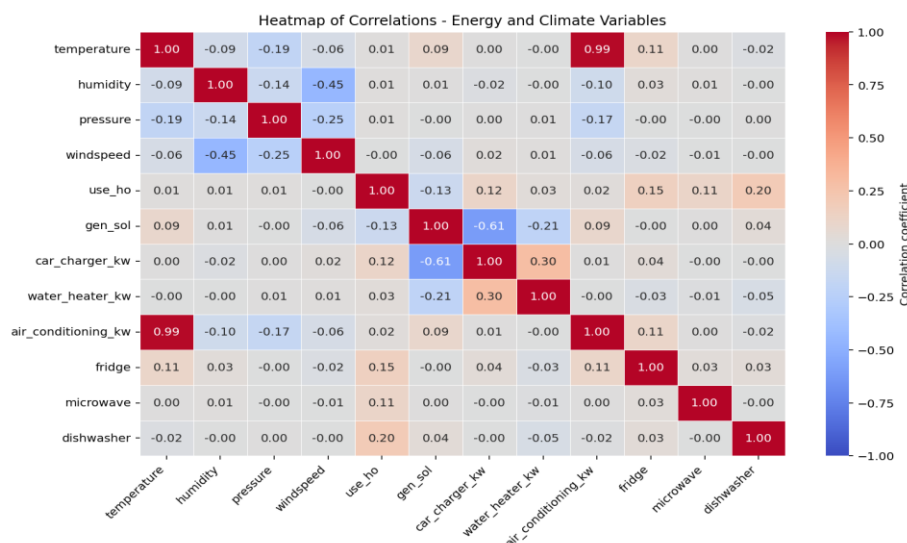


Fig. 20. Correlations between energy and climate variables

Next, the results obtained for each evaluated metric are presented, making it possible to clearly visualize the performance of the IoT system, as shown in Fig. 19.

The model demonstrates solid performance in detecting low temperatures according to the metrics obtained. The precision value of 0.915 indicates that more than 91% of the positive predictions were correct, maintaining a minimal level of false positives. The recall value of 0.84 shows that the system correctly identifies most real cases of low temperature, with few values going undetected. The F1-score of 0.88 confirms a strong balance between precision and recall, reflecting stable performance.

These results are comparable to those reported by Ramos and colleagues [28], who also obtained high metrics in precision (0.78-0.88), recall (0.89-0.90), and F1-score (0.82-0.88), demonstrating that his model achieved considerable reliability in predicting domestic activities. Likewise, Liu and colleagues [49], in their research on cooking activities in a realistic environment, achieved an average accuracy of 92.44% and an F1-score of 88.27%, which implies that their approach was realistic and valid. Srivatsa and Plötz [50], with their graph network model for recognizing domestic activities, achieved an F1-score of up to 92.4%, which implies that their system was robust even with

40% missing data in its sensors. Yuan and other authors [51], with their activity recognition system in smart spaces, reported an F1-score of up to 85% across the three different datasets analyzed. Fährmann and their team [52], using their BiGRU deep learning sensor-based model, obtained F1-score results of 94.91% for single occupancy and 91.55% for multiple occupancy; these results imply major improvements for safety in intelligent environments. Lentza and other researchers [53] evaluated multilabel classification methods for activity recognition in smart homes with multiple residents.

The RAKELd model with MLP achieved an F1-score of 0.909 in house 2 and showed a low Hamming loss, demonstrating the effectiveness of this approach against class imbalance. These results highlight the model's capacity to improve safety and well-being in intelligent environments, even with imbalanced data. Chen and coauthors [54] investigated the AttCLHAR model for human activity recognition with environmental sensors. The model outperformed other approaches, achieving an F1-score of 97.04% and precision of 97.02% in semi-supervised and transfer learning scenarios. These results indicate a significant improvement in model generalization, making it effective for working with unlabeled and imbalanced data, which is crucial for applications in smart homes. Islam and their research group

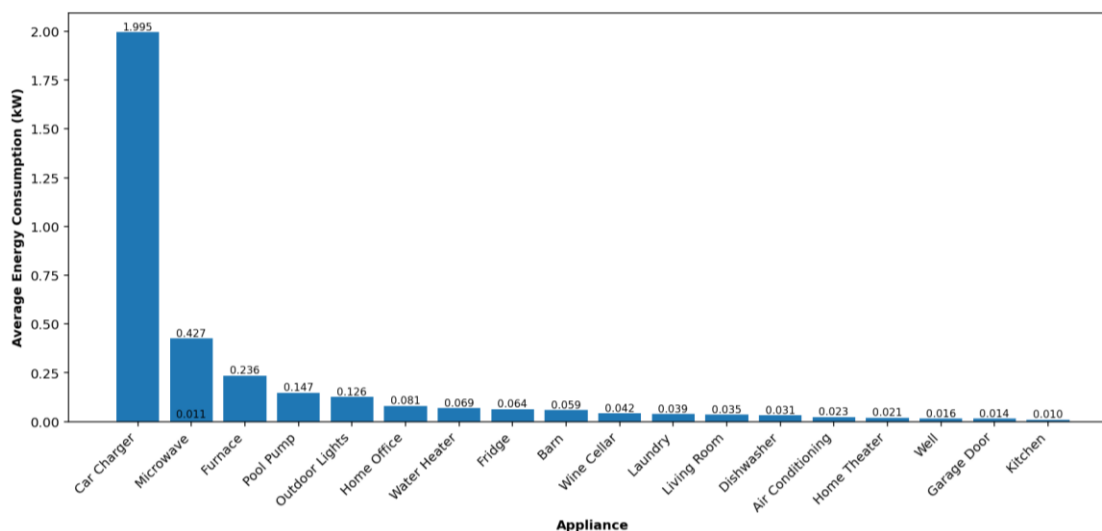


Fig. 21. Average energy consumption by appliance in smart homes

[55] presented HAMLET, a multimodal human activity recognition algorithm for collaborative robotic systems. The model outperformed other approaches, achieving a top-1 accuracy of 95.12% on the UTD-MHAD dataset and 97.45% on UT-Kinect, and an F1-score of 81.52% on UCSD-MIT. These results demonstrate that HAMLET effectively fuses unimodal data through a multimodal attention mechanism, improving the robot's ability to recognize human activities more accurately and robustly in real-world scenarios.

5.1.2 Heatmap

The heatmap was constructed from a correlation matrix in order to analyze the linear relationship between the energy and climate variables in the dataset. For this purpose, the Pearson correlation coefficient was used, calculated through the `corr()` function, whose values range from -1 to 1. A value close to 1 indicates a strong positive correlation, a value close to -1 represents a strong negative correlation, while a value close to 0 reflects a weak or nonexistent linear relationship (see Fig. 20).

The heatmap shows the linear relationship between climate variables and energy consumption variables. The most relevant result is observed between temperature and `air_conditioning_kw`, with a very high positive correlation of 0.99, indicating that an increase in

temperature is strongly associated with higher air-conditioning consumption. Likewise, a moderate negative correlation is observed between `gen_sol` and `car_charger_kw` (-0.61), as well as between humidity and windspeed (-0.45).

Overall, these results show that certain climate variables directly or indirectly influence energy consumption patterns in the smart home.

Teh and collaborators [16] investigated the availability and quality of datasets used in sensor studies, including smart homes. Using heatmaps, the study visualized which datasets are most frequently used and the coverage of data in these datasets. The results showed significant biases in dataset availability, with few well-covered public datasets. This implies that many studies depend on a limited number of datasets, highlighting the need for greater openness and standardization in data publication.

Park [57] analyzed domestic sensor data to detect redundant sensors and design a more efficient IoT system. Correlation matrices and heatmaps were used to identify strong correlations between environmental sensors and energy consumption. The heatmap revealed groups of highly correlated signals, which made it possible to propose sensor fusion or elimination. This implies cost savings and simplification in the design of IoT systems. Khan and colleagues [41] analyzed energy management in homes, using

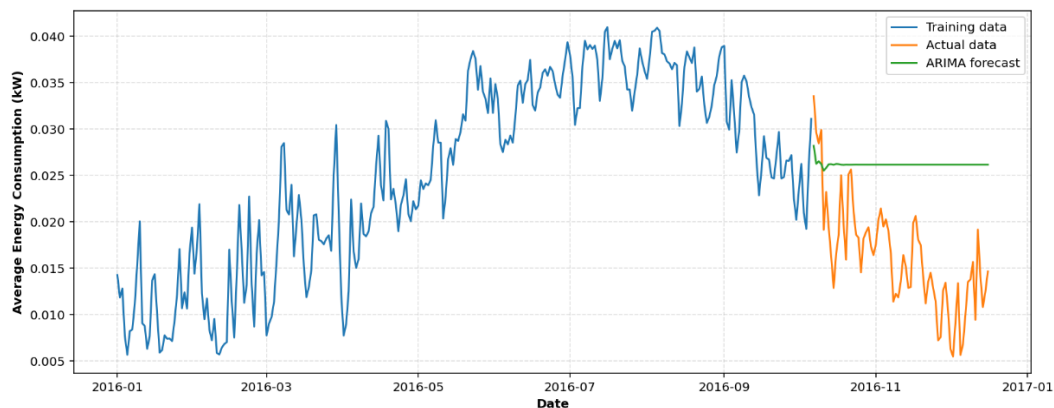


Fig. 22. ARIMA forecast of air-conditioning energy consumption

heatmaps to visualize correlations among energy consumption, climate conditions, and control variables. The heatmap facilitated the detection of negative and positive relationships, such as those involving the overall house and its oven. This implies the design of more precise control rules and the selection of better features for forecasting models.

Singh and Nagahara [58] worked with household consumption data to predict electrical self-sufficiency. They combined the LightGBM model with correlation analysis, visualized through heatmaps. The heatmap identified variables such as temperature, appliance-level consumption, and PV generation as highly correlated with self-sufficiency. This implies the selection of relevant features for predictive models and the support of energy efficiency policies.

Tonkin and other authors [59] published a multi-sensor dataset for domestic activity recognition. They included an exploratory analysis using correlation matrices and heatmaps among sensors such as accelerometers, environmental sensors, and presence sensors.

The heatmaps showed temporal and spatial correlations among sensors, which facilitated activity inference. This implies that the design of HAR models, or human activity recognition models, can take advantage of the sensor groupings detected through heatmaps. Tonkin and their team [59] addressed the prediction of user actions in smart home environments by presenting a synthetic and real dataset. They

used correlation heatmaps to explore temporal relationships between actions and sensors. The heatmaps made it possible to identify the time windows and sensors most relevant for predicting user actions. This analysis implies that correctly selecting the appropriate windows and sensors improves the proactivity of domestic agents in intelligent environments.

Dewangan and other authors [61] performed an analysis with heatmaps on household consumption series, identifying strong correlations between temperature and energy consumption. This guided data augmentation on the 'tempm' feature. It implies that understanding correlations makes it possible to generate coherent synthetic data and improve models when real data are limited.

5.2 Temporal Analysis of Energy Consumption in Smart Homes

Figures 21 and 22 present a complementary analysis of energy consumption in smart homes. First, the average consumption by appliance is identified, which makes it possible to recognize the devices with the highest energy demand within the household. Then, a temporal forecast is developed using ARIMA to analyze the future evolution of air-conditioning consumption, due to its direct relationship with climate conditions.

As shown in the results, the car charger presents the highest average energy consumption, with 1.995 kW, greatly exceeding the rest of the devices. In second place is the

Table 6. Post-Test Results of Gc and Ge for I1, I2, I3, and I4

No.	Report Time		Query Time		Incorrect %		Missing %		Citizen Satisfaction	
	Ge	Gc	Ge	Gc	Ge	Gc	Ge	Gc	Ge	Gc
1	3.2287	686	90.74	11564	0	4.44	0	2.74	5	2
2	2.1323	763	113.75	10805	0	3.89	0	2.12	4	3
3	3.9335	715	103.01	11923	0	5.52	0	3.22	5	2
4	4.0467	748	97.52	12069	0	6.67	0	3.52	5	3
5	2.9679	714	59.61	10430	0	5.78	0	3.37	4	2
6	3.9162	610	81.35	13762	0	6.65	0	2.13	5	3
7	2.5122	895	98.96	13187	0	5.72	0	2.11	5	2
8	2.6192	713	58.91	14882	0	7.42	0	3.16	4	2
9	4.0098	711	100.44	12714	0	7.44	0	3.66	5	2
10	3.9028	713	87.96	10781	0	6.06	0	2.91	4	2
11	3.9894	865	121.6	11016	0	6.45	0	2.75	5	2
12	3.6522	623	143.75	14805	0	5.8	0	2.61	5	2
13	3.4204	680	90.43	12100	0	5.86	0	3.79	4	3
14	3.8533	680	98.56	13040	0	7.36	0	2.78	4	1
15	2.8052	803	64.19	14554	0	5.59	0	2.52	4	2
16	3.5218	864	115.52	10008	0	6.53	0	3.44	4	3
17	2.1864	819	97.11	10886	0	6.8	0	3.53	4	2
18	3.4438	748	78.47	12058	0	6.3	0	3.8	5	3
19	2.3708	653	87.69	10115	0	5.05	0	2.9	5	1
20	3.5593	665	127.52	11186	0	7.48	0	3.18	5	2
21	2.3198	865	58.06	10817	0	5.31	0	3.61	5	3
22	4.0187	780	60.62	14423	0	5.83	0	3.2	4	3
23	3.0035	627	126.53	13124	0	7.82	0	2.84	5	1
24	3.4591	836	56.19	12833	0	5.84	0	3.44	5	3
25	3.7616	871	99.03	11165	0	5.28	0	3.04	5	2
26	3.4194	824	147.8	11515	0	6.48	0	1.53	5	3
27	3.458	601	57.72	10615	0	5.34	0	3.13	5	2
28	3.4876	854	76.2	10104	0	7.25	0	3.39	5	3
29	2.3164	610	83.59	12288	0	7.07	0	3.29	4	3
30	2.1904	822	137.31	10534	0	6.23	0	2.65	5	2

microwave with 0.427 kW, followed by the furnace with 0.236 kW and the pool pump with 0.147 kW.

These values show that energy consumption in the smart home is not uniformly distributed, but is mainly concentrated in high-demand devices. From a practical perspective, these results make

it possible to identify the equipment that should be prioritized in energy efficiency strategies.

In particular, the electric vehicle charger requires greater monitoring, time scheduling, and automated control due to its high impact on total demand. Likewise, devices such as the

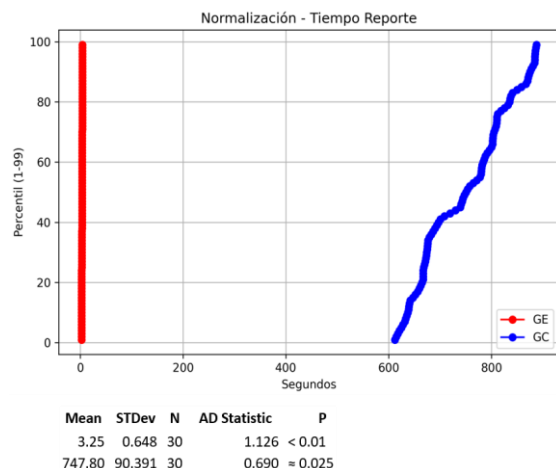


Fig. 23. Normality test for indicator 1

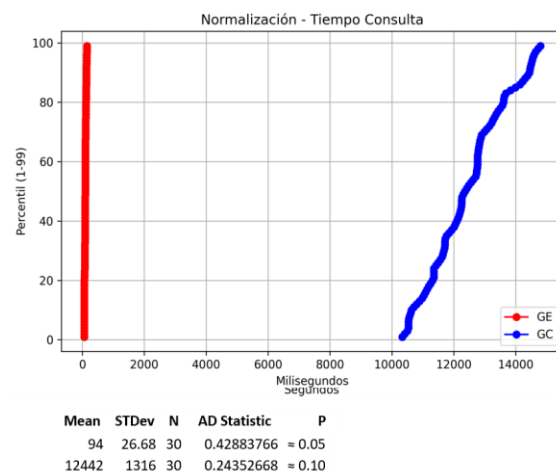


Fig. 24. Normality test for indicator 2

microwave, furnace, and pool pump can be considered in efficient-use policies, especially when the goal is to reduce costs and optimize residential consumption.

According to the results, the ARIMA model makes it possible to represent the temporal evolution of average air-conditioning consumption. The training series shows a progressive increase in consumption between January and August 2016, reaching values close to 0.040 kW, followed by a decrease in the subsequent months. This behavior suggests a possible seasonal relationship between air-conditioning use and climate conditions.

In the testing phase, the actual values show greater variability than the forecast generated by ARIMA. The forecasted line maintains a more stable trend, around 0.026 kW, indicating that the model captures the general behavior of the series but does not fully reproduce the abrupt fluctuations in consumption.

As an implication, the ARIMA forecast may be useful for anticipating general trends in energy demand associated with air conditioning. However, to improve accuracy in scenarios with high variability, it would be advisable to incorporate more robust or hybrid models, such as SARIMA, Prophet, Random Forest, XGBoost, or LSTM, integrating climate variables such as temperature, humidity, and apparent temperature.

5.3 Experimental Comparison between Gc and Ge

5.3.1 Experimental Results

Finally, comparative measurements were performed between the control group (Gc) and the experimental group (Ge), considering 30 observations per indicator. Variables associated with system efficiency and data quality were evaluated, including report preparation time, query response time, percentage of incorrect data, percentage of missing data, and user satisfaction (see Table 6).

The inclusion of these indicators is methodologically relevant because it makes it possible to analyze the real effect of implementing the IoT system based on AWS, DuckDB, and Power BI on energy data processing, error reduction, and the generation of useful information for decision-making in smart homes.

5.3.2 Normality Test

The normality test was applied to the indicators evaluated in the experimental group (Ge) and the control group (Gc) in order to verify the data distribution before performing the comparative tests. For this purpose, 30 observations per group were analyzed, considering indicators such as report time, query time, percentage of incorrect data, and percentage of missing data.

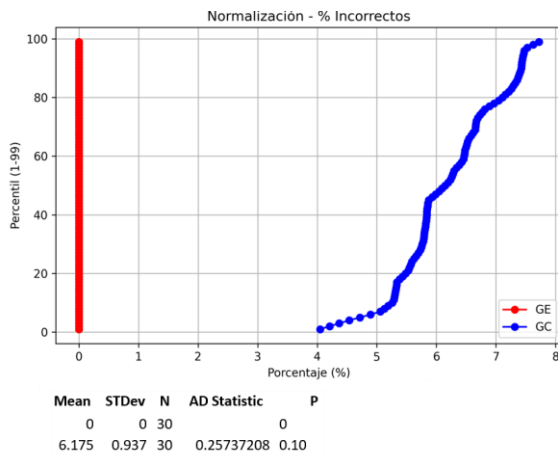


Fig. 25. Normality test for indicator 3

Fig. 23 shows the normality test applied to the report time indicator, comparing the behavior of the data in the experimental group (Ge) and the control group (Gc). The representation makes it possible to observe the distribution of the values recorded in both groups, as well as their main descriptive statistics and the associated significance value.

The results show p-values of < 0.01 for Ge and 0.025 for Gc, both below the significance level $\alpha = 0.05$. This indicates that the data from both groups do not exhibit normal behavior. In addition, a marked difference is observed between the means: Ge records 3.25 seconds, whereas Gc reaches 747.80 seconds, evidencing a considerable reduction in report time after system implementation.

Fig. 24 presents the normality test applied to the query time indicator, comparing the results of the experimental group (Ge) and the control group (Gc). The figure shows the percentile distribution of both groups, together with the mean, standard deviation, sample size, AD statistic, and p-value.

Ge records a mean of 94 ms, whereas Gc reaches 12,442 ms, evidencing a considerable reduction in response time after system implementation. Regarding normality, Ge presents a p-value of approximately 0.05 and Gc a p-value of approximately 0.10; therefore, the data show behavior close to or compatible with normality at the significance level $\alpha = 0.05$,

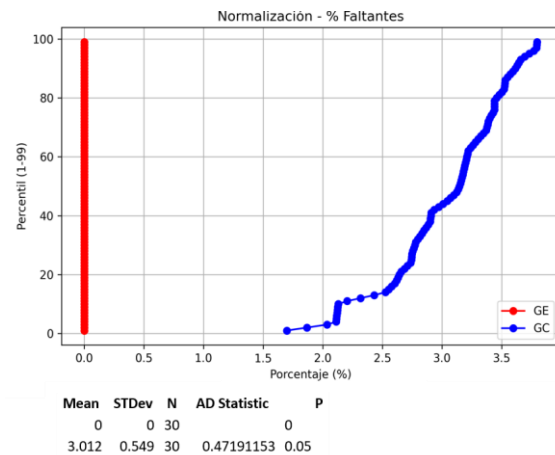


Fig. 26. Normality test for indicator 4

although Ge should be interpreted with caution because it is at the limit of the statistical criterion.

Fig. 25 presents the normality test applied to the percentage of incorrect data indicator, comparing the results of the experimental group (Ge) and the control group (Gc). The figure shows the percentile distribution of both groups, together with the mean, standard deviation, sample size, AD statistic, and p-value.

Ge presents a mean of 0%, whereas Gc records a mean of 6.175%, evidencing an improvement in data quality after system implementation. Regarding normality, Gc shows a p-value of 0.10, higher than $\alpha = 0.05$; therefore, its data present behavior compatible with normality. In the case of Ge, since it records constant values of 0%, there is not enough variability to evaluate a normal distribution.

Fig. 26 presents the normality test applied to the percentage of missing data indicator, comparing the results of the experimental group (Ge) and the control group (Gc). The figure shows the percentile distribution of the data, together with the mean, standard deviation, sample size, AD statistic, and p-value for each group.

Ge presents a mean of 0%, whereas Gc records a mean of 3.012%, evidencing that the implementation of the system made it possible to eliminate missing values in data processing. Regarding normality, Gc shows a p-value of 0.05, located at the threshold of the significance level $\alpha = 0.05$. In the case of Ge, since it presents

Table 7. Descriptive statistics results

Sample	No.	Mean	StDev	AD	p-value
I1: Post-test Ge	30	3.25	0.648	1.126	0.01
I1: Post-test Gc	30	747.8	90.39	0.69	0.025
I2: Post-test Ge	30	94	26.68	0.429	0.5
I2: Post-test Gc	30	12442	1316	0.244	0.1
I3: Post-test Ge	30	0	0	0	0
I3: Post-test Gc	30	6.175	0.937	0.257	0.1
I4: Post-test Ge	30	0	0	0	0
I4: Post-test Gc	30	3.012	0.549	0.472	0.05

constant values of 0%, there is not enough variability to evaluate normal behavior.

5.3.3 Descriptive Statistics and Distribution Analysis

To complement the normality test, a descriptive statistical analysis was conducted on the indicators evaluated in the post-test, considering the experimental group (Ge) and the control group (Gc). This analysis made it possible to characterize data behavior through measures of central tendency, dispersion, confidence intervals, skewness, and kurtosis.

Its application is methodologically relevant because it makes it possible to evaluate the stability and consistency of the results obtained for each indicator, especially in terms of report time, query time, percentage of incorrect data, and percentage of missing data, as shown in Tables 7 and 8.

The descriptive results show clear differences between the experimental group (Ge) and the control group (Gc). In I1, report time, Ge obtained a mean of 3.25 s, compared with 747.8 s in Gc. Likewise, in I2, query time, Ge recorded 94 ms, whereas Gc reached 12,442 ms, evidencing a substantial improvement in the system's operational efficiency. Regarding data quality, Ge presented 0% incorrect data and 0% missing data, compared with Gc, which recorded 6.175% and 3.012%, respectively. This confirms that the

implemented solution improved the integrity and reliability of the processed information.

As an implication, the system based on IoT, AWS, DuckDB, and Power BI optimizes report generation, reduces query times, and improves data quality, strengthening decision-making on energy consumption in smart homes.

For Indicator I1 (Report Time)

I1, report time, shows a considerable difference between both groups: Gc obtained a mean of 747.8 s, whereas Ge reached 3.25 s, evidencing a substantial reduction after system implementation. However, in both groups the p-values were lower than $\alpha = 0.05$, so the assumption of normality was not met.

In Ge, the 95% confidence interval ranged from 3.018 to 3.482 s, reflecting stability in the times obtained. The skewness value of -0.472 indicates a slight concentration toward lower values, the kurtosis value of -1.19 suggests the absence of relevant extreme values, and $Q3 = 3.83$ confirms that 75% of the reports were generated in times equal to or lower than this value.

Among related studies involving faster execution than the standard approach is the paper presented by Boškov in collaboration with other authors [18], Time-to-Provision Evaluation of IoT Devices Using Automated Zero-Touch Provisioning, which aimed to improve the Soft-AP method by proposing ZTP solutions based on

Table 8. Summary of Results for the Indicators

Indicator	N	Mean	StDev	95% Confidence Intervals for the Mean	Kurtosis	Skewness	Q3
I1: Report Time (Ge)	30	3.25	0.648	3.018-3.482 sec.	-1.19	-0.472	3.83
I2: Query Time (Ge)	30	94.005	26.667	84.45-103.55 ms	-0.75	0.301	11.605
I3: % Incorrect Data (Ge)	30	0	0	0	0	0	0
I4: % Missing Data (Ge)	30	0	0	0	0	0	0

Soft-AP and Bluetooth. These solutions achieved improvements from 54% to 434% compared with experts, and from 313% to 880% compared with procedures performed by non-experts.

This result shows that the implementation of the system drastically reduces report generation time, moving from a slow process in Gc to an almost immediate response in Ge. In practical terms, this improves operational efficiency, facilitates timely monitoring of energy consumption, and allows users to make faster decisions regarding energy use in the smart home.

For Indicator I2 (Query Time)

I2, query time, shows a substantial improvement in Ge, with a mean of 94.005 ms, compared with 12,442 ms in Gc. The 95% confidence interval for Ge ranged from 84.45 to 103.55 ms, confirming stability in response times. The standard deviation of 26.667 ms shows controlled dispersion. Likewise, the skewness value of 0.301 indicates a slight concentration toward lower values, while the kurtosis value of -0.75 suggests the absence of relevant extreme values.

The results obtained in this study are related to previous research showing substantial improvements after the implementation of an optimization algorithm. For example, Sugiartana and collaborators [10] reported that a Gc had an average time of 5.045 seconds, while after applying optimization to a Ge, a response time of 0.408 seconds was achieved. Likewise, Xiaowei [13] shows that processing 1,000 data volumes takes 5.99 seconds, while processing 4,000 volumes takes a total of 8.05 seconds, demonstrating the high reliability of the optimized

model applied to the dataset. Zhang (2025) found that ML- and RL-based query optimizers do not always generalize well; they may perform poorly under certain workloads, which affects response time when executing complex queries. These methods are compared with simpler adaptive query processing techniques, suggesting that, in many cases, simpler methods may offer better response times. Reiss and colleagues [47], when working with stream data and very large historical data, involving millions of records, observed that several queries required considerable time.

They used bitmap indexes and read optimizations, showing that certain queries can respond within 0.1 to 1 second when the dataset is manageable. Izaguirre and other authors [48] observed in online surveys that high response times can affect data quality and participation, since delays in queries or forms may discourage responses.

Therefore, ensuring low loading and response times and properly managing the interface and back end can improve the user experience and the quality of the data obtained. Chi and Chi [65] indicate that traditional smart home systems had slow query times, which delayed device response. To address this issue, they implemented a big data architecture with distributed cache and an optimized algorithm, reducing query and control time to less than 1.4 seconds.

Wang and their team [66] propose an offloading algorithm called BMDCO, designed to dynamically decide which tasks should be executed locally and which should be sent to the edge in order to minimize delay. In simulations, the method notably reduces response time and

the average queue length compared with conventional approaches, as shown in the backlog and latency graphs.

This implies that intelligent offloading management can significantly improve query time in IoT and smart home systems. Iggena and other authors [67] found that IoT queries were slow because data were dispersed and lacked a common index, generating high latency and overload under concurrency. To solve this, they created IoTcrawler, an engine with unified metadata, distributed indexing, and caching, allowing queries to be answered from the index instead of from the devices. This significantly reduced latency and maintained stable response times even with many users.

The system enables almost immediate queries, which improves access to processed energy information and facilitates timely analysis through tools such as DuckDB, AWS, and Power BI. This strengthens decision-making in smart homes based on updated and rapidly available data.

For Indicator I3 (% Incorrect Data)

I3, percentage of incorrect data, shows a total improvement in Ge, with a mean of 0%, compared with 6.175% recorded in Gc. This evidences that the implemented system eliminated the errors detected in the processed records. In Ge, the standard deviation was 0, indicating absence of variability because all values remained constant at 0%.

The findings of this study, specifically the elimination of duplicate or erroneous data, are related to the research conducted by Teh and their research group [16] in their work *Sensor Data Quality: A Systematic Review*, where they used their own dataset with 13,057 records. After applying an optimization model that detected duplicate records, 6,193 duplicate records were identified and removed, guaranteeing dataset reliability. Similarly, Adewole and collaborators [17] had to clean their data before using them to feed their ML model. After implementing data cleaning on their sample, 308,381 duplicate samples were identified out of a total of 2,830,743, thereby avoiding potential bias in the model.

Aguilar Suarez and Beroza [42] found that seismology datasets used for ML contained widespread label errors, including unlabeled earthquakes, noise classified as earthquakes, mislabeled arrival times, missing signals, and other issues. They reported an average error of 3.9% in the evaluated datasets. To improve data quality, they flagged the erroneous examples, generating a list of problematic cases with the intention that these examples be excluded from future training. Garza and colleagues [40] found highly variable error rates, ranging from as low as 2 errors per 10,000 fields to 2,784 errors per 10,000 fields. For the most commonly used method, MRA or medical record abstraction, the pooled error was approximately 6.57%.

They recommend prioritizing lower-error methods, for example, double-data entry, which shows very low rates of approximately 0.14%, compared with MRA, approximately 6.57%. Singh and Nagahara [58] investigated household energy data analysis to predict electrical self-sufficiency and used heatmaps to explore relationships. They found that 26.1% of households, 151 out of 578, were discarded due to critical inaccuracies in the data. This implies that dataset quality can bias results, and that it is essential to filter and clean data and document losses before modeling. Ou together with other researchers [56] compared time-series models for household consumption and quantified MAPE improvements across architectures.

They reported a relative improvement from 4.52% to 10.53% in MAPE, depending on the prediction horizon, for their best model. This implies that architecture selection and prediction horizon affect the percentage error, with trade-offs between accuracy and horizon. Dewangan and other authors [61] reviewed and conducted experiments with smart meters and smart homes, presenting representative MAPE metrics. In similar contexts, they reported a MAPE of approximately 5.055% in certain models, which serves as a reference value in comparative studies.

This implies that a MAPE of 5-10% is considered good performance in household consumption and can serve as a benchmark when evaluating new models or datasets. Khan and their team [41] worked on forecasting and

energy control in smart homes, evaluating models using error metrics. They reported a MAPE of approximately 12.3% for their best model in the described tests.

This implies that, although the model is useful for planning, there is still an important margin of error for critical applications, suggesting the need for more data and greater explainability.

The result confirms that the solution improves the quality of the processed data by reducing inconsistencies and errors in the information. This strengthens the system's reliability for analyzing energy consumption and generating accurate reports in smart homes.

For Indicator I4 (% Missing Data)

I4, percentage of missing data, shows a total improvement in Ge, with a mean of 0%, compared with 3.012% recorded in Gc. This result indicates that, after system implementation, no null or absent values were identified in the processed data. In addition, the standard deviation of Ge was 0, confirming total stability in this indicator.

Other studies aimed at addressing missing data include the research by Sayeed and coauthors [14], which used imputation techniques to solve these problems by replacing missing values with estimated substitutes. It can also be observed that the data obtained in our research are not as volatile as the dataset studied by Prakash and Kumar [15], who also analyzed electrical consumption directly from sensors.

However, there were moments in which data discrepancies occurred, resulting in empty data that could compromise several smart grid functionalities.

Alwaterr together with other researchers [62] explained that missing values in their analysis caused bias and reduced statistical effectiveness, or could even invalidate conclusions.

To address this, they proposed using imputation techniques, either through mean values, last observation, or the implementation of probabilistic models, machine learning algorithms, and deep learning. Ren and other authors [63] identified three main difficulties when analyzing missing values: 1) missingness mechanisms (MCAR, MAR, MNAR); 2) missingness patterns; and 3) missing data rates. They compared different strategies, such

as deleting cases with missing values, complete-case analysis and pairwise deletion, and imputation methods, including mean/mode, KNN, MICE, and trees, evaluating the performance of each one.

Majib and collaborators [25] published the DOMUS dataset and reported one week of missing data due to a hub outage during a 52-week campaign, representing 1.9% lost sampling.

This implies that DOMUS users should apply imputation or robustness against gaps and document temporal biases in their experiments. Kwon and Kim [31] studied imputation in smart meter series and tested scenarios with artificial missing rates between 5% and 50%.

They documented that performance drops significantly when the missing rate exceeds 15-20% if imputation is not performed correctly. This implies that LSTM models can handle moderate missingness, but they are not effective at high rates without losing accuracy.

Choi and colleagues [33] conducted experiments on environmental data, similar to smart home sensors, generating missing rates, and reported that predictive performance degrades significantly when missing data exceed 15%.

They propose using ensemble methods and imputation to mitigate this problem. This implies that, above 15%, robust imputation is no longer optional.

Decorte and other authors [64] evaluated 12 imputation methods in environmental monitoring, applicable to smart homes, and presented experiments with real and synthetic missingness. They reported significant differences in performance when the missing data rate exceeds 20-30%. This implies that, in household datasets, more than 20% missing data requires the use of advanced strategies, especially if spatial or temporal correlation exists among sensors.

The elimination of missing data strengthens dataset integrity and improves the reliability of energy analysis. This makes it possible to generate more consistent reports, reduce visualization errors, and support more accurate decisions regarding energy consumption in smart homes.

6 Conclusions and Future Research

This research aimed to design, implement, and validate a system for monitoring and analyzing energy consumption in smart homes by integrating IoT sensors, Python, DuckDB, AWS, and Power BI. The results show that the proposed architecture improved operational efficiency and the quality of processed data, in line with the purpose of building a scalable system for residential energy management.

Regarding report time, the experimental group obtained an average of 3.25 seconds, compared with 747.8 seconds in the control group, evidencing a considerable reduction in report generation.

Likewise, query time reached 94.005 ms in the experimental group, compared with 12,442 ms in the control group, demonstrating an almost immediate response for energy data analysis.

These results confirm that the combined use of AWS, DuckDB, and Power BI supports fast querying, efficient visualization, and timely decision-making. Regarding data quality, the system achieved 0% incorrect data and 0% missing data in the experimental group.

This result evidences that the data cleaning, normalization, and validation process strengthened the integrity of the information used for energy analysis. Consequently, the implemented solution improves not only processing efficiency but also the reliability of the reports generated for smart homes.

The main limitations were associated with the configuration of permissions in AWS Athena for data transfer to Power BI and the initial treatment of columns with null or non-informative values within the dataset.

As future research lines, it is recommended to incorporate machine learning and artificial intelligence models to strengthen energy consumption prediction, as well as to integrate real-time physical sensors to capture data directly from smart homes. Likewise, it would be relevant to evaluate hybrid models, such as XGBoost, LSTM, or SARIMA, and incorporate additional climate variables to improve predictive accuracy and expand the applicability of the system in real residential scenarios.

References

1. Condon, F.J., Martínez, J.M., Eltamaly, A.M. (2023). Design and Implementation of a Cloud-IoT-Based Home Energy Management System. *Sensors*, Vol. 23, No. 1, pp. 176. DOI: 10.3390/s23010176.
2. Mehmood, A., Lee, K.-T., Kim, D.-H. (2023). Energy Prediction and Optimization for Smart Homes with Weather Metric-Weight Coefficients. *Sensors*, Vol. 23, No. 7, pp. 3640. DOI: 10.3390/s23073640.
3. Oprea, S.-V., Bâra, A. (2024). A Recommendation System for Prosumers Based on Large Language Models. *Sensors*, Vol. 24, No. 11, pp. 3530. DOI: 10.3390/s24113530.
4. Fay, C.D., Corcoran, B., Diamond, D. (2024). Green IoT Event Detection for Carbon-Emission Monitoring in Sensor Networks. *Sensors*, Vol. 24, No. 1, pp. 162. DOI: 10.3390/s24010162.
5. Meteier, Q., El Kamali, M., Angelini, L. (2023). A Recommender System for Increasing Energy Efficiency of Solar-Powered Smart Homes. *Sensors*, Vol. 23, No. 18, pp. 7974. DOI: 10.3390/s23187974.
6. Habibu, M.A., Sivakumar, S., Kanagachidambaresan, G.R. (2025). An Effective IoT-Based Demand Response for Energy-Efficient Smart Homes. *Energy Informatics*, Vol. 8, pp. 125. DOI: 10.1186/s42162-025-00590-w.
7. Quesada, C., Astigarraga, L., Merveille, C. (2024). An Electricity Smart Meter Dataset of Spanish Households: Insights into Consumption Patterns. *Scientific Data*, Vol. 11, pp. 59. DOI: 10.1038/s41597-023-02846-0.
8. Li, W., Yigitcanlar, T., Nili, A. (2025). Responsible Smart Home Technology Adoption: Exploring Public Perceptions and Key Adoption Factors. *Internet of Things*, Vol. 32, pp. 101622. DOI: 10.1016/j.iot.2025.101622.
9. Wenninger, M., Maier, A., Schmidt, J. (2021). DEDDIAG, a Domestic Electricity Demand Dataset of Individual Appliances in

- Germany. Scientific Data, Vol. 8, pp. 176. DOI: 10.1038/s41597-021-00963-2.
10. **Nursuwars, F.M.S., Shofa, R.N., Andang, A. (2024).** IoT APIs: Time Response Optimization in Edge Computing Data Communication for Power Phase Detection System. E3S Web of Conferences, Vol. 450, pp. 01013. DOI: 10.1051/e3sconf/202450001013.
 11. **Al-Dahhan, A., Hussien, A., Baker, T. (2025).** IntEnergy: An Efficient Federated Learning Model for Energy Consumption Forecasting in Smart Homes. Discover Internet of Things, Vol. 5, pp. 93. DOI: 10.1007/s43926-025-00197-3.
 12. **Al-Kababji, A., Alsalem, A., Himeur, Y. (2022).** Interactive Visual Study for Residential Energy Consumption Data. Journal of Cleaner Production, Vol. 366, pp. 132–841. DOI: 10.1016/j.jclepro.2022.132841.
 13. **Ma, X. (2024).** The Analysis of the Internet of Things Database Query and Optimization Using Deep Learning Network Model. PLoS ONE, Vol. 19, No. 6, pp. 0306291. DOI: 10.1371/journal.pone.0306291.
 14. **Sayeed, S., Ahmad, A.F., Peng, T.C. (2024).** Smartic: A Smart Tool for Big Data Analytics and IoT. F1000Research, Vol. 11, pp. 17. DOI: 10.12688/f1000research.73613.2.
 15. **Prakash, K.P., Kumar, Y.V.P. (2021).** Simple and Effective Descriptive Analysis of Missing Data Anomalies in Smart Home Energy Consumption Readings. Journal of Energy Systems, Vol. 5, No. 3, pp. 199–220. DOI: 10.30521/jes.878318.
 16. **Teh, H.Y., Kempa-Liehr, A.W., Wang, K.I. (2020).** Sensor Data Quality: A Systematic Review. Journal of Big Data, Vol. 7, pp. 11. DOI: 10.1186/s40537-020-0285-1.
 17. **Adewole, K.S., Jacobsson, A., Davidsson, P. (2025).** Intrusion Detection Framework for Internet of Things with Rule Induction for Model Explanation. Sensors, Vol. 25, No. 6, pp. 1845. DOI: 10.3390/s25061845.
 18. **Boškov, I., Yetgin, H., Vučnik, M. (2020).** Time-to-Provision Evaluation of IoT Devices Using Automated Zero-Touch Provisioning. Proceedings of the IEEE International Conference on Communications (ICC). DOI: 10.1109/icc40277.2020.9149089.
 19. **Chakraborty, A., Islam, M., Shahriyar, F. (2023).** Smart Home System: A Comprehensive Review. Journal of Electrical and Computer Engineering, Vol. 2023, p. 761–6683. DOI: 10.1155/2023/7616683.
 20. **Gouiza, N., Jebari, H., Reklaoui, K. (2024).** Integration of IoT-Enabled Technologies and Artificial Intelligence in Diverse Domains: Recent Advancements and Future Trends. Journal of Theoretical and Applied Information Technology, Vol. 102, No. 5, pp. 1975–2029.
 21. **Djumanazarov, O., Väänänen, A., Haataja, K. (2022).** An Overview of IoT-Based Architecture Model for Smart Home Systems. Intelligent Systems Design and Applications, pp. 696–706. DOI: 10.1007/978-3-030-96308-8_65.
 22. **Bouchabou, D., Nguyen, S.M., Lohr, C. (2021).** A Survey of Human Activity Recognition in Smart Homes Based on IoT Sensors Algorithms: Taxonomies, Challenges, and Opportunities with Deep Learning. Sensors, Vol. 21, No. 18, pp. 6037. DOI: 10.3390/s21186037.
 23. **Purna Prakash, K., Pavan Kumar, Y.V. (2022).** Analytical Approach to Exploring the Missing Data Behavior in Smart Home Energy Consumption Dataset. Journal of Renewable Energy and Environment, Vol. 9, No. 2, pp. 37–48. DOI: 10.30501/jree.2021.313536.1277.
 24. **Singh, D., Merdivan, E., Hanke, S. (2017).** Convolutional and Recurrent Neural Networks for Activity Recognition in Smart Environment. Towards Integrative Machine Learning and Knowledge Extraction, Vol. 10344, pp. 194–205. DOI: 10.1007/978-3-319-69775-8_12.
 25. **Majib, Y., Alosaimi, M., Asaturyan, A. (2023).** Dataset for Cyber-Physical Anomaly Detection in Smart Homes. Frontiers in the Internet of Things, Vol. 2, pp. 127–5080. DOI: 10.3389/friot.2023.1275080.

26. **Yan, J. (2024).** AIoT in Smart Homes: Challenges, Strategic Solutions, and Future Directions, *Highlights in Science. Engineering and Technology*, Vol. 87, pp. 59–65. DOI: 10.54097/8hzgaf51.
27. **Buil-Gil, D., Kemp, S., Kuenzel, S. (2022).** The Digital Harms of Smart Home Devices: A Systematic Literature Review. *arXiv*. DOI: 10.48550/arxiv.2209.05458.
28. **Ramos, R.G., Domingo, J.D., Zalama, E. (2022).** SDHAR-HOME: A Sensor Dataset for Human Activity Recognition at Home. *Sensors*, Vol. 22, pp. 8109. DOI: 10.3390/s22218109.
29. **Dasappa, N.S., Kumar G, K., Somu, N. (2024).** Multi-Sensor Data Fusion Framework for Energy Optimization in Smart Homes. *Renewable and Sustainable Energy Reviews*, Vol. 193, pp. 114–235. DOI: 10.1016/j.rser.2023.114235.
30. **Sun, F., Chen, G., Ma, H. (2025).** Multi-Objective IoT Service Composition with Replication Using Memetic NSGA-II With Bottleneck-Driven Local Search. *Future Generation Computer Systems*, Vol. 175, pp. 108039. DOI: 10.1016/j.future.2025.108039.
31. **Kwon, H.-R., Kim, P.-K. (2021).** A Missing Data Compensation Method Using LSTM Estimates and Weights in AMI System. *Information*, Vol. 12, No. 9, pp. 341. DOI: 10.3390/info12090341.
32. **Iturbe-Araya, J.I., Rifà-Pous, H. (2025).** Enhancing Unsupervised Anomaly-Based Cyberattacks Detection in Smart Homes through Hyperparameter Optimization. *International Journal of Information Security*, Vol. 24, pp. 45. DOI: 10.1007/s10207-024-00961-6.
33. **Choi, C., Jung, H., Cho, J. (2021).** An Ensemble Method for Missing Data of Environmental Sensor Considering Univariate and Multivariate Characteristics. *Sensors*, Vol. 21, No. 22, pp. 7595. DOI: 10.3390/s21227595.
34. **Huang, X., Zhang, S. (2023).** Human Activity Recognition Based on Transformer in Smart Home. *Proceedings of the 2023 2nd Asia Conference on Algorithms, Computing and Machine Learning*, pp. 520–525. DOI: 10.1145/3590003.3590100.
35. **Al-Kababji, A., Alsalem, A., Himeur, Y. (2022).** Interactive Visual Study for Residential Energy Consumption Data. *Journal of Cleaner Production*, Vol. 366, pp. 132–841. DOI: 10.1016/j.jclepro.2022.132841.
36. **Arrotta, L., Bettini, C., Civitarese, G. (2023).** MICAR: Multi-Inhabitant Context-Aware Activity Recognition in Home Environments. *Distributed and Parallel Databases*, Vol. 41, pp. 571–602. DOI: 10.1007/s10619-022-07403-z.
37. **Keleşoğlu, N., Sobczak, L. (2024).** ZigBeeNet: Decrypted Zigbee IoT Network Traffic Dataset in Smart Home Environment. *Applied Sciences*, Vol. 14, pp. 10844. DOI: 10.3390/app142310844.
38. **Pfitzinger, T., Koch, M., Schlenke, F. (2024).** Multi-Modal Dataset of Human Activities of Daily Living with Ambient Audio, Vibration, and Environmental Data. *Data*, Vol. 9, No. 12, pp. 144. DOI: 10.3390/data9120144.
39. **Köckemann, U., Alirezaie, M., Renoux, J. (2020).** Open-Source Data Collection and Data Sets for Activity Recognition in Smart Homes. *Sensors*, Vol. 20, No. 3, pp. 879. DOI: 10.3390/s20030879.
40. **Garza, M.Y., Williams, T., Ounpraseuth, S. (2023).** Error Rates of Data Processing Methods in Clinical Research: A Systematic Review and Meta-Analysis of Manuscripts Identified Through PubMed. *Research Square*. DOI: 10.21203/rs.3.rs-2386986/v2.
41. **Khan, M.A., Sabahat, Z., Farooq, M.S. (2024).** Optimizing Smart Home Energy Management for Sustainability Using Machine Learning Techniques. *Discover Sustainability*, Vol. 5, pp. 430. DOI: 10.1007/s43621-024-00681-w.
42. **Aguilar Suarez, A.L., Beroza, G. (2025).** Pervasive Label Errors in Seismological Machine Learning Datasets. *arXiv*. DOI: 10.48550/arxiv.2511.09805.
43. **Pizarro, R., Valle, R., Buenaposada, J.M. (2025).** Pose-Guided Token Selection for the

- Recognition of Activities of Daily Living. Image and Vision Computing, Vol. 162, pp. 105–686. DOI: 10.1016/j.imavis.2025.105686.
44. **Gamboa-Cruzado, J., Blanco-Maslucan, R., Galindo-Arone, D. (2025).** Augmented Reality as a Tool for Cultural Learning in Educational Institutions in Peru. *Computación y Sistemas*, Vol. 29, No. 2, pp. 901–915. DOI: 10.13053/cys-29-2-5739.
 45. **Elassy, M., Al-Hattab, M., Takruri, M. (2024).** Intelligent Transportation Systems for Sustainable Smart Cities. *Transportation Engineering*, Vol. 16, pp. 100–252. DOI: 10.1016/j.treng.2024.100252.
 46. **Wahla, S.Q., Ghani, M.U. (2023).** Visual Fall Detection from Activities of Daily Living for Assistive Living. *IEEE Access*, Vol. 11, pp. 108876–108890. DOI: 10.1109/access.2023.3321192.
 47. **Reiss, F., Stockinger, K., Wu, K. (2007).** Enabling Real-Time Querying of Live and Historical Stream Data. *Proceedings of the 19th International Conference on Scientific and Statistical Database Management (SSDBM)*, pp. 28. DOI: 10.1109/ssdbm.2007.34.
 48. **Izaguirre Olmedo, J.A., Jordán Correa, D.P., Palacios Sarmiento, T.Y. (2025).** Response Time Efficiency and Data Quality in Online Surveys. *Revista Venezolana de Gerencia*, Vol. 30, No. 13, pp. 407–419. DOI: 10.52080/rvgluz.30.especial13.27.
 49. **Liu, M., Suh, S., Zhou, B. (2023).** Smart-Badge: A Wearable Badge with Multi-Modal Sensors for Kitchen Activity Recognition (v2). *arXiv*. DOI: 10.48550/arxiv.2210.00888.
 50. **Srivatsa, P., Plötz, T. (2024).** Using Graphs to Perform Effective Sensor-Based Human Activity Recognition in Smart Homes. *Sensors*, Vol. 24, No. 12, pp. 3944. DOI: 10.3390/s24123944.
 51. **Yuan, S., Pargoo, N.S., Ortiz, J. (2025).** GeXSe (Generative Explanatory Sensor System): An Interpretable Deep Generative Model for Human Activity Recognition in Smart Spaces (v3). *arXiv*. DOI: 10.48550/arxiv.2306.15857.
 52. **Fährmann, D., Boutros, F., Kubon, P. (2024).** Ubiquitous Multi-Occupant Detection in Smart Environments. *Neural Computing and Applications*, Vol. 36, pp. 2941–2960. DOI: 10.1007/s00521-023-09162-z.
 53. **Lentzas, A., Dalagdi, E., Vrakas, D. (2022).** Multilabel Classification Methods for Human Activity Recognition: A Comparison of Algorithms. *Sensors*, Vol. 22, No. 6, pp. 2353. DOI: 10.3390/s22062353.
 54. **Chen, H., Gouin-Vallerand, C., Bouchard, K. (2024).** Enhancing Human Activity Recognition in Smart Homes with Self-Supervised Learning and Self-Attention. *Sensors*, Vol. 24, No. 3, pp. 884. DOI: 10.3390/s24030884.
 55. **Islam, M.M., Iqbal, T. (2020).** HAMLET: A Hierarchical Multimodal Attention-Based Human Activity Recognition Algorithm. *arXiv*. DOI: 10.48550/arxiv.2008.01148.
 56. **Ou Ali, I.H., Agga, A., Ouassaid, M. (2024).** Predicting Short-Term Energy Usage in a Smart Home Using Hybrid Deep Learning Models. *Frontiers in Energy Research*, Vol. 12, pp. 1323357. DOI: 10.3389/fenrg.2024.1323357.
 57. **Park, S. (2023).** Machine Learning-Based Cost-Effective Smart Home Data Analysis and Forecasting for Energy Saving. *Buildings*, Vol. 13, No. 9, pp. 2397. DOI: 10.3390/buildings13092397.
 58. **Singh, N.K., Nagahara, M. (2024).** LightGBM-, SHAP-, and Correlation-Matrix-Heatmap-Based Approaches for Analyzing Household Energy Data: Towards Electricity Self-Sufficient Houses. *Energies*, Vol. 17, No. 17, pp. 4518. DOI: 10.3390/en17174518.
 59. **Tonkin, E.L., Holmes, M., Song, H. (2023).** A Multi-Sensor Dataset with Annotated Activities of Daily Living Recorded in a Residential Setting. *Scientific Data*, Vol. 10, pp. 162. DOI: 10.1038/s41597-023-02017-1.
 60. **Jelić, M., Pujić, D., Batić, M. (2020).** Correlation between IoT Sensor Measurements and Total Electricity Consumption in Smart Homes.
 61. **Dewangan, F., Abdelaziz, A.Y., Biswal, M. (2023).** Load Forecasting Models in Smart

- Grid Using Smart Meter Information: A Review. *Energies*, Vol. 16, No. 3, pp. 1404. DOI: 10.3390/en16031404.
62. **Alwateer, M., Atlam, E.-S., Abd El-Raouf, M.M. (2024).** Missing Data Imputation: A Comprehensive Review, *Journal of Computer and Communications*, Vol. 12, No. 11, pp. 53–75. DOI: 10.4236/jcc.2024.1211004.
 63. **Ren, L., Wang, T., Sekhari Seklouli, A. (2023).** A Review on Missing Values for Main Challenges and Methods. *Information Systems*, Vol. 119, pp. 102–268. DOI: 10.1016/j.is.2023.102268.
 64. **Decorte, T., Mortier, S., Lembrechts, J.J. (2024).** Missing Value Imputation of Wireless Sensor Data for Environmental Monitoring. *Sensors*, Vol. 24, No. 8, pp. 2416. DOI: 10.3390/s24082416.
 65. **Chi, H., Chi, Y. (2022).** Smart Home Control and Management Based on Big Data Analysis. *Computational Intelligence and Neuroscience*, Vol. 2022, pp. 3784756. DOI: 10.1155/2022/3784756.
 66. **Wang, Y., Wang, L., Zheng, R. (2021).** Latency-Optimal Computational Offloading Strategy for Sensitive Tasks in Smart Homes. *Sensors*, Vol. 21, No. 7, pp. 2347. DOI: 10.3390/s21072347.
 67. **Iggena, T., Bin Ilyas, E.B., Fischer, M. (2021).** IoT-Crawler: Challenges and Solutions for Searching the Internet of Things. *Sensors*, Vol. 21, No. 5, pp. 1559. DOI: 10.3390/s21051559.
 68. **Padilla-Gomez, M., Gamboa-Cruzado, J., Távara-Aponte, S. (2024).** Systematic Literature Review on Cybersecurity and Its Influence on Cyber Attacks Targeting IoT Devices. *Computación y Sistemas*, Vol. 28, No. 4, pp. 1847–1864. DOI: 10.13053/cys-28-4-5006.
 69. **Andrade-Mogollon, T., Gamboa-Cruzado, J., Amayo-Gamboa, F. (2025).** Systematic Literature Review of Generative AI and IoT as Key Technologies for Precision Agriculture. *Computación y Sistemas*, Vol. 29, No. 2, pp. 857–882. DOI: 10.13053/cys-29-2-5738.
 70. **Mina-Rios, A., Albornoz-Yanac, S., Lucich-Muñoz, S., Gamboa-Cruzado, J., Salazar-Marzal, A., Chávez-Herrera, C., Del-Valle-Jurado, C., Pichardo-Lagunas, O. (2026).** IoT-Based Intelligent System for Monitoring and Predicting Urban Air Pollution: Spatial and Temporal Analysis in Seoul. *Computación y Sistemas*, Vol. 30, No. 1, pp. 839–864. DOI: 10.13053/cys-30-1-6346.

Article received on 19/06/2026; accepted on 10/09/2026.

*Corresponding author is Liliana Chanona-Hernández.