ALFA BK UNIVERSITY
FACULTY OF INFORMATIONTECHNOLOGY
FACULTY O F MATHEMATICS AND COMPUTER SCIENCE
International Scientific Conference

Alfa BK Univerzitet

UDK:711.45:004.78
004.912=93PYTHON
DOI:10.46793/ALFATECHproc25.137DJ
Original scientific paper

Smart Home Data Preprocessing Using Python

Ana ĐOKIĆ, Hana STEFANOVIĆ and Dragana DUDIĆ

Abstract: Smart home systems generate large volumes of data from various sensors, such as temperature, humidity, motion, and energy consumption. Effective preprocessing is essential to enhance data quality, reduce noise, and enable accurate analysis. This paper examines Python-based data preprocessing techniques for smart home environments. The preprocessing workflow includes data cleaning, handling missing values, normalization, feature selection, and data transformation. Data cleaning addresses duplicate records, outliers, and inconsistencies, while missing values are imputed using statistical and machine learning approaches. Normalization techniques standardize sensor readings to ensure consistency across data points. Feature engineering and dimensionality reduction refine the dataset for improved predictive modeling. By enhancing data quality, preprocessing contributes to smarter home automation, efficient anomaly detection, and optimized energy management. This study underscores the critical role of preprocessing in smart home analytics, facilitating reliable and meaningful insights for decision-making. The proposed techniques enhance the integration of smart home data into machine learning models, driving advancements in intelligent home automation systems.
Keywords: data preprocessing; Python programming language; smart homes.

1 INTRODUCTION

This paper provides a comprehensive overview of data preprocessing techniques employed in smart home applications, focusing specifically on the utilization of the Python programming language Additionally, it demonstrates these techniques through a use case on a selected smart home dataset.

Smart homes generate vast quantities of heterogeneous data from various sensors and devices (Alrefaei, 2024; Rizki, 2020). This raw data, however, is often noisy, incomplete, and inconsistent making direct analysis and modeling challenging (Bilal, 2022). Effective data preprocessing is therefore a crucial step in transforming raw sensor readings into structured, high-quality datasets suitable for machine learning, data mining or other analytical tasks (Tayeb, 2020).

In recent years, there has been a significant expansion of AutoML tools capable of automating all or some steps of data preprocessing. Notable examples include H2O-AutoML, Auto-sklearn, AutoWEKA, TPOT, AutoGluon, and DataRobot. However, no AutoML technique is yet mature enough to completely handle data preprocessing with complete accuracy (Bilal, 2022; Salehin, 2024). To maintain flexibility, enable advanced customization, and retain full control over data transformations, we opted for manual data preprocessing using Python programming language.

Python was chosen for this study due to its widespread use in data exploration and analysis. As a versatile, general-purpose, interpreted, and open-source language with dynamic semantics, Python supports rapid application development and is widely employed in data science applications, including data manipulation, automation, business analytics, and big data research (Lasser, 2021; Raschka, 2020).

In this paper, we will examine common preprocessing steps, relevant Python libraries like numpy, pandas, matplotlib and seaborn, and best practices for handling specific data challenges encountered in smart home environments.

The increasing complexity of smart home systems, with their interconnected devices and diverse data streams, necessitates a robust and adaptable preprocessing pipeline to ensure the accuracy and reliability of subsequent analyses (Essafi, 2024).

By providing a structured approach to preprocessing smart home data, this paper aims to serve as a valuable resource for researchers, engineers, and developers working in this domain. Through practical insights and real-world applications, we emphasize the importance of high-quality data preparation in unlocking the full potential of smart home analytics.

2 DATA ACQUISITION

Smart homes leverage a diverse array of sensors to capture a rich tapestry of data reflecting the activities and conditions within the home environment. These sensors encompass a wide range of modalities, including motion sensors detecting movement within specific areas (Nef, 2015; Babangida, 2022), temperature sensors monitoring ambient conditions (Kaur, 2024; Pirzada, 2018), light sensors measuring illumination levels (Pirzada, 2018), door/window sensors tracking entry and exit points (Babangida, 2022), and smart appliance usage data recording energy consumption and operational status (Maguluri, 2024; Mpawenimana, 2020; Umamageswari , 2024). The heterogeneity of these sensor types contributes to the complexity of smart home data, demanding tailored preprocessing strategies for each data modality.

The methods of data acquisition vary considerably depending on the specific sensor and its intended application. Some systems employ continuous data streams, constantly monitoring and recording sensor readings (Pirzada, 2018). This approach provides a detailed picture of the home's dynamic state but results in large volumes of data that require efficient processing and storage. Other systems may utilize event-based logging, only recording data when a significant event occurs, such as a door opening or a motion detection (Tax, 2019). This method reduces data volume but may miss subtle changes or patterns. The choice between continuous and event-based data acquisition involves a trade-off between data richness and computational efficiency.

The data collected in smart homes encompasses a wide spectrum of information relevant to various applications. Energy consumption data, typically measured by smart meters, is crucial for energy management and forecasting (Maguluri, 2024; Mpawenimana, 2020; Umamageswari, 2024). This data reveals patterns of energy use, enabling optimization strategies to reduce costs and improve efficiency. Activity patterns, inferred from sensor data, provide insights into residents' daily routines and habits (Babangida, 2022), (Tax, 2019; Mishra, 2020). This information is valuable for applications such as personalized healthcare recommendations, activity-based lighting control, and security systems. Moreover, smart home data can even serve as a proxy for health indicators (Hunter, 2020; Mishra, 2020), providing unobtrusive monitoring of an individual's well-being. For instance, changes in activity patterns or bathroom usage might signal a decline in health status. The versatility of smart home data makes it a powerful resource for a wide array of applications, but this diversity also necessitates sophisticated preprocessing techniques to extract meaningful insights from the raw data.

3 DATA PREPROCESSING

The raw data collected from smart homes is rarely suitable for direct analysis. It often suffers from various imperfections that necessitate careful preprocessing to ensure accurate and reliable results. This section delves into the common preprocessing techniques employed to address these issues.

3.1 Data Cleaning

Missing values and outliers are common occurrences in smart home data (Mpawenimana, 2020; Naware, 2024). Missing values arise from various sources, including sensor malfunctions, communication errors, or simply the absence of specific events. Outliers represent data points that significantly deviate from the expected pattern, often due to noise, errors, or unusual events. The presence of missing values and outliers can significantly distort the results of data analysis and lead to inaccurate conclusions.

Several methods exist for handling missing values. Simple imputation techniques include mean imputation, where missing values are replaced with the mean of the available data, and median imputation, where missing values are replaced with the median. However, these methods can distort the data distribution if a substantial portion of the data is missing. More sophisticated methods include k-nearest neighbors imputation, which uses the values of similar data points to estimate missing values, and model-based imputation, which uses a statistical model to predict missing values (Mpawenimana, 2020). The choice of imputation method depends on the characteristics of the data and the amount of missing data.

Outliers can be identified using various statistical methods. Box plots provide a visual representation of data distribution, highlighting data points that fall outside the interquartile range. The z-score, which measures the number of standard deviations a data point is from the mean, can also be used to identify outliers. Once identified, outliers can be handled through removal, transformation (e.g., using logarithmic transformation to reduce the impact of extreme values), or imputation (Naware, 2024). The decision on how to handle outliers depends on the context and the potential impact of the outliers on the analysis. Removing outliers might lead to loss of information, while retaining them could bias the results. Careful consideration is necessary to choose the most appropriate approach.

3.2 Data Transformation

Data transformation is essential for ensuring that features are on a comparable scale. Features with different scales can disproportionately influence machine learning models, leading to inaccurate results. Two commonly used transformation techniques are normalization and standardization (Baydomu, 2021).

Normalization scales features to a specific range, typically between 0 and 1. This is achieved by subtracting the minimum value of the feature and dividing by the range (maximum minus minimum). Normalization is particularly useful when the features have different ranges and the scale is important for the model's interpretation.

Standardization, also known as z-score normalization, transforms features by subtracting the mean and dividing by the standard deviation. This results in features with a mean of 0 and a standard deviation of 1. Standardization is preferred when the data is not uniformly distributed or when the range of the features is not meaningful. It is especially useful for machine learning algorithms sensitive to feature scaling, such as support vector machines (SVMs) (Chang, 2022; Mittelsdorf, 2018; Nef, 2015) and neural networks (Mpawenimana, 2020; Yuan, 2024; Umamageswari, 2024).

The choice between normalization and standardization depends on the specific dataset and the machine learning algorithm used. In some cases, both techniques may be applied to different features or at different stages of the preprocessing pipeline.

3.3 Feature Engineering and Selection

Feature engineering and selection are essential steps in preparing smart home data for analysis. Feature engineering involves generating new features from existing ones to enhance model performance (Yuan, 2024). This process may include combining sensor data to represent higher-level concepts, creating temporal features such as moving averages or time lags (Naware, 2024), and encoding categorical variables into numerical representations. For example, aggregating motion sensor data from multiple locations can produce a feature representing overall activity levels in the home. Likewise, calculating the moving average of energy consumption over a specific time window can help identify trends that may not be evident in the raw data.

Feature selection aims to identify the most relevant features for a given task, reducing dimensionality and enhancing model efficiency (Madhukar, 2024). High-dimensional data can lead to overfitting, where the model performs well on training data but poorly on unseen data. Feature selection techniques help mitigate this issue by selecting a subset of the most informative features for the prediction task.

Common feature selection methods include filter methods (e.g., correlation analysis, information gain), wrapper methods (e.g., recursive feature elimination), and embedded methods (e.g., L1 regularization). Additionally, principal component analysis (PCA) (Madhukar, 2024) is a widely used dimensionality reduction technique that transforms original features into a smaller set of uncorrelated principal components, capturing the most significant variance in the data.

The choice of feature engineering and selection techniques depends on the specific application and the characteristics of the dataset. It is essential to carefully balance model complexity and performance to achieve optimal results.

3.4 Data Smoothing and Noise Reduction

Smart home sensor data is often affected by noise, which can obscure underlying patterns and degrade the performance of machine learning models (Wei, 2025). Noise may arise from various sources, such as sensor inaccuracies, environmental interference, or communication errors. To address this, data smoothing techniques are employed to reduce noise while preserving essential features of the data.

Moving averages, where each data point is replaced by the average of its neighboring points, are a simple and effective smoothing technique (Yoon, 2024). The window size of the moving average determines the degree of smoothing. Larger window sizes result in smoother data but may also obscure fine-grained details. Median filters, which replace each data point with the median of its neighboring points, are more robust to outliers than moving averages. They are effective in reducing impulsive noise, which is characterized by sudden, sharp spikes in the data.

More advanced techniques, such as wavelet transforms and Kalman filtering, can be used for more sophisticated noise reduction (Wei, 2025). Wavelet transforms decompose the signal into different frequency components, allowing for selective removal of noise in specific frequency bands. Kalman filtering is a recursive algorithm that estimates the state of a dynamic system by combining noisy measurements with a model of the system's dynamics. The choice of smoothing and noise reduction technique depends on the characteristics of the noise and the desired level of smoothing.

4 USE CASE

4.1 The dataset

To demonstrate the data preparation process, we utilized the publicly available smart home dataset provided by Taranveer (2020), which includes weather information.

The dataset is available in CSV format and contains 32 columns with smart meter readings for home appliances in kW, recorded at one-minute intervals over a span of 350 days, along with weather information specific to that location. Original column names in the dataset are described in Table 1.

Table 1 Table Descriptions of column names from the dataset

The name of column	*Description*
time	Time
use [kW]	Total energy consumption
gen [kW]	Total energy generated by means of solar or other power generation resources
House overall [kW]	Overall house energy consumption
Dishwasher [kW]	Energy consumed by specific appliance
Furnace 1 [kW]	Energy consumed by specific appliance
Furnace 2 [kW]	Energy consumed by specific appliance
Home office [kW]	Energy consumed by specific appliance
Fridge [kW]	Energy consumed by specific appliance
Wine cellar [kW]	Energy consumed by specific appliance
Garage door [kW]	Energy consumed by specific appliance
Kitchen 12 [kW]	Energy consumption in kitchen 1
Kitchen 14 [kW]	Energy consumption in kitchen 2
Kitchen 38 [kW]	Energy consumption in kitchen 3
Barn [kW]	Energy consumed by specific appliance
Well [kW]	Energy consumed by specific appliance
Microwave [kW]	Energy consumed by specific appliance
Living room [kW]	Energy consumption in Living room
Solar [kW]	Solar power generation
temperature	Temperature
icon	Overall weather condition (clear-night:39%; clear-day:36%; Other:25%)
humidity	Humidity
visibility	Visibility
summary	Summarise weather (Clear:75%; Partly Cloudy:12%; Other:13%)
apparentTemperature	Apparent temperature
pressure	Pressure
windspeed	Wind speed
cloudCover	Cloud cover (0 :14%; 0.31 :10%; Other :77%)
windBearing	Wind bearing
precipIntensity	Precipitation Intensity
dewpoint	Dew point
precipProbability	Precipitation probability

4.2 Data cleaning and feature selection

For preliminary results, we conducted data cleaning and feature selection. A program was written to load the dataset, display its shape, show the first five rows, print the columns and their respective data types, check for entries with null values, drop rows containing missing values and checks the unique values.

We used the following Python libraries to clean the obtained dataset: NumPy, Pandas, Matplotlib, Seaborn, and Time.

We checked the data types and resolved any inconsistencies. Although 'cloudCover' is one of the unique entries, it appears that it should be a float. The 'cloudCover' entry appears in the first 57 minutes of the first hour, which seems to be an error. These values have been replaced with the first valid entry, assuming that cloud cover does not change dramatically in the first hour. Additionally, we observed that the dataset includes columns related to energy generation, energy consumption per room/appliance, weather data, and a 'time' column.

From the correlation matrix of the dataset, it is evident that three pairs of variables are highly correlated: 'use [kW]' and 'House overall [kW]', 'gen [kW]' and 'Solar [kW]', and 'Temperature' and 'apparentTemperature'. To facilitate analysis, the variables 'House overall [kW]', 'Solar [kW]', and 'apparentTemperature' were dropped. Since the 'time' column is now the index of the data frame, it has also been removed. The object columns 'icon' and 'summary' were deemed irrelevant for further analysis and were discarded. Additionally, two variables related to the 'Furnace' and three variables related to the 'kitchen' were combined by summing the corresponding variables into new columns, and the individual columns were dropped. Finally, the column names were renamed to remove spaces and the '[kW]' unit.

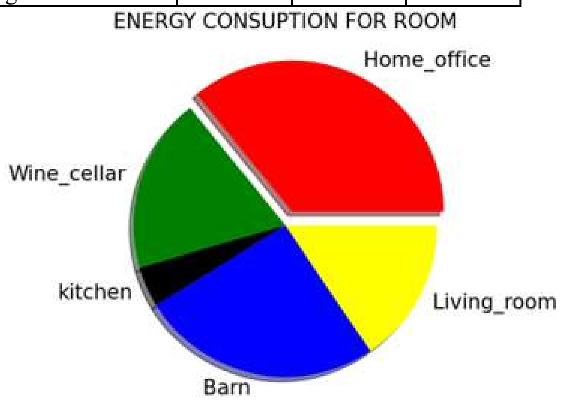
The final results of data cleaning are presented in Table 2.

Table 2 The dataset information of prepared data

Column	*Non-Null*	*Count*	*dtype*
use	466308	not-null	float64
gen	466308	not-null	float64
Dishwasher	466308	not-null	float64
Furnace	466308	not-null	float64
Home_office	466308	not-null	float64
Fridge	466308	not-null	float64
Wine_cellar	466308	not-null	float64
Garage_door	466308	not-null	float64
Kitchen	466308	not-null	float64
Barn	466308	not-null	float64
Well	466308	not-null	float64
Microwave	466308	not-null	float64
Living_room	466308	not-null	float64
temperature	466308	not-null	float64
humidity	466308	not-null	float64
visibility	466308	not-null	float64
pressure	466308	not-null	float64
windspeed	466308	not-null	float64
cloudCover	466308	not-null	float64
windBearing	466308	not-null	float64
precipIntensity	466308	not-null	float64
dewpoint	466308	not-null	float64
precipProbability	466308	not-null	float64

4.3 Data visualisation

To effectively present the data, data visualization is both essential and highly beneficial. A program was developed to generate a pie chart illustrating the data in columns containing variables related to energy consumption within the premises and for specific devices, as shown in Figure 1.



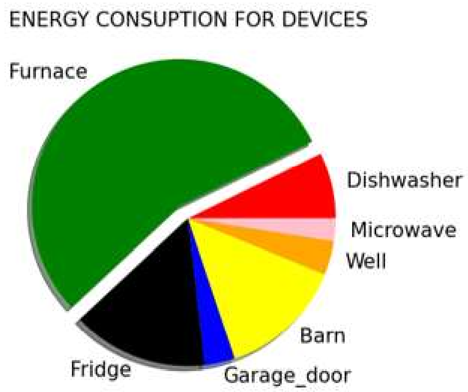


Figure 1 Energy consumption in the rooms and for devices

A line graph illustrating the dependency of various energy consumption per month throughout the year is presented in Figure 2, with months on x axis, and a numerical variable Electricity Consumption [kW] on y axis.

4.3 Discussion

Considering only the variables related to energy consumption in different rooms, it can be concluded that energy consumption is highest in the home office and lowest in the kitchen for the given dataset. Approximately the same energy consumption is recorded in the wine cellar and the living room.

Similarly, when analyzing the variables related to energy consumption for specific devices, it is evident that the furnace consumes the most energy, while the microwave and garage door consume the least. The fridge and barn exhibit approximately the same energy consumption. The high energy consumption of the furnace is expected, as it generates heat using electric resistance coils, which is then distributed throughout the house. This heating system is known for operating safely, cleanly, and without producing hazardous byproducts.

Regarding monthly energy consumption, the results indicate that the highest values are recorded in February, July, and November. These findings align with expectations, given the increased energy usage during winter months for heating and during summer months for cooling. Additionally, solar panel energy production is significantly reduced during the winter months. For devices such as the microwave, fridge, and garage door, no significant variations in energy consumption are observed throughout the year.

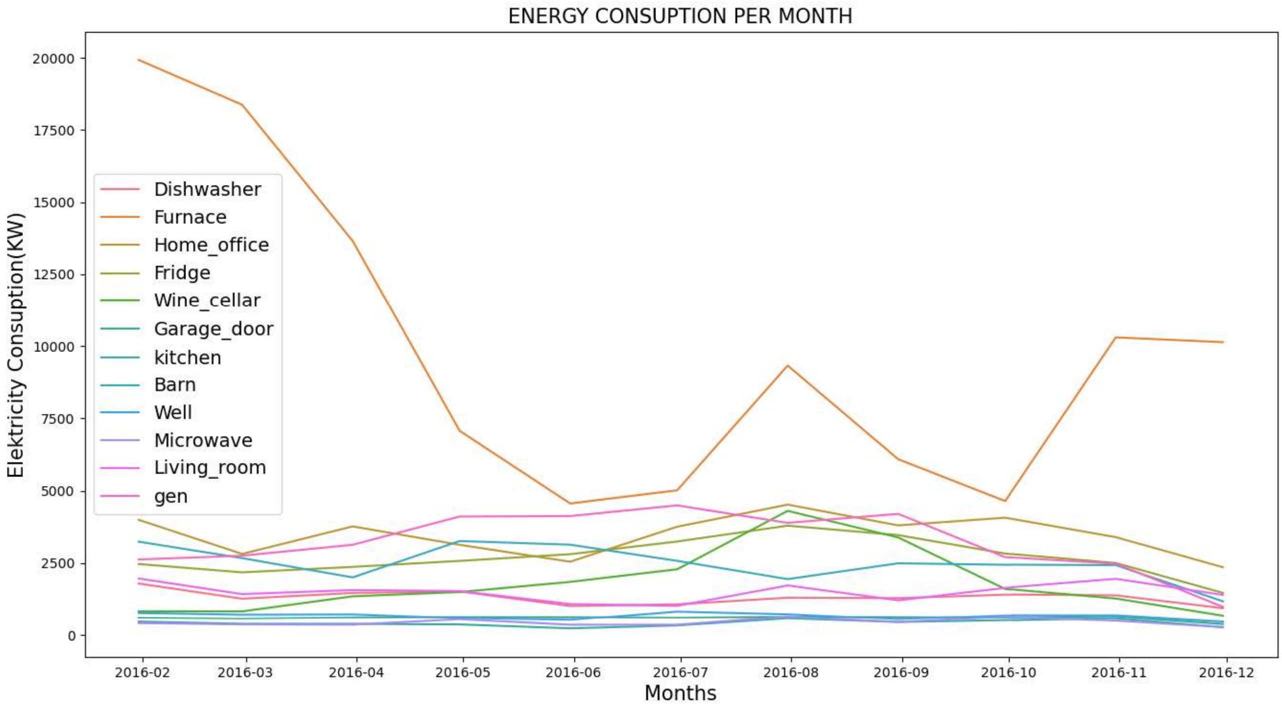


Figure 2 Energy consumption per month throughout the year

5 CONCLUSION

Data preprocessing is a crucial step in smart home data analysis, as it directly impacts the accuracy, reliability, and overall quality of insights derived from the data. The techniques employed must be carefully selected and rigorously evaluated to ensure that smart home data is properly cleaned, transformed, and structured for meaningful analysis. The choice of preprocessing methods depends on various factors, including the specific application, the nature of the data, and privacy considerations. Given the diverse and complex nature of smart home data, preprocessing remains a challenging yet indispensable task.

Despite advancements in data preprocessing techniques, several challenges persist, particularly in handling high-dimensional, heterogeneous, and rapidly generated smart home data. The variability in sensor readings, missing or inconsistent values, and privacy concerns further complicate the preprocessing pipeline. Addressing these challenges requires a strategic approach that balances computational efficiency with data integrity.

Future research should initially focus on evaluating the obtained results and completing a comprehensive preprocessing workflow to ensure high-quality data preparation. Additionally, further efforts should be directed toward developing more sophisticated, efficient, and privacy-preserving preprocessing techniques. These advancements will not only enhance data quality but also improve the effectiveness of machine learning models and analytical frameworks used in smart home applications.

Moreover, the establishment of standardized preprocessing pipelines and evaluation frameworks would significantly advance the field. Standardization would promote reproducibility, facilitate comparability across studies, and support the broader adoption of best practices in smart home data analysis. Ultimately, by refining data preprocessing methods and addressing existing challenges, researchers and developers can unlock the full potential of smart home technologies, leading to smarter, more efficient, and privacy-aware intelligent home automation systems.

6 REFERENCES

[1] Alrefaei, A. & Ilyas, M. (2024). Using machine learning multiclass classification technique to detect IoT attacks in real time. Italian National Conference on Sensors. https://doi.org/10.3390/s24144516

[2] Babangida, L., Perumal, T., Mustapha, N., & Yaakob, R. (2022). Internet of things (iot) based activity recognition strategies in smart homes: a review. IEEE Sensors Journal. https://doi.org/10.1109/JSEN.2022.3161797

[3] Baydomu, G. K. (2021). The effects of normalization and standardization an internet of things attack detection. European Journal of Science and Technology. https://doi.org/10.31590/ejosat.1017427

[4] Bilal, M., Ali, G., Iqbal, M. W., Anwar, M., Malik, M. S. A., & Kadir, R. A. (2022). Auto-prep: efficient and automated data preprocessing pipeline. IEEE Access. https://doi.org/10.1109/ACCESS.2022.3198662

[5] Chang, J., Kang, M., & Park, D. (2022). Low-power on-chip implementation of enhanced svm algorithm for sensors fusion-

based activity classification in lightweighted edge devices. Multidisciplinary Digital Publishing Institute. https://doi.org/10.3390/electronics11010139

[6] Essafi, K., & Moussaid, L. (2024). The potential of the internet of things for human activity recognition in smart home: overview, challenges, approaches. *Indonesian Journal of Electrical Engineering and Computer Science.* http://doi.org/10.11591/ijeecs.v36.i1.pp302-317

[7] Hunter, I., Elers, P., Lockhart, C., Guesgen, H. W., Singh, A., & Whiddett, D. (2020). Issues associated with the management and governance of sensor data and information to assist aging in place: focus group study with health care professionals. JMIR Publications. https://doi.org/10.2196/24157

[8] Kaur, J., Oetomo, A., Chauhan, V., & Morita, P. (2024). 0291 evaluating sleep quality metrics using zero-effort technology: implications for public health dynamics. Sleep. https://doi.org/10.1093/sleep/zsae067.0291

[9] Lasser J, Manik D, Silbersdorff A, Säfken B, Kneib T. Introductory data science across disciplines, using Python, case studies, and industry consulting projects. Teaching Statistics. 2021; 43: S190–S200. https://doi.org/10.1111/test.12243

[10] Madhukar, S. R., Singh, K., Kanniyappan, S., Krishnan, T., Sarode, G. C., & Suganthi, D. (2024). Towards efficient energy management of smart buildings: a lstm-ae based model. None. https://doi.org/10.1109/ICECCC61767.2024.10593988

[11] Maguluri, L., Shankar, M., Aruna, R., Devi, D. C., & Suganya, M. J. (2024). A comprehensive evaluation of machine learning algorithms for precise energy consumption forecasting in smart homes. None. https://doi.org/10.11591/ijpeds.v15.i4.pp2138-2144

[12] Mishra, P., Busetty, S. M., & Gudla, S. K. (2020). Enhanced activity recognition of the iot smart home users through cluster analysis. World Forum on Internet of Things. https://doi.org/10.1109/WF-IoT48130.2020.9221177

[13] Mittelsdorf, M., Hwel, A., Klingenberg, T., & Sonnenschein, M. (2018). Submeter based training of multi-class support vector machines for appliance recognition in home electricity consumption data. International Conference on Smart Grids and Green IT Systems. https://doi.org/10.5220/0004380001510158

[14] Mpawenimana, I., Pegatoquet, A., Roy, V., Rodriguez, L., & Belleudy, C. (2020). A comparative study of lstm and arima for energy load prediction with enhanced data preprocessing. Sensors Applications Symposium. https://doi.org/10.1109/SAS48726.2020.9220021

[15] Nef, T., Urwyler, P., Bchler, M., Tarnanas, I., Stucki, R., Cazzoli, D., Mri, R. M., & Mosimann, U. P. (2015). Evaluation of three state-of-the-art classifiers for recognition of activities of daily living from smart home ambient data. Multidisciplinary Digital Publishing Institute. https://doi.org/10.3390/s150511725

[16] Pirzada, P., White, N., & Wilde, A. (2018). Sensors in smart homes for independent living of the elderly. None. https://doi.org/10.1109/imtic.2018.8467234

[17] Raschka, S., Patterson, J., & Nolet, C. (2020). Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence. Information, 11(4), 193. https://doi.org/10.3390/info11040193

[18] Rizki, R. P., Hamidi, E. A. Z., Kamelia, L., & Sururie, R. (2020). Image processing technique for smart home security based on the principal component analysis (pca) methods. None. https://doi.org/10.1109/ICWT50448.2020.9243667

[19] Salehin, I., Islam, M. S., Saha, P., Noman, S., Tuni, A., Hasan, M. M., & Baten, M. A. (2023). AutoML: A systematic review on automated machine learning with neural architecture search. Journal of Information and Intelligence, 2(1), 52–81. https://doi.org/10.1016/j.jiixd.2023.10.002

[20] Tax, N. (2019). Mining insights from weakly-structured event data. Cornell University. https://doi.org/10.48550/arxiv.1909.01421

[21] Tayeb, H. F., Karabatak, M., & Varol, C. (2020). Time series database preprocessing for data mining using python. International Symposium on Digital Forensics and Security. https://doi.org/10.1109/ISDFS49300.2020.9116260

[22] Umamageswari S, & Kannan M. (2024). Smart home automation system for energy consumption using tensorflow-based deep ensemble learning. ICTACT Journal on Soft Computing. https://doi.org/10.21917/ijsc.2024.0461

[23] Wei, M., Zhao, D., Zhang, L., Wang, C., Zhang, Y., Wang, Q., Fan, X., Zhong, Y., & Mao, S. (2025). Wi-fitness: improving wi-fi sensing with video perception for smart fitness. IEEE Internet of Things Journal. https://doi.org/10.1109/JIOT.2024.3476291

[24] Yoon, N., Lee, S., Kim, S. K., Park, C., Kim, T., & Jin, H. (2024). Energy consumption prediction using cnn-lstm models: a time series big data analysis of electricity, heating, hot water, and water. None. https://doi.org/10.1109/ICCE-Asia63397.2024.10774020

[25] Yuan, Q. (2024). Development and verification of a machine learning based motor energy consumption prediction algorithm. None. https://doi.org/10.1109/PEEEC63877.2024.00028

Contact information:

Ana ĐOKIĆ 2, MSc, Lecturer
Information Technology School
Cara Dušana 34, Zemun, Serbia
ana.djokic@its.edu.rs
https://orcid.org/0009-0002-3469-586X

Hana STEFANOVIĆ 1, PhD, Senior Lecturer
Academy of Technical and Art Applied Studies
School of Electrical and Computer Engineering
Vojvode Stepe 283, , Belgrade, Serbia
hanas@viser.edu.rs
https://orcid.org/0000-0003-0890-4410

Dragana DUDIĆ 3, PhD, Assistant Professor
(Corresponding author)
Faculty of Computer Science and Informatics, University Union Nikola Tesla
Cara Dušana 62-64, Belgrade, Serbia
ddudic@unionnikolatesla.edu.rs
https://orcid.org/0000-0001-8513-6529