JJEE Volume X | Number X | Month 20XX | Pages XXX-XXX

Jordan Journal of Electrical Engineering

ISSN (print): 2409-9600, ISSN (online): 2409-9619

Homepage: jjee.ttu.edu.jo

AI-Based Energy Management and Prediction System for Smart Cities

Ahmed Saeed[1], Rao Muhammad Asif[2], Ateeq Ur Rehman[3*], Syed Rizwan Hassan[4], Salil Bharany[5], Habib Hamam[6, 7, 8, 9]

[1, 2] Department of Electrical Engineering, Superior University, Lahore, Pakistan
[3]School of Computing, Gachon University, Seongnam-si, Republic of Korea
E-mail: 202411144@gachon.ac.kr
[4]School of Computing Sciences, NFC Institute of Engineering and Fertilizer Research, Faisalabad, Pakistan
[5]Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India
[6]Faculty of Engineering, University of Moncton, Moncton, Canada
[7]School of Electrical Engineering, University of Johannesburg, Johannesburg, South Africa
[8] International Institute of Technology and Management (IITG), Av. Grandes Ecoles, Libreville BP 1989, Gabon
[9]Bridges for Academic Excellence - Spectrum, Tunis, Centre-Ville, Tunisia

Received: Oct 05, 2024 *Revised: Dec 19, 2024* *Accepted: Dec 29, 2024* *Available online: Feb 18, 2025*

***Abstract*—** In recent years, there has been an increasing interest in smart city concepts in response to a growing demand for more sustainable and efficient urban spaces. In this paper, an in-depth exploration of load forecasting in an innovative city framework is carried out through the building of a software application with a graphical user interface (GUI) and prediction using long short-term memory (LSTM). To fulfill their role in enabling intelligent cities, accurate load forecasting is needed to enable energy management, reduce peak demand, and allow for well-informed decisions and energy distribution optimization. Specifically, in this work, we consider the application of LSTM to load forecasting systems within the context of a smart city. Since LSTM has a high capability in capturing complicated temporal structures, it is indeed an efficient approach to load forecasting, especially considering long-term dependencies. We focus on the strength of LSTM in comparison with the traditional statistical analysis and other machine learning techniques, most notably, LSTM's ability to handle nonlinear and dynamic load behavior typical in innovative city energy systems. The GUI is the system's front end where the user enters the data, and at this point, we have the city officials, energy managers, and the community people and get their customized load forecast. The discussion of the outcomes of the experiment with the cross-platform GUI application and the improved LSTM forecasting model is given. Based on the evaluation of the model in terms of accuracy and performance, the research is done in an accepted manner to predict the complex load pattern in a smart city environment by proving the efficiency of the LSTM model. The effectiveness of different decisions in various cases is evaluated to consider the influence on decision-making processes and energy optimization; various cases demonstrate how the proposed GUI application is helpful in facilitating better management of energy. This research will, therefore, endeavor to create knowledge in the area of energy management to foster the development of effective intelligent cities.

***Keywords*—** Smart cities; Load forecasting; Long short-term memory; Energy management; GUI.

1. INTRODUCTION

Several advancements in renewable energy have been seen in the last few years. Information and communication technology (ICT) combined with artificial intelligence (AI) has enhanced the effectiveness of green energy by introducing advanced approaches for accurate demand-side management at each site [1-3]. To exploit these predictions for efficient energy management, this research addresses the fundamental concept of load forecasting utilizing deep learning algorithms for smart city environments [4].

* Corresponding author
DOI: https://doi.org/ 10.5455/jjee.204-1728140025

The idea of a "smart city" has become more popular in recent years due to the growth of the Internet and now impacts both big and small cities, see Fig. 1. The adjective "smart," which refers to a digital, intelligent, and sustainable city, is key to understanding the concept of smart cities.

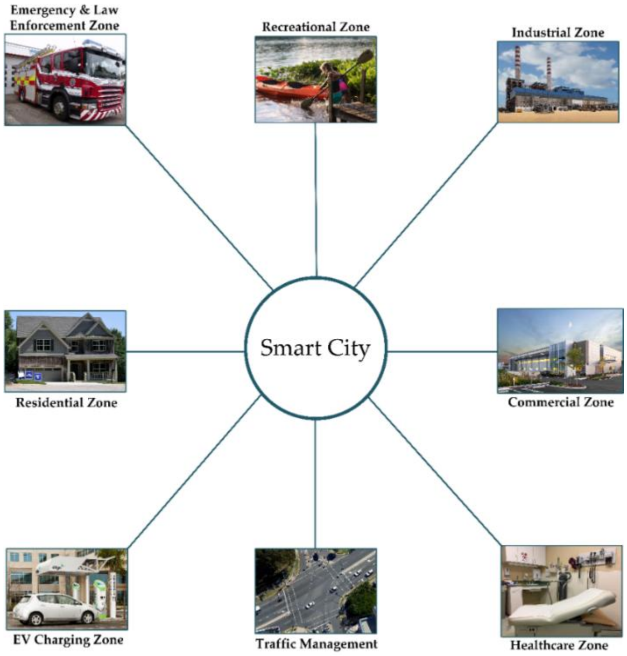

Fig. 1. Smart city zones.

Recently, AI has been considered a potent instrument for the development of smart cities [5]. Applications powered by AI are still being developed, and as such, their full potential has not yet been realized. However, the introduction of AI has already revealed a potential double-edged sword, where negative impacts can go unnoticed due to a propensity toward technology. A "smart city" is a technologically enhanced urban environment that uses a variety of electrical devices or sensors to collect specific data. This information is utilized to enhance daily operations for effectively managing the city's resources, services, and assets [6], see Fig. 2. Data are collected from people, devices, structures, and assets to monitor and regulate transportation and traffic systems, hospitals, rescue stations, charging stations for electric vehicles, and parks. Cities are referred to as "smart cities" [7] if they successfully employ technology in their organizing, tracking, evaluation, and management. In order to interact with residents and enhance operations and services, a concept known as a "smart city" combines ICT with a variety of physical devices connected with the Internet of Things (IoT) network [8, 9]. Given the broad range of technologies involved, providing a specific definition of a "smart city" is challenging [10].

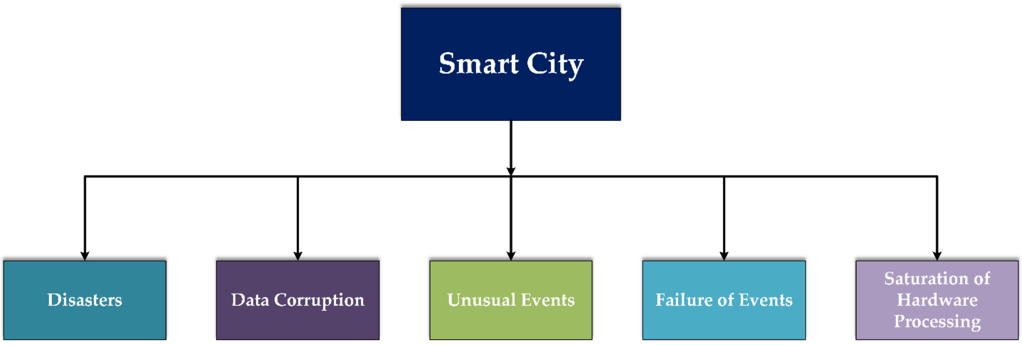

Fig. 2. Factors affecting smart cities.

Short term load forecasting (STLF) is important for the effective operation of electrical devices to be used as a part in energy management systems. Also, the importance of STLF is apparent because it can affect electric equipment immediately if it is incorrect. Time stands out as the key element in STLF, with daily trends being especially significant. Furthermore, external factors include weather, including variations in temperature and humidity as well as shifts in the demand for electricity during holidays further contribute to the complexity and accuracy of STLF predictions [11-17]. Effective short-term forecasting can be achieved by combining time-series data with other techniques. These methods include time-series models, statistical regression models, and deep learning-based models. Along with the previously mentioned factors, the dimensions of the residence, the age of appliances and machinery, and international problems such as epidemics can all affect the load estimate for short- and long-term predictions. Despite a few slight changes, the majority of techniques have the same features. Time series can be used to visualize datasets regarding load usage. Recurrent neural networks (RNNs), a type of artificial neural network, are made to process successive inputs, including those used for translating languages or load estimation. RNNs' ability to maintain a mental state (memory) and their continuous connections to identical neurons from the previous time step makes them particularly well suited for mimicking temporal behavior. Backpropagation across time is commonly used to train RNNs; however, this might result in the issue of vanishing gradients, which makes the neural network neglect older input, particularly over long periods. This issue has been addressed using long short-term memory (LSTM) networks due to their enhanced information storage and prediction accuracy. Table 1 summarizes the existing studies.

Several problems have been addressed in the literature:

- Inadequate Handling of Nonlinear and Dynamic Load Patterns: One common issue is the inadequate handling of nonlinear and dynamic load patterns.
- Limited Integration of Real-Time and Diverse Data Sources: Real-time data streams and a variety of data sources, including weather, economic indicators, and data from Internet of Things sensors, are not fully integrated in many of the load forecasting studies that are currently available. To give precise and timely load estimates, models that can integrate and handle real-time data from many sources must be developed.

  Lack of User-Friendly Tools for Stakeholders: Typically, the implementation of advanced load forecasting models does not receive any ground because there are no user-friendly tools or interfaces that can make these models available to non-expert stakeholders. This gap clearly shows the necessity of creating GUI applications that could ease the process of introducing data into models as well as understanding the results obtained by those models.
- Insufficient Focus on Demand-Side Management and Optimization: Many investigations have concentrated on the technical aspects of load prediction, yet few have examined effective strategies and optimal algorithms at the demand side that are able to use forecasted loads for minimizing energy consumption costs despite their levels.

The main contributions of this article may be summarized as follows:

- Using LSTM to forecast load: The research conducted proved that LSTM neural networks are effective in accurately forecasting energy loads within the smart city context, addressing the complexity of nonlinear and dynamic load patterns.

Table 1. Comparison of the existing studies.

Ref.	Key contribution	Technique	Limitations
[18]	Using Deep Learning to Power a Smart City Energy Management Decision Support System	Convolutional Neural Network	There should be many training data but avoid encoding the object orientation and position
[19]	Utilizing Artificial Neural Networks and Reinforcement Learning for Demand Response in the Management of Home Energy	Artificial Neural Network and Multi-Agent Reinforcement Learning	Not Reduced the irrelevant data
[20]	Smart Monitoring, Artificial Intelligence, and Energy Consumption Optimization for Energy Sustainability in Smart Cities	Support Vector Machine	Computational time is very high
[21]	Data-driven air conditioner load predictions utilizing an ANN based on the Levenberg–Marquardt algorithm for demand response	(LMA)-based Artificial Neural Network (ANN)	Quite complicated because no strategy was utilized to remove unnecessary data
[22]	Smart Grid Price-Based Demand Response Is Proposed for Economical Energy Management of IoT-Enabled Smart Homes	Intelligent Forecaster and IoT	Increased privacy concerns
[23]	The benefits of deep learning and its uses on the Internet of Things	Deep Learning driven IoT	The incoming data can be used by IoT to learn hierarchical representations.
[24]	Using a hybrid PCA and ARIMA algorithm, electrical consumption for Internet of Things smart houses are predicted.	PCA and ARIMA using the IoT	IoT Highly Dependent on Internet
[25]	Machine learning-based short- and long-term electric load forecasting	Machine Learning	Complexity of the used models' computations
[26]	Understanding error calculation techniques in the context of energy forecasting	A Novel Technique Related to Errors	Underestimating the situation and failing to make sufficient preparations using multiple dimensions to forecast performance
[27]	Smart meter data and deep learning for load forecasting: Online Adaptive Recurrent Neural Network	Deep Learning	The buffering module's goal is to locate batches for which the model could not effectively perform and temporarily store them
[28]	Using data from smart meters, online adaptable recurrent neural networks can forecast loads	Adaptive Recurrent Neural Network	The buffering module's goal is to locate and temporarily hold batches where the model failed to deliver
[29]	Online adaptive recurrent neural networks can forecast loads using input from smart meters	Parallel Deep LSTM-CNN and ML Techniques	The findings may vary depending on the weather. The load was predicted using the previous consumption as a parameter
[30]	Accurate Probability Distribution and Load Forecasting of Peak Loads	Best Fit Models and Autoregression and Exponential Smoothing Models for Forecasting	Best Fit does not achieve good results as compared to the AR
[31]	Smart Grid Load Forecasting Using RNN and LSTM	RNN and LSTM	Simple RNN does not predict accurately in Long Term Dependencies
Proposed Model	AI-Based Energy Management and Prediction System for Smart Cities	LSTM	There is a need to create hybrid models that combine LSTM with other machine-learning techniques

- Development of a user-friendly GUI: created a GUI application that simplifies the process of load forecasting for city administrators, energy managers, and residents, making the model's output accessible and actionable for stakeholders.
- Weather data integration: weather data and historical load statistics are incorporated into the LSTM model, enhancing the model's capacity to identify seasonal patterns or improve forecasting precision.
- Evaluation of model performance: an extensive evaluation of LSTM model's effectiveness was performed through multiple metrics, and it is shown that LSTM model has significant superiority compared to conventional statistical techniques as well as other machine learning algorithms in determining the subtle load pattern.

The rest of the article is arranged as follows: the introduction section describes the context, significance, and research contributions related to smart cities. The data collection and methodology section details the use of RNNs and LSTMs, along with the data collection and model evaluation strategies. The last section outlines the system's development and implementation, including specific applications, reports findings and suggests future research directions.

2. DATA COLLECTION AND METHODOLOGY

Neural network techniques, specifically deep learning techniques, are the main emphasis of this study. This section discusses RNN, the most significant family of neural networks.

2.1. RNNs

Feed-forward neural networks are the usual approach used for networks with a directed graph devoid of cycles. Fig. 3 illustrates an unrolled diagram of an RNN, where A represents the repeating section of the neural network, and x_t is an input that produces the output h_t.

2.2. LSTM Neural Networks

Although simple RNNs can theoretically handle "long-term dependencies," it appears that in reality, they are unable to acquire them. Hochreiter and Schmidhuber suggested LSTM [32], a type of RNN that can memorize information for extended periods, as a solution to this issue.

LSTM differs from a standard RNN in its repeating module construction. The structure in Fig. 4 is designed to enhance the clarity and understanding of the architecture involved.

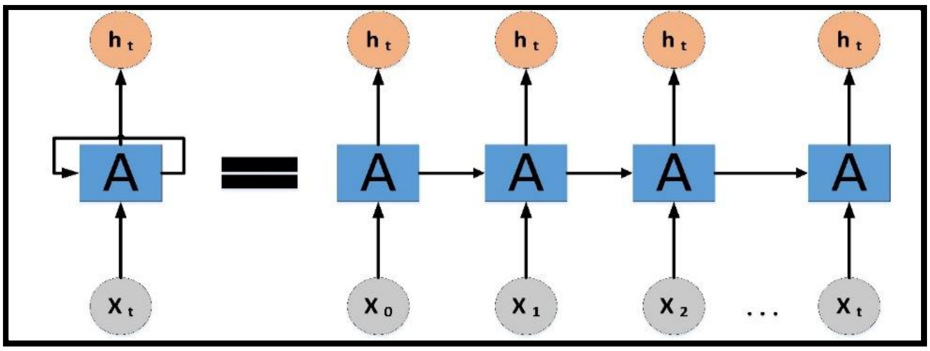

Fig. 3. Unrolled RNN [32].

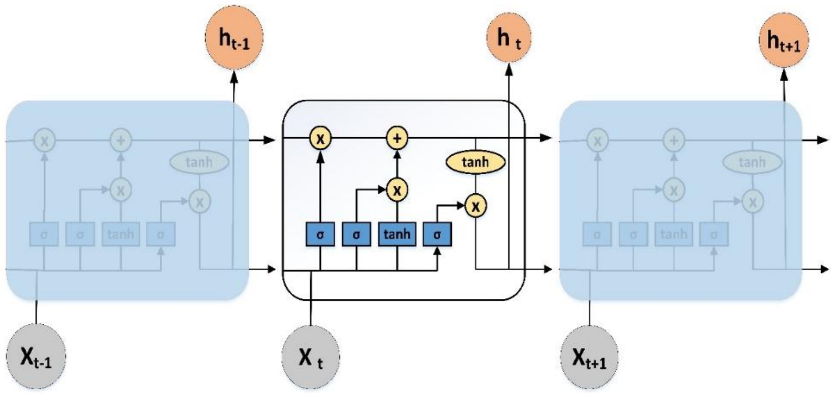

Fig. 4. LSTM layers [32].

The main distinctiveness of the LSTMs is the two recurrent connections in the repeating module. LSTMs use the output of the prior prediction in addition to the typical recurrent connections present in RNNs (shown in Fig. 4 by the top horizontal arrow). Each LSTM network element is referred to as a "cell". Each cell has two outputs and three inputs. In Fig. 4, x_t is input at time step t, h_{t-1} is previous hidden state, $C_{t-1}$is previous cell state, h_t is the update of the hidden state is used for the prediction of the output, C_t is current cell state.

LSTM makes use of a unique theory to manage the memorization process. LSTM gates, sometimes referred to as gating mechanisms, multiply analog memory elements pointwise with a sigmoid activation function and store the outputs in the 0 to 1 range to create probabilistic scores. Gates regulate the flow of information into and out of LSTM cells. Fig. 5 illustrates the layered architecture of LSTM.

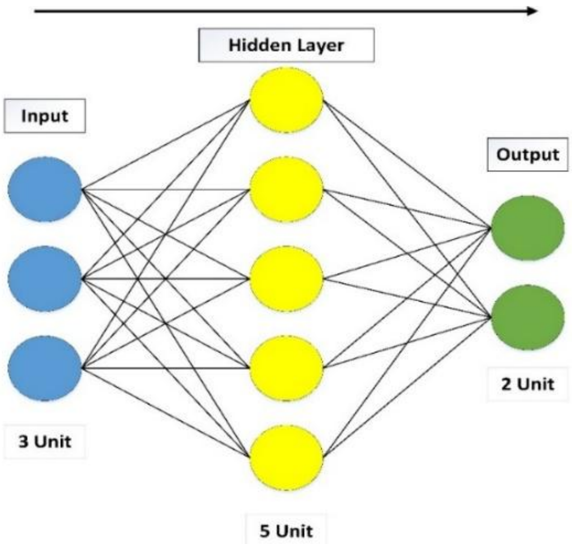

Fig. 5. Different layers of LSTM [11].

Four major steps involved in LSTMs are:

a) Forget gate**:** the data to be deleted from the cell state are chosen by the forget gate, the first part of an LSTM as shown in Fig. 6. Eq. 1 says that for every number, in the cell state C_{t-1}, it outputs a number between 0 (destroy it completely) and 1 using a sigmoid layer that mixes the input x_t with the hidden state h_{t-1}(keep it completely)

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{1}$$

Where f_t is forget gate activation time step, σ is sigmoid activation function, x_t is input at the current time step t, W_f is weight matrix for the forget gate, b_f is bias term for the forget gate and h_{t-1} is hidden state from the previous step (t-1)

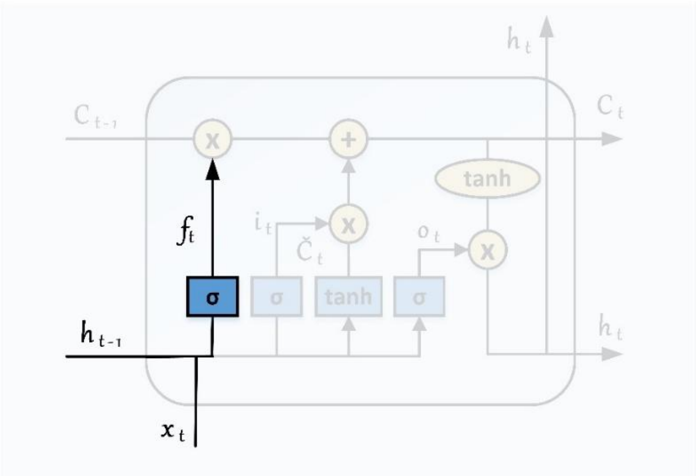

Fig. 6. First step of LSTM [32].

b) Input gate: Which data should be included in the cell state is decided by the input gate, which is represented by the second gate shown in Fig. 7. It is made up of two parts. First, a sigmoid layer uses h_{t-1} and x_t are to produce a vector i_t. This vector i_t contains values between 0 and 1, indicating the proportion of the cell state values that will be updated. Next, a tanh layer combines h_{t-1} and x_t to generate a vector of new candidate values $C_{t-1}^{\sim}$.

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \quad (2)$$

$$C^{\sim}{}_t = tanh(W_c.[h_{t-1}, x_t] + b_c) \quad (3)$$

where i_t is input gate activation at time step t, W_i is weight matrix associated with input gate, $b_i$is bias vector for the input gate, $C^{\sim}{}_t$is candidate cell step at time step t, $tanh$ is hyperbolic tangent activation function, W_c is weight matrix for the candidate cell and $b_c$is bias vector for candidate cell

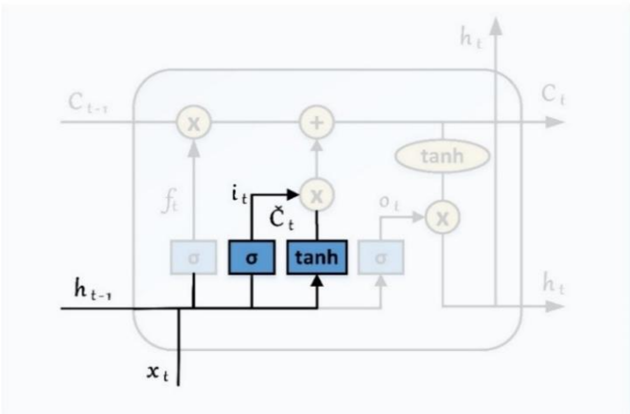

Fig. 7. Second step of LSTM [32].

c) Cell state: an essential element that allows a system with an LSTM system to store and update data across lengthy sequences is the cell state. It serves as a unit of memory, storing pertinent data and passing it on selectively to subsequent time steps so that the network can identify long-term dependencies. The cell state can be updated using the outcomes of the previous processes as shown in Fig. 8. The new cell state forgets the data chosen in the first step after multiplying the previous state by f_t. As a result, the new state is updated with the new information from step two. The modified values i_t and the new candidates' values $C_t^{\sim}$ are multiplied to obtain this new information. Eq. 4 contains all of the operations.

$$C_t = f_t * C_{t-1} + i_t * C_t^{\sim} \quad (4)$$

Where: C_t is current cell state at time step.

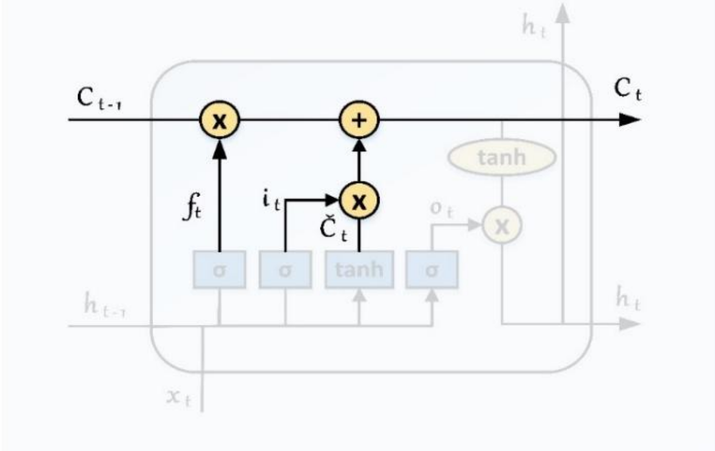

Fig. 8. Third step of LSTM [32].

d) Output gate: the output is generated as the final step, as depicted in Fig. 9, and it represents a refined description of the updated cell state. Initially, a sigmoid layer decides the relevance of elements from the cell state for output based on h_{t-1} and x_t as described in Eq. 5. The cell state values are then scaled between -1 and 1 by applying tanh to the sigmoid layer's output (Eq. 6).

$$O_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = O_t * tanh(C_t) \quad (6)$$

Where: O_t is output gate function at time step t, W_o is weight matrix for output gate, $b_o$is bias vector for output gate, h_t is hidden state at the time step and O_t is the output gate activation function.

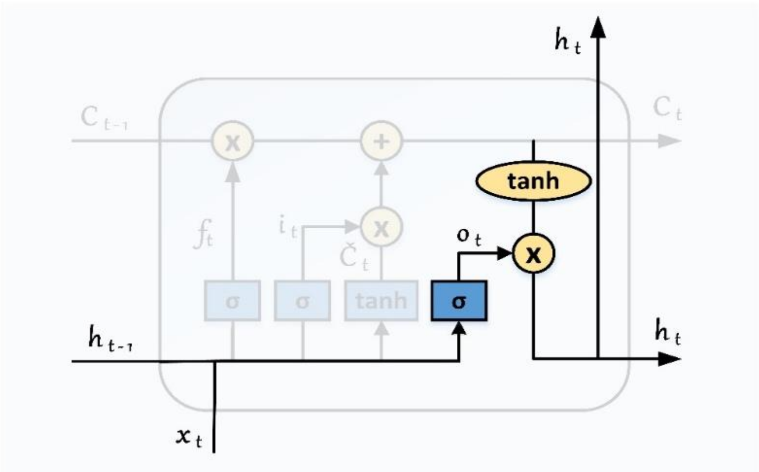

Fig. 9. Fourth step of LSTM [32].

2.3. Proposed Integration Strategy

In this step, the forecast is further optimized by lowering the RMSE. The integration approach does not consider twenty-year data containing four seasons, per-month data transformation, ANN, or choosing features as a final option; instead, it concentrates on the best method for the pertinent week of each season. Additionally, we modified the RMSE computations to consider the integration method that we created for each specific building. The quality of the input data and the chosen hyperparameters are two factors that affect LSTM networks, much like any other AI model, and can influence how effectively they function. In general, LSTM networks have been demonstrated to be an extremely potent and effective tool for a range of machine learning applications, particularly for natural language processing.

A LF artificial neural network was used to validate the algorithm, which is outlined in more detail in Table 2. Finding the inaccurate data points is the aim of this integration. A workable solution should be created after calculating the root mean square error for every set of data. The hardware should then receive more precise data. The simulation results show that an acceptable RMSE of less than one is achieved by the suggested five-stage LSTM model, which uses current data as an input for the error-correcting function. The memory cell acts as a long-term repository for relevant data, while the concealed state acts as a short-term memory that selectively retrieves information from it. Fig. 10 shows each phase of the suggested process.

Table 2. Sequence of simulations.

Step	Procedure
1	Set the hidden state and cell state to 0.
2	According to the input sequence, at each time step t: • Calculate signal to the gate using the present input vectors and the previous hidden state i_t. • Utilizing the prior hidden state and the current input vector, compute the forget gate f_t. • Calculate memory cell vector using the current input vector and the previous concealed state C_t. • Update memory cell vector utilizing the prospective memory cell vector C_t, the forget gate $f_t$and the input gate i_t. • Calculate output gate o_t utilizing the prior hidden state and the current input vector state. • Apply o_t to the vector C_t using a hyperbolic tangent function to determine the current hidden state h_t.
3	As the output, give the hidden state sequence $h_1, h_2, h_3 \dots\dots\dots..h_t$.
4	Plot the Figures of LSTM Prediction.
5	Plot the Figures of Future Values.

2.4. Model Evaluation

Evaluation of the LSTM for the load model to measure the model's precision in forecasting a smart city's LF based on input features such as power involves forecasting prediction. Numerous metrics, such as the MAE, MSE, RMSE, R-squared, accuracy, and F1 score, can be used to assess the LSTM model's efficacy. The particular scenario and the kind of data being used determine which evaluation metrics should be applied. For the RMSE, the formula is:

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(C_n - \hat{C}_n\right)^2} \tag{7}$$

When N is the total number of cycles, C_n is the predicted capacity, and $\hat{C}_n$ is the ground truth capacity. To demonstrate the utility of the LSTM model as a load prediction tool, we split the data from the smart cities' dataset into training, confirmation, and testing datasets. Using the LSTM model, we train a model using the desired variables and input attributes from the training dataset. The hyperparameters of the LSTM model are adjusted using the validation dataset, and the training performance of the model is evaluated using this dataset. Evaluation of the testing dataset's performance with this final model is evaluated. The data preparation for the initial phase of LSTM model initial assessment.

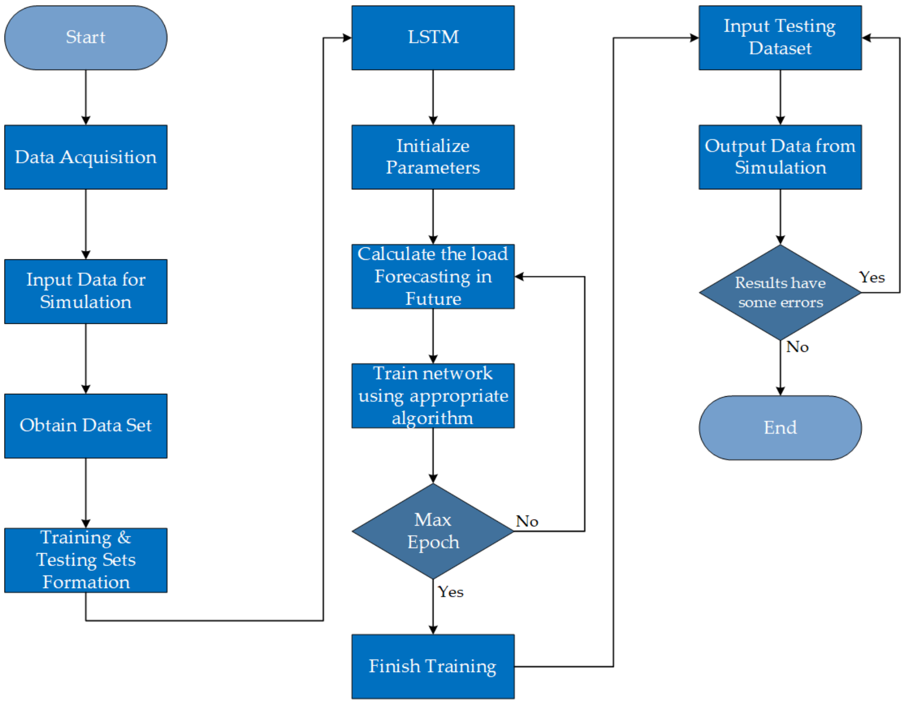

Fig. 10. Sequence of flow chart of simulation.

2.5. Comprehensive Evaluation Metrics

In this article we assessed the performance of the LSTM model with the help of different metrics like accuracy, F1 Score, RMSE, and MAE. There's a different kind of metric, and it helps explain, in more depth, how well the model predicts loads for smart cities.

- RMSE: RMSE is defined as the square root of the mean squared differences between the actual values, and the expected ones. This gives us a nice metric to use to isolate cases where the model has large deviation from the actual demand as an error matrix that punishes the error further into the costs as it is larger.
- MAE: Because the average value is taken on the absolute difference between the model's predictions and actual, MAE is a well understood measure of the typical error size. Unlike RMSE, MAE does not weigh daily forecasting reliability with the error trends. By combining MAE and RMSE, we assault a balance between a high sensitivity to larger discrepancy and a straightforward representation of overall predictive accuracy.
- F1 Score: In situations where the model categorizes demand levels (such as high vs. low energy demand thresholds), the F1 Score is very pertinent. The model's precision in properly identifying periods of high demand and its recall in avoiding missed detections are both balanced by the precision and recall harmonic mean. The F1 Score is helpful for smart city applications where distinguishing between various energy consumption levels is crucial for energy management.
- Application in Forecasting: Ultimately, the F1 Score allows us to assess how well the LSTM model understands whether an interval is a high or low demand interval, before we panic about how well it predicts future demand when the actual demand is not

balanced. Better F1 Score is the indicator of more balanced effectiveness of preventing false alarms and detection of real high demand times.

- Accuracy: Forecasting models are frequently used and yet their accuracy is typically insufficient to evaluate them in dynamic situations. As a supplement to the measures listed above. Accuracy allows us to measure the entire correct predictions in an overall picture based on all predicted data points, giving us an idea of how reliable the model is.

2.6. Hyperparameter Tuning for Optimization of LSTM Model Performance

2.6.1. Importance of Hyperparameter Tuning in the LSTM Model

Deep learning model works as good and accurate as possible depends on hyperparameter adjustment. Keys for LSTM networks to learn temporal dependencies are these hyperparameters such as number of layers, neurons, learning rate, batch size, dropout rate, see Table 3. To balance an accurate prediction and computational efficiency, we adjust these parameters.

Table 3. Optimal hyperparameters.

Hyperparameter	Tested value	Optimal value	Effect on model performance
Number of layers	1,2,3	2	Prevents overfitting while capturing temporal patterns
Neurons per layer	5,100,150,200	100	Balances computational effectiveness with model capacity
Learning rate	0.001,0.005,0.01	0.005	Assures great precision and steady convergence
Batch size	16,32,64,128	32	Balances training speed and generalization
Dropout rate	0.1,0.2,0.3,0.4,0.5	0.3	Keeps the model accurate by avoiding overfitting

2.6.2. Hyperparameter Tuning and Optimization Process

To ensure optimal performance of the LSTM model in load forecasting, the following hyperparameters were fine-tuned for this research:

- Number of LSTM Layers and Neurons in Each Layer: However, the model can be enriched with more LSTM layers and neurons to improve detection of more complexed patterns in the data but any number of neurons excessive will make the model overfitting and take long training time. By considering a range of configurations with 1 to 3 LSTM layers and 50 to 1000 neurons, we demonstrate that a two-layer LSTM with 100 neurons per layers offers an optimal trade off between accuracy and efficiency.
- Learning Rate: During training, the model's weights are updated in steps determined by the learning rate. While a low learning rate increases precision but may result in longer training periods, a high learning rate speeds up training but increases the chance of missing the best answer. We investigated learning rates ranging from 0.001 to 0.01 using a grid search technique; a rate of 0.005 produced great accuracy and sustained convergence.

- Batch Size: The number of samples processed before the model's weights are updated depends on the batch size. While higher batch sizes speed up training but may result in less accurate updates, smaller batch sizes usually improve generalization but extend training time. We tested batch sizes ranging from 16 to 128 and found that a batch size of 32 achieved an ideal balance between assuring reasonable training times and retaining high accuracy.
- Dropout: Dropout is a regularization strategy that randomly "drops" a portion of neurons during training to avoid overfitting. For this research, we investigated dropout rates between 0.1 and 0.5. Dropout rate was found to be 0.3 which was optimal, it minimised the overfitting and still had a good accuracy percentage.

2.6.3. Hyperparameter Tuning Method: Grid Search

We utilized grid search, a systematic approach to hyper parameter tuning where we specify a range of possible values for each hyperparameter and evaluate multiple values. We trained model for each configuration and evaluated it with RMSE and MAE, see Fig. 11. Like other problems, the model was selected by finding the configuration having the minimum RMSE and MAE over the validation dataset.

The final chosen hyperparameters are: number of layers is 2; number of neurons per layers is 100; Learning rate: 0.005; batch size is 32; dropout rate is 0.3

Finally, we obtain various behavioral visualizations provided by the LSTM.

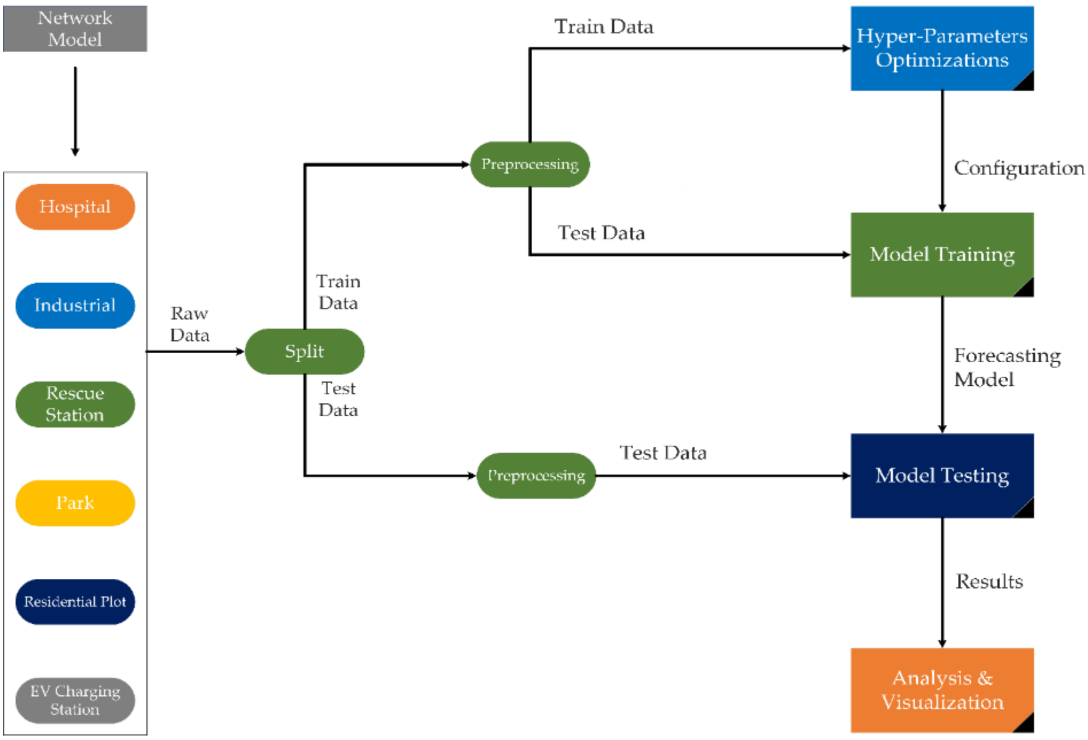

Fig. 11. Data flow between different parts of the developed model.

3. DATA COLLECTION AND RESULTS

Various load data are gathered to show how the load in a smart city operates differently to validate short-term load forecasts, see Table 4. The quantity of load that is monitored and sent alongside the system varies depending on the day. The different figures and data graphs are given below.

Table 4. Load forecasting data of smart cities starting in 2002.

Date	Residential Plot [W]	Industry [W]	Hospital [W]	Park [W]	Rescue [W]
01/01/2002	5715	62452.32	15128	2170	1754
02/01/2002	5641	6245.43	14526	2012	1845
03/01/2002	5113	5211.41	13832	1845	1945
04/01/2002	5258	3236.12	12456	2015	1245
05/01/2002	5456	3778.47	11256	2312	1452
06/01/2002	5651	3897.7	14556	2415	1654

The energy consumption of residential plots, hospitals, industries, parks, and rescue services is thoroughly examined in Figs. 12–26. This analysis includes yearly consumption patterns, LSTM-predicted data, and future forecasted values. The dynamic trends in energy use, the precision of the LSTM model predictions, and the expected future demands over the analysed time are all displayed in these visuals, which offer a thorough perspective. Now, we discuss the Residential Plots data, as described in Figs. 12-14.

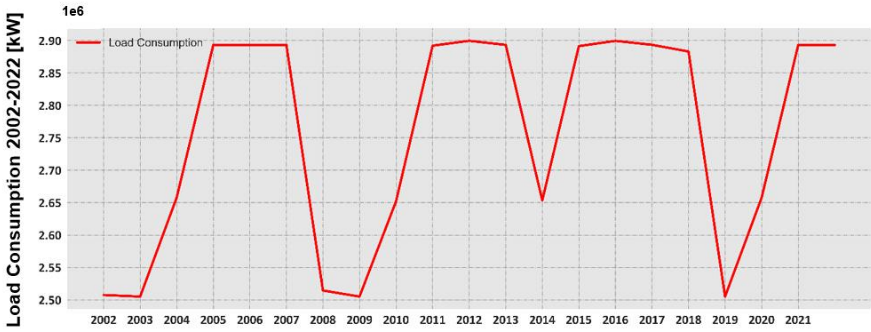

Fig. 12. Annual consumption data of residential plots.

The annual load consumption for residential plots over a 20-year period is shown in Fig. 12, highlighting trends and fluctuations in energy use. The graph shows clear peaks and troughs that correspond to times when energy consumption was high and low. These kinds of visualizations are essential for spotting seasonal patterns and irregularities in energy usage, which serve as the foundation for accurate load forecasting.

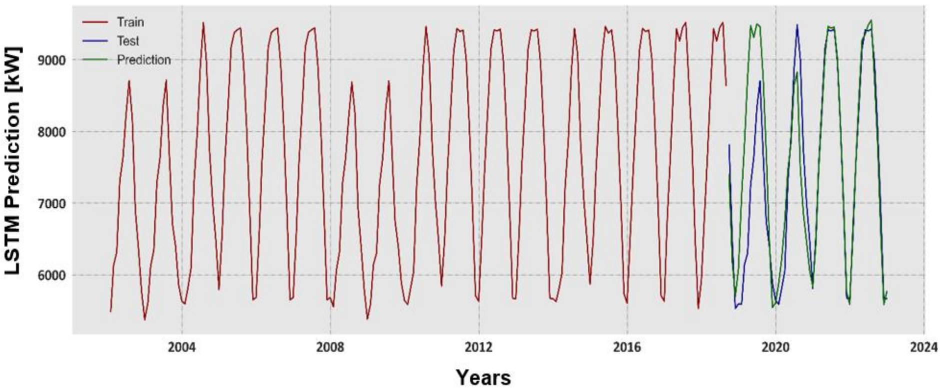

Fig. 13. LSTM predicted data.

The training and testing stages of the LSTM model for residential load forecasting are displayed in Fig. 13. While the test data (blue line) assesses the model's performance on unknown data, the training data (red line) covers prior years. The model's capacity to grasp temporal dependencies and precisely estimate energy demands in dynamic contexts is illustrated by the tight alignment of the predicted values (green line) with actual trends.

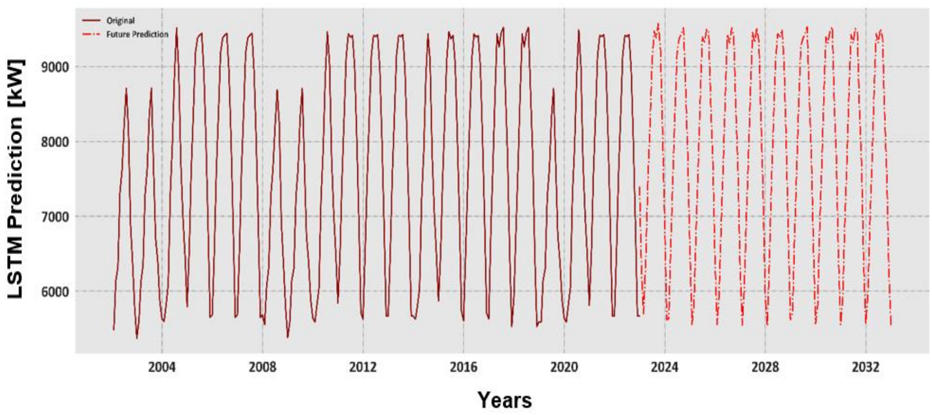

Fig. 14. Original vs forecasted data.

Projected energy consumption for the residential sector is visualized in Fig. 14, which extrapolates the LSTM model's projections to future levels. The future projections (dotted line) project future trends, while the original data (solid line) acts as the baseline. Now, we discuss the Hospital data, as described in Figs. 15-17.

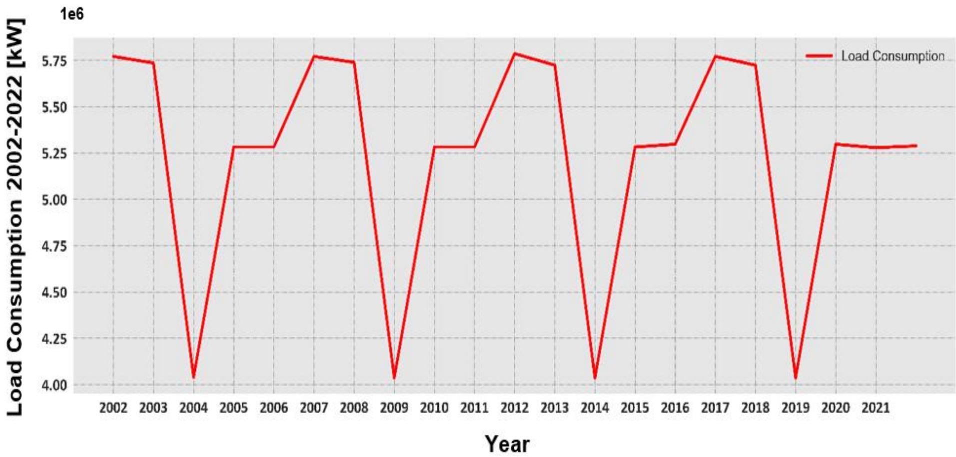

Fig. 15. Annual consumption data of hospitals.

The annual patterns in energy consumption for the given time period are displayed in Fig. 15, which shows both load demand stability and fluctuation. The peaks and troughs highlight times when energy use produced and dropped, providing information about potential operational or seasonal variables affecting consumption trends.

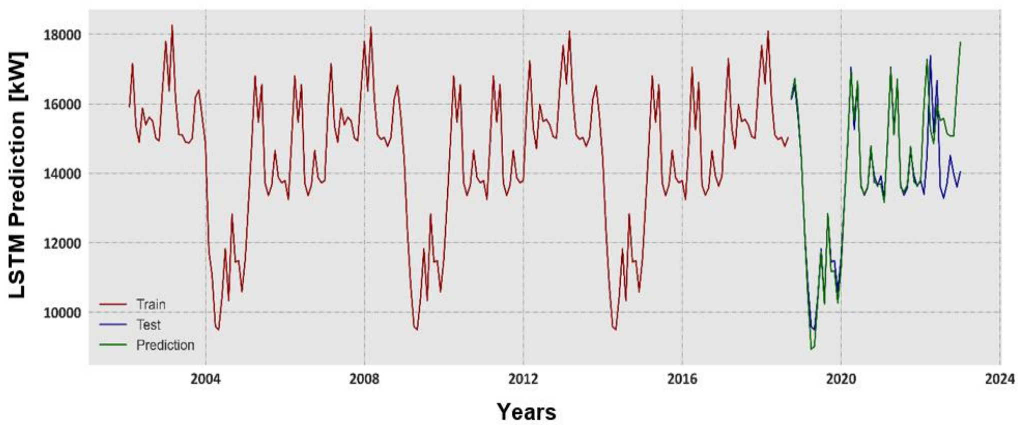

Fig. 16. LSTM predicted data.

This plot shows the development of the LSTM model (Fig. 16) with training (red line), testing (blue line), and predictions (green line). Throughout testing, its correctness and reliability in processing non-linear and time-dependent data are demonstrated as the model closely matches real patterns. The correlation between expected and actual data confirms effectiveness of the model in forecasting job outcomes in complex situations.

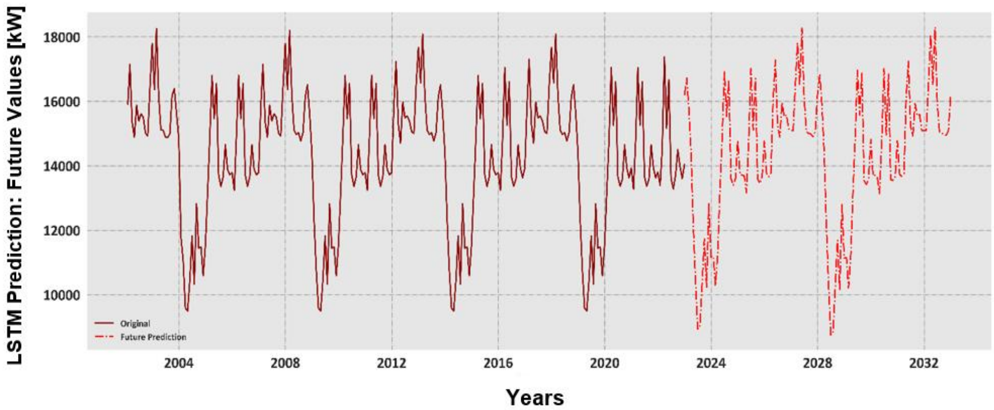

Fig. 17. Original vs forecasted data.

Using the predictive capacity of the LSTM model, Fig. 17 shows how load can be forecasted into the future. Future predictions (dotted line) shows the trend of energy consumption and original dataset (solid line) shows the history of energy consumption. Predicted values show the fluctuations, which represent changes in the dynamics of energy demand conditions for better response planning of resource management and alters of urban infrastructure.

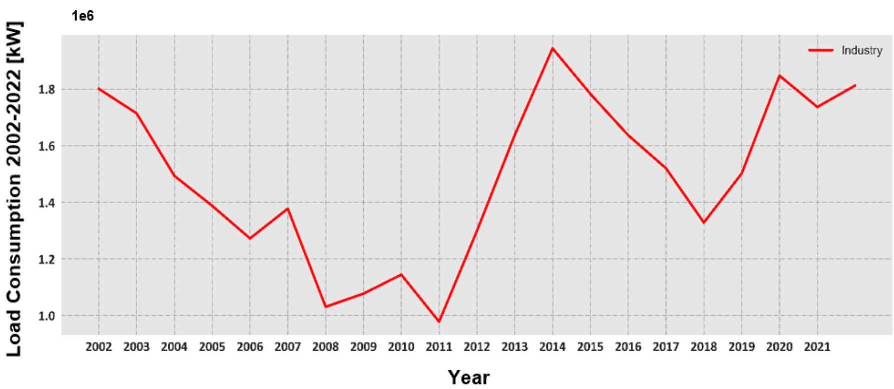

Fig. 18. Annual industrial consumption data.

Fig. 18 demonstrates the annual patterns of load forecasting of industrial loads by the years 2002 to 2022.

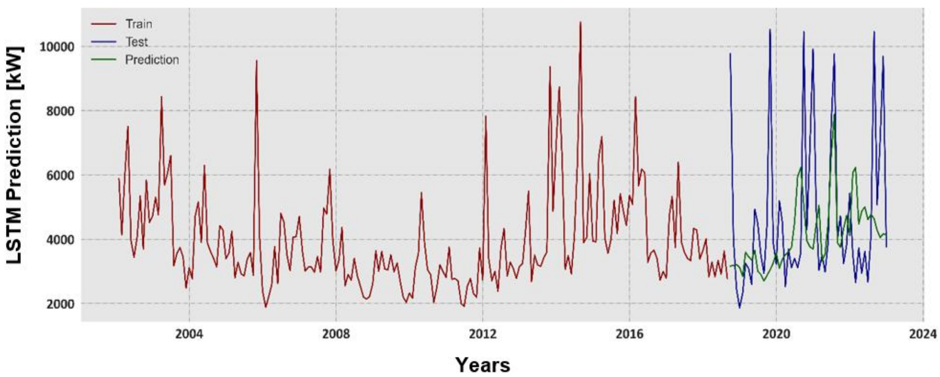

Fig. 19. LSTM predicted data.

In particular, Fig. 19 shows the training, testing, and prediction phases of the LSTM model for forecasting industrial energy load. The algorithm learns historical pattern represented by red line while the test data (blue line) evaluates the accuracy of the model's predictions. The successfully predicted values (green line) nearly match the test data and show the model's capacity to adapt to the particular unpredictability of industrial energy demand.

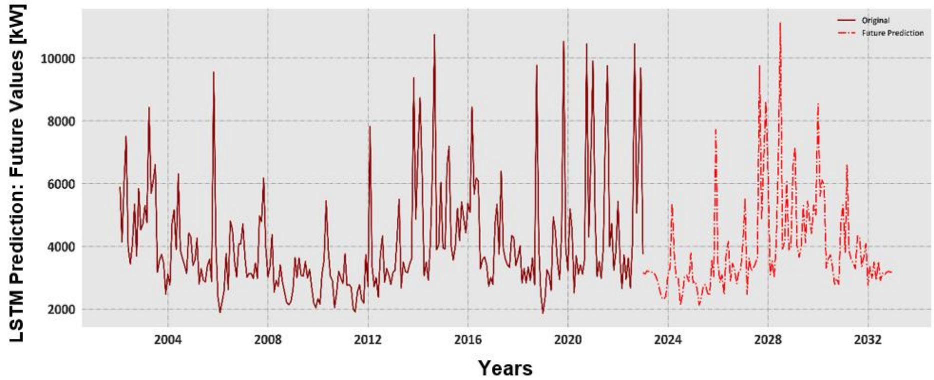

Fig. 20. Original vs forecasted data.

By projecting energy demands for the industrial sector, this graphic (Fig. 20) expands the forecasting power of the LSTM model into the future. Whereas the projected values (dotted line) foretell future trends, the original data (solid line) offers historical context. Proactive planning of energy resources depends on these projections, which help industry deal with possible issues like supply shortages or peak demand. Now, we discuss the data park, as shown in Figs. 21-23.

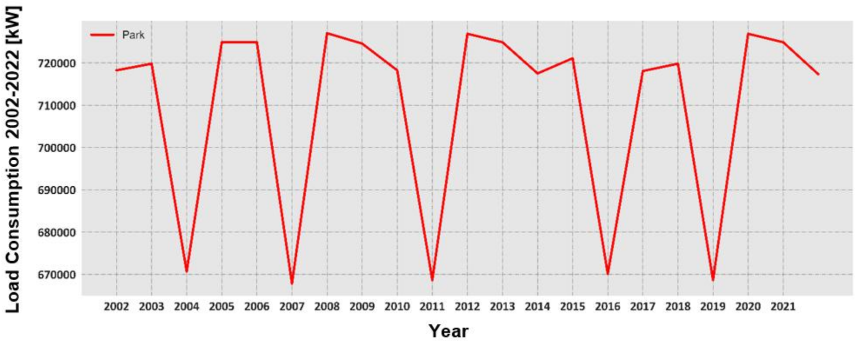

Fig. 21. Annual consumption data of park.

Annual patterns of energy use are depicted in Fig. 21, which shows a comparatively constant demand over time with sporadic declines and recoveries. By displaying variations that could be caused by outside variables like policy changes or upgrades to infrastructure, the illustration emphasizes the energy network's resilience and adaptability.

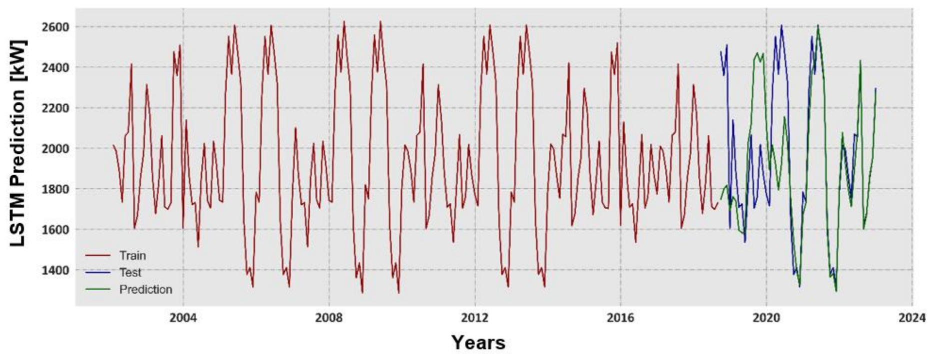

Fig. 22. LSTM predicted data.

The LSTM prediction development is shown in this plot (Fig. 22), which displays training (red line), testing (blue line), and forecast data (green line). The alignment of test and prediction data shows that the LSTM model has strong accuracy. Notably, this is a useful technique for real-time load forecast because the fluctuations in energy use reflect changing urban demands.

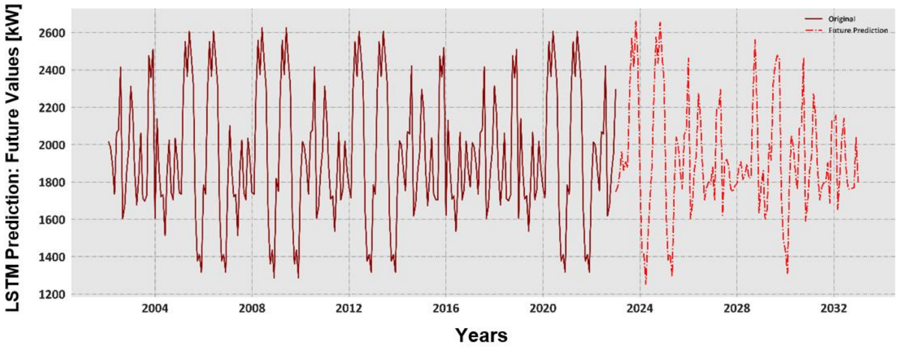

Fig. 23. Original vs forecasted data.

With the initial data (solid line) providing a strong basis for the LSTM model's predictions (dotted line), Fig. 23 expands the prediction timeline to encompass future values. The graphic depicts possible patterns in energy consumption while taking operational and seasonal variances into consideration. It highlights how the LSTM model can offer practical insights for strategic decision-making and future energy management. Now, we discuss the Rescue data, as shown in Figs. 24-26.

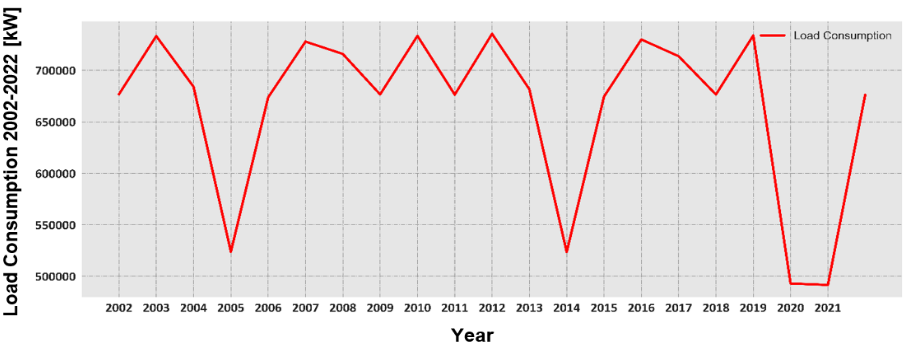

Fig. 24. Annual consumption rescue data.

The annual energy consumption patterns of rescue services over a two-decade period are depicted in Fig. 24. The peaks and troughs show varying energy demands, which may be caused by operational expansions, emergency activity levels, or outside disturbances. These kinds of insights are essential for allocating energy as efficiently as possible while preserving rescue operations.

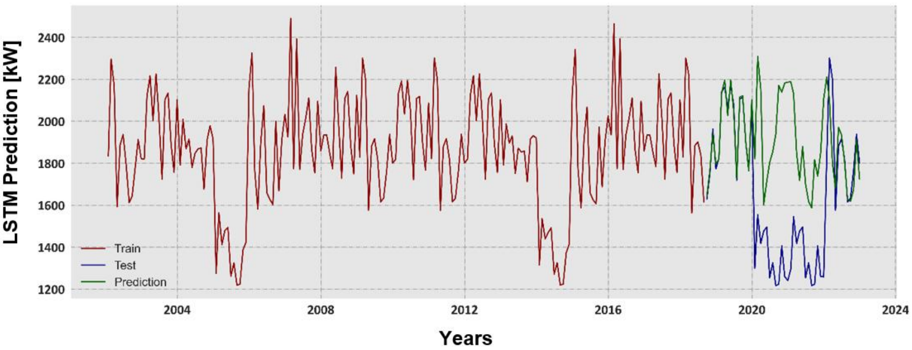

Fig. 25. LSTM predicted data.

The training (red line), testing (blue line), and forecast (green line) phases of the LSTM model for rescue services are depicted in Fig. 25. The dynamic and frequently erratic energy patterns that are typical of this industry are captured by the model. The model's accuracy in predicting energy demands is demonstrated by the overlap of test and forecast data, which guarantees effective resource planning for emergency operations.

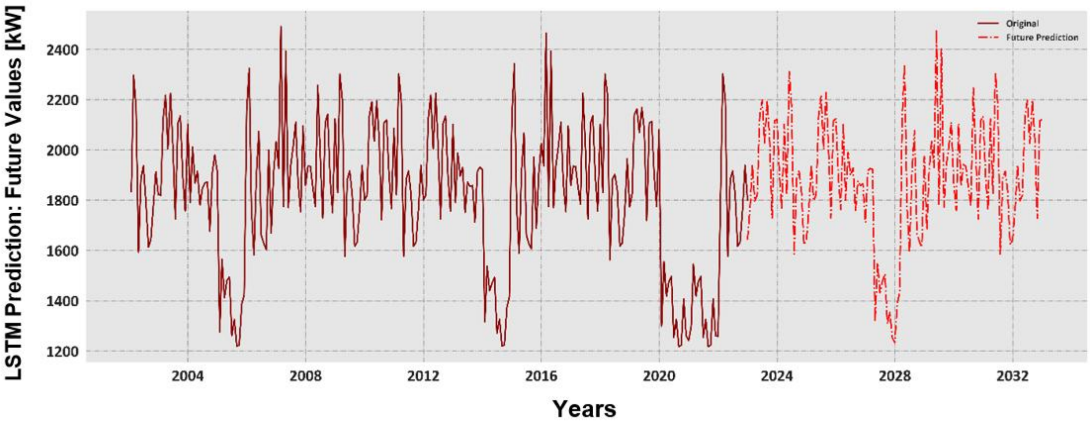

Fig. 26. Original forecasted data.

The energy forecasting timeline is broadened in Fig. 26, displaying forecasts (dotted line) that are based on historical data (solid line). To assist crucial rescue missions, future load estimates facilitate proactive infrastructure expansion and contingency planning by considering fluctuations and potential increases in energy demand. The precision of the Smart City data predictions is illustrated in the Table 5, which also presents the RMSE and loss associated with the model.

Table 5. RMSE, loss and accuracy of inputs.

Input	RMSE	Loss	Loss [%]	Accuracy
Residential plots	0.040939193257074055	0.0001	4.0	96%
Hospital	0.04350975468418003	0.0001	4.3	95.7%
Park	0.051962410171365395	0.0001	5.1	94.6%
Rescue	0.03746223564954682	0.0001	3.7	96.3%
Industry	0.09736663127160096	0.0014	9.7	92.3%

4. RESULTS AND DISCUSSION

In this study, we describe in detail the development of a graphical user interface (GUI) based on LSTM artificial neural networks for load forecasting in smart cities. Additionally, a visual depiction, for example, that given in Fig. 27, can be used to provide further insight into the design and functions of the GUI application.

4.1. Structure and Functionality of the GUI Program

4.1.1. Understanding Load Forecasting

Load forecasting is affected by many factors such as historical load pattern, weather conditions, other special occurrences and economic indicators. As LSTM neural networks, as a kind of machine learning technology, can recognize complex temporal patterns and make accurate prediction for the future load requirements, this thesis has demonstrated research results.

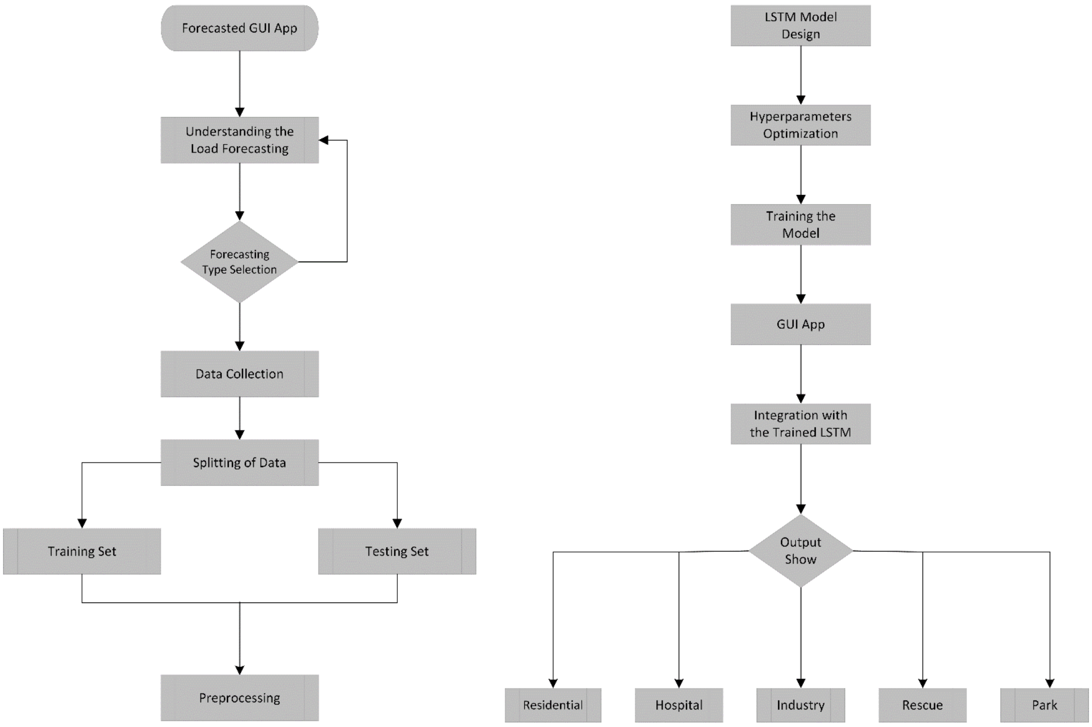

Fig. 27. Structure and work of a GUI.

4.1.2. *Data Collection and Preprocessing*

Developing a load forecasting system involves making a collection of as much relevant data as possible. Historical load data allows you to see patterns of consumption in the past, weather data gives indications of the impact of temperature, humidity and other atmospheric conditions on load demand.

4.1.3. *LSTM Model Design*

One of the RNNs is LSTM neural network which can learn sequential data temporally. When developing an LSTM model, several parameters must be set during the design: number of hidden layers, amount of LSTM per layer, type of activation functions. We combine past load trends with weather information in a LSTM model that generates accurate forecasts.

Finally, if you pick the wrong optimization method and loss function, model training will only be successful.

4.1.4. *Training the LSTM Model*

When we finally give it some data, we tell the LSTM model that rather than continuously updating its model parameters, so that its predictions match with its prediction error. So, we set the data as the training set and validation set. The model is trained on the training set and evaluated on the validation set. An LSTM model can train accurately to detect temporal relationship and predicts any load point. It's an iterative training process where the model can satisfy the validation set. Early stops and using dropout to reduce overfitting to generalizing the model better making it easier.

4.1.5. *GUI Development*

The LSTM model is trained and a user-friendly GUI for load forecasting is then developed. The interfaces with an application, which is the GUI that allows users (venue managers, utility operators, etc.) to input data about the date, time and weather conditions and generate load forecasts for that period of time.

4.1.6. Integration and Deployment

Once the trained LSTM model is established, we need to link the GUI application to it for real time load forecasting. Smooth integration of GUI and the forecasting engine is one of the criteria of this integration as well as integrating the parameters of the model learnt into the application. Depending on scale and needs of smart city initiative, programs can be installed on local servers or cloud platforms. Applications are disregarded when their security, scalability, and reliability are not considered. The GUI of the application is shown in Fig. 28, five options are provided for selecting the input data column. The raw data supplied to the application is used an attached LCD (80x6) to show progress of model training and overall loss.

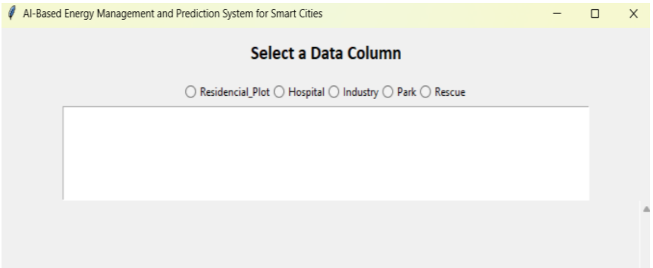

Fig. 28. Application interface and running mode display.

4.2. Smart City Inputs in Terms of Load

4.2.1. Residential Plot

Data and cutting-edge algorithms have completely changed the way the energy is forecasted in smart city residential areas, where the intricacy of the capacity has been predicted. Insight gives us the ability to proactively manage demand on the demand side, taking care of surges and encouraging energy savings to residents. By using AI driven systems that are more efficient and sustainable, the future of smart communities looks greener as seen in Fig. 29.

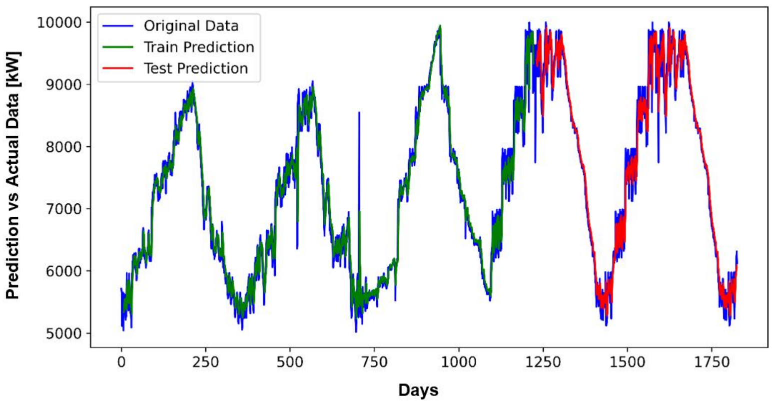

Fig. 29. Residential plot results.

4.2.2. Hospitals

Since hospitals rely on efficient energy management to sustain operations, ensure patient comfort and power advanced medical devices, hospitals require energy efficient equipment to ensure patient welfare and to meet critical mission requirements. These AI driven systems focus on optimization of energy use, reduce waste and improve efficiency to support high quality healthcare delivery, see Fig. 30.

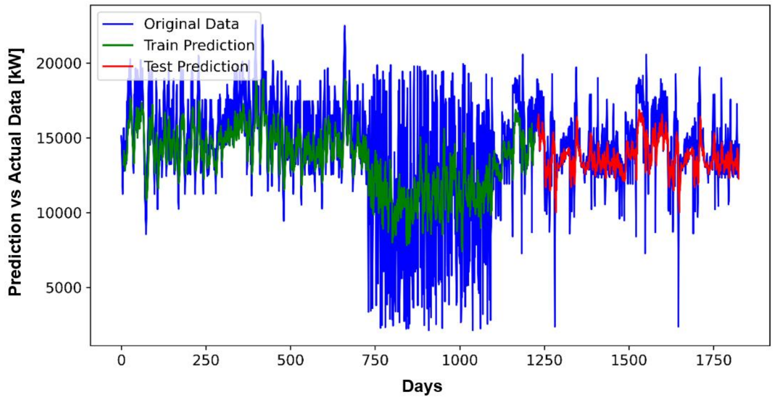

Fig. 30. Hospital results.

4.2.3. *Industry*

In embracing AI-driven energy management systems across industries, it's a fantastic opportunity to increase energy efficiency, decrease costs and minimize environmental footprints. Through an analysis of energy consumption patterns, business can improve productivity, promote sustainability and gain a competitive advantage at a global scale. We can all power a brighter future together, see Fig. 31.

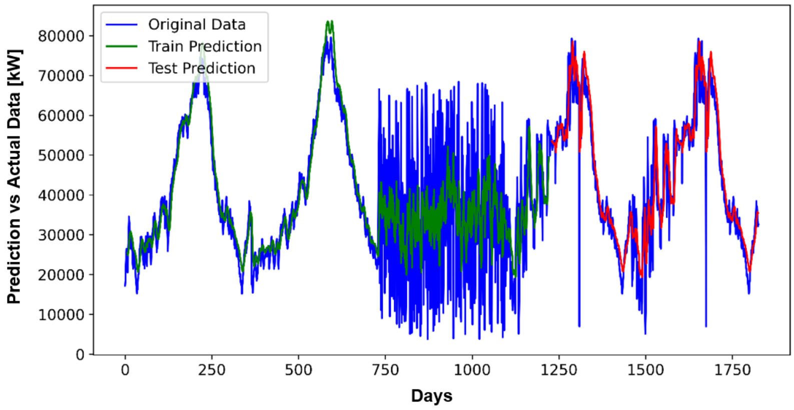

Fig. 31. Industry results.

4.2.4. *Park*

Energy needs in smart city parks were successfully forecasted by AI energy management, which was able to trace and predict such fine-grained consumption, see Fig. 32.

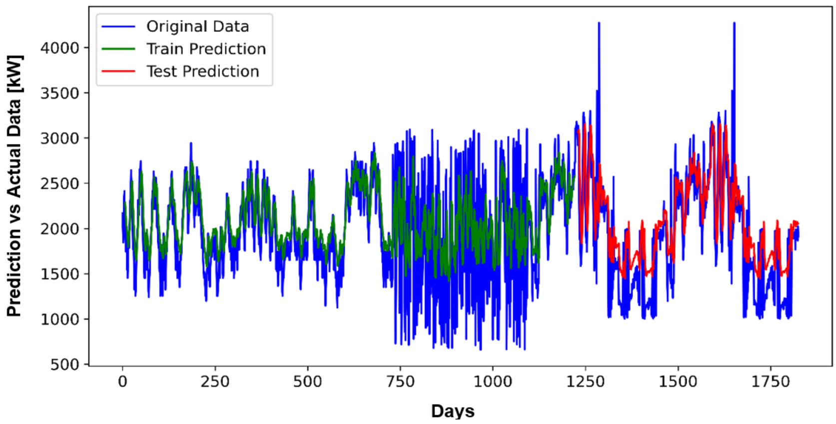

Fig. 32. Park results.

4.2.5. Rescue

An AI driven load forecasting model was developed using a 5-year dataset obtained from a grid station measuring residential, hospital, industrial, park and emergency services. To evaluate performance the split into equal 3.5 years for training and 1.5 years for testing, see Fig. 33.

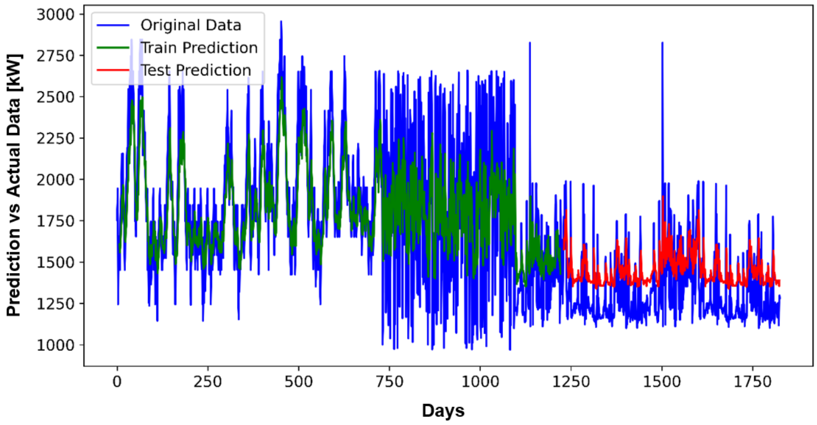

Fig. 33. Rescue Results

4.3. Comparative Advantage of LSTM in Dynamic and Nonlinear Load Forecasting

The traditional statistical methods, such as ARIMA, require stability and linearity; however, they are not always viable for handling the varying load patterns in smart cities. Additionally, Smith et al. (2023) [33] recently demonstrated that LSTM delivered significantly higher prediction accuracy than ARIMA, increasing forecasting accuracy by more than 20% for smart grids in load forecasting. A well-captured temporal dependency for long periods is, among other things, great about load forecasting. This is something that traditional machine learning methods like Support Vector Machines (SVM) or Decision Trees don't do at this level. In their exciting research on forecasting tasks with time-dependent data in smart cities, Lee and Kim (2024) [34] found that LSTM models dropped RMSE by 15% over SVM. In the situations where the load patterns change drastically with time, LSTM family shines most, improving prediction accuracy by representing the short term as well as the long-term dependencies. It demonstrates the potential of advanced techniques in changing our approach to load forecasting.

Furthermore, while highly sophisticated deep learning models, such as CNN-LSTM hybrids, other deep learning models, and beaconing, can be utilized as alternatives in real-time forecasting cases, they are bound by the high processing requirements and the large dataset requirements. On the other hand, only LSTM networks are capable of straddling high predictive power and low computational complexity. In [35], by utilizing such low computing cost, CNN-LSTM achieved slightly more accuracy than the LSTM alone. However, it was almost as good as CNN-LSTM with much lower computing costs for scalable and real-time load forecasting in smart city systems.

4.4. Training Process and Verification of Results Using LSTM

4.4.1. Training Time and Computational Resources

The present research used NVIDIA RTX 3060 GPU, commonly considered ideal for managing deep learning calculations for load forecasting across some branches of industries such as residential, hospitals, industry, parks, and rescue facilities. An average training time of only 1 hour and 8 minutes. The duration of this is indicative of the complex and interesting data patterns each sector presents, which are, in turn, a consequence of different hyperparameters. To achieve high training efficiency and performance, we adjusted a handful of key factors including a number of LSTM layers, learning rate, and batch size, and we did it very successfully. In order to clarify, Table 6 provides a detailed account of training durations for each sector model.

By bosting the GPU, the model could effectively learn complex temporal patterns in a surprisingly short time.

Table 6. Comparative advantage of LSTM in dynamic and nonlinear load forecasting.

Model	RMSE	Accuracy Improvement [%]	Key Feature	Limitations
ARIMA	0.15	-20%	Linear relationships	Assumes stationarity, struggles with nonlinearity
SVM	0.12	-15%	Non-temporal models	Lacks temporal dependency modeling
CNN-LSTM	0.08	+5%	Captures spatial and temporal data	High computational cost, large data needs
LSTM (Proposed)	0.037-0.097	+6.5%	Models short and long-term dependencies	Nonsignificant in this context

4.4.2. Verification of Results Across Sectors

4.4.2.1. Quantitative Validation

Performance Metrics: Here we started comparing our forecasts with real, observed data and calculated for each sector a RMSE and MAE. These are important metrics, like prediction accuracy, they show RMSE, larger but MAE, average error.

Tables 7 and 8 show the results for RMSE and MAE for each sector, its model could very accurately predict load patterns.

Table 7. Training time comparisons.

Sector	Training time [min]	Adjusted model parameters
Residential	20	Learning rate, batch size
Hospitals	14	Number of layers, dropout
Industry	11	Optimizer type, sequence length
Parks	15	Batch size, learning rate
Rescue	8	Epochs, early stopping criteria

Table 8. LSTM model performance metrics.

Sector	RMSE	MAE
Residential	0.0409	0.0312
Hospitals	0.0435	0.0328
Industry	0.0973	0.0659
Parks	0.0519	0.0417
Rescue	0.0375	0.0294

4.4.2.2. Cross-Validation

We First we divided the data into training, validation and test sets and then used cross validation to ensure that our model performs well when presented with unseen data. By assigning 80% to training, 10% for validation and 10% for testing, we built a solid framework for success in all industries.

The attentiveness in these distributions increases our hope that this model will indeed behave so well and reliably.

4.4.2.3. Qualitative Analysis

Visual Comparison of Predicted vs. Actual Data: As seen in Figs. 12–26, we plotted the actual observed values over time against the predicted values from the LSTM model for each sector. The ability of the model to identify trends and swings in load patterns unique to each sector was clearly demonstrated by this visual comparison.

The LSTM model demonstrated its applicability for forecasting in dynamic smart city scenarios by successfully predicting both short-term and long-term changes in energy consumption.

4.4.2.4. Validation through GUI

To further confirm the model's stability, we integrated the results into a simple Graphical User Interface (GUI) that enables administrators, energy managers, and other stakeholders to view load estimates and enter data directly. A hybrid LSTM architecture is proposed to boost efficiency without sacrificing accuracy, all while refining our model based on valuable insights gathered from our extended GUI implementation.

4.5. Comparative Analysis of LSTM with Conventional and Machine Learning Approaches

4.5.1. Comparison with Statistical Approaches

Let's explore some well-known statistical methods for load forecasting, such as SARIMA and ARIMA! These powerful techniques are widely used in time-series analyses, although they can sometimes struggle with non-linear trends. Challenges are:

- Limitations of Statistical Models: Statistical models like ARIMA and SARIMA play a valuable role in analysing data, but they face challenges in the ever-changing landscape of smart city energy systems. Their linear approach can limit their ability to keep up with dynamic, non-linear load behaviours. However, this opens exciting opportunities for developing innovative methods that can better adapt to the evolving energy needs of our vibrant cities.
- Supporting Evidence: Exciting findings revealed that while ARIMA models had RMSE values about 15-20% higher than LSTM models, the performance of both methods in load forecasting for smart systems shows great potential for future advancements in cities [33]. This difference highlights an exciting opportunity ARIMA's focus on linear relationships presents a challenge, but understanding non-linear dynamics is key for enhancing load forecasting in our vibrant, demand-rich urban landscapes.

4.5.2. Comparison with Machine Learning Algorithms

Explore the exciting world of traditional machine learning techniques such as K-nearest neighbors (KNN), Decision Trees, and Support Vector Machines (SVM)! These methods shine in regression and classification tasks, though they face challenges with time-series data that has sequential dependencies.

- Limitations of machine learning algorithms: temporal dependencies are not intrinsically accounted for by traditional machine learning models, and including time-based patterns frequently requires significant pre-processing. For load forecasting, which mostly depends on recognizing historical trends to estimate future demand, this lessens their efficacy.
- Load forecasting in smart city applications, LSTM models produced a 15% lower RMSE than SVM. Because SVM has trouble handling sequential dependencies, the researchers observed that LSTM's recurrent nature allowed it to properly handle time dependencies [34]. This is a significant advantage over SVM.
- Summary of the results: the performance of the LSTM is summarized up in Table 9 below, which highlights its usefulness for practical applications by showcasing its low RMSE and MAE values across several sectors.

Table 9. Summary of the LSTM performance.

Technique	RMSE	MAE	Key advantages	Key limitations
ARIMA	0.15	0.12	Simple implementation	Assumes linearity, lacks adaptability
SVM	0.12	0.10	Effective for classification tasks	Poor handling of time-series, lacks temporal awareness
Decision tree	0.14	0.11	Easy interpretation	Sensitive to noise, limited in time-series handling
LSTM (Proposed)	0.037-0.097	0.029-0.065	Effective for time-series, adaptable	Moderate training time

4.5.3. Practical Benefits of LSTM for Real-Time Forecasting

For real-time load forecasting in smart cities, LSTM offers useful advantages in addition to its high prediction accuracy:

- Processing efficiency: LSTM is appropriate for real-time applications without requiring a lot of hardware because it only needs moderate processing resources.
- Configuration user interfaces: because of LSTM's flexibility, it can be easily integrated into applications such as the created GUI, enabling stakeholders to make accurate, data-driven decisions about energy management.

4.6. Comparative Analysis of LSTM with Machine Learning and Deep Learning Techniques

4.6.1. Limitations of CNNs and Hybrid Models in Real-Time Forecasting

- Computational cost: significant computational resources, such as more potent GPUs and extensive training datasets, are needed for CNN and hybrid models like CNN-LSTM. These models' lengthier training timeframes and higher energy usage can make them challenging to implement successfully for real-time applications in smart cities. In contrast, the efficiency of LSTM networks in computations renders them a more suitable choice for real-time predictions.

- Example: CNN-LSTM models outperformed standalone LSTM models in terms of accuracy (by roughly 3–5%), they necessitated 40% more computational resources and longer training times, which is frequently unfeasible for load forecasting in smart cities [35].

4.6.2. Advantages of LSTM in Capturing Temporal Dependencies

- Time series adaptability: because LSTM networks are designed to process sequential data, time-series forecasting jobs like load prediction are a natural fit for them. Because of its special memory cells and gating mechanisms, LSTM can successfully simulate both short- and long-term dependence in load patterns and preserve crucial information across lengthy periods.
- Supporting study: research conducted indicates that LSTM models exceeded the performance of CNN models in load forecasting for intelligent cities, showing RMSE values that were 15–25% lower [36]. In tasks involving sequential dependencies, where CNNs alone could not match the accuracy because of their lack of temporal processing, the researchers found that LSTM's temporal capabilities made it a superior fit, see Table 10.

Table 10. Advantages of LSTM compared to other machines and deep learning techniques.

Model	RMSE	MAE	Key advantages	Key limitations
CNN	0.13	0.11	Excellent for spatial features	Lacks temporal awareness
CNN-LSTM	0.08	0.07	Captures spatial and temporal dependencies	High computational cost, long training time
LSTM (proposed)	0.037-0.097	0.029-0.065	Effective for time series, efficient for real-time	Moderate training time, suitable for temporal data

4.7. Scalability and Flexibility of the LSTM Model and GUI for Different Urban Environments

4.7.1. Scalability of the LSTM Model Across Different City Environments

Scalability refers to the LSTM model's capacity to handle a range of data inputs and perform effectively in cities with varying infrastructure, inhabitants, and energy consumption patterns.

- Model adjustments for different city scales: the LSTM model can be adapted to suit various urban environments by modifying its architecture. This includes adjusting the number of layers, neurons, and hyperparameters such as the learning rate and batch size. In cities with extensive datasets, increasing the model's depth can improve its ability to capture complex energy usage trends.
- Adaptability to diverse data sources: to enhance forecasting precision in diverse urban settings, the LSTM model can integrate information from multiple sources, such as demographic statistics, weather predictions, and live sensor data. For example, cities equipped with advanced sensor networks can utilize real-time data flows to refine the forecasts produced by the LSTM model.

- Supporting evidence: LSTM models can be efficiently scaled to handle high-volume datasets in larger cities, as shown in [37]. They noted that enhancing model complexity, such as adding layers, led to better performance without significantly increasing computing requirements.

4.7.2. Flexibility of the GUI for Diverse Urban Infrastructures

Due to the adaptable nature of the GUI developed for this research, city officials and energy managers are able to engage with the model and modify it to fit their specific infrastructure requirements and data preferences.

- Customization for varying data inputs: by giving users the option to select or decline specific data input categories like weather information, economic metrics, or consumption statistics relevant to specific sectors (such as residential versus industrial), the GUI can be customized for diverse urban environments. This adjustment ensures that the GUI is compatible with cities at varying stages of technological advancement, from fully developed smart cities to those just beginning their digital transformation journey.
- Scalable deployment options: the graphical user interface (GUI) is suitable for cities with different IT infrastructures, as it can be installed on either local servers or cloud-based systems. Utilizing a cloud-based setup removes the need for on-site hardware by allowing remote data processing in areas with limited computational capabilities.
- Supporting evidence: flexible graphical user interfaces boost user participation and improve operational effectiveness by allowing stakeholders to personalize data entries and visualize results in ways that align with their specific city needs, based on a study conducted by Lee and Park (2023) regarding scalable GUI tools in urban energy administration [38], see Table 11.

Table 11. Comparison of LSTM model and GUI in urban environments.

Aspect	LSTM Model	GUI
Scalability	Adjustable architecture for large data volumes	Cloud or local server deployment options
Data adaptability	Integrates real-time sensor and demographic data	Selectable data input types for flexibility
Urban fit	Suitable for cities with varying energy patterns	Customizable for diverse infrastructure

The LSTM model and GUI are appropriate for various urban settings due to their scalability and versatility. The LSTM model's architecture can be modified to manage massive data volumes in bigger cities by modifying the model depth and hyperparameters. Additionally, the model's ability to integrate several data sources allows it to adapt to the unique infrastructure requirements and energy use patterns of different cities.

The GUI broadens this versatility and enables cities with varying degrees of IT proficiency to use it by letting users change data inputs and choose between local as well as cloud-based deployment.

LSTM models with flexible GUIs enhance engagement and use across various urban settings. Overall, the summary for the LSTM Model Summary is shown in Table 12.

Table 12. LSTM model metric summary.

Metric	Definition	Model value	Relevance
RMSE	Penalizes larger errors to highlight significant deviations	0.037-0.097	Useful for identifying outlier errors
MAE	Measures the average error magnitude	0.029-0.065	Offers a balanced view of typical prediction accuracy
F1 Score	Harmonic mean of precision and recall	0.85	Balances precision and recall in demand classification
Accuracy	Overall measure of correct predictions	90.3%-97.9%	Provides a general indication of model reliability

5. CONCLUSIONS AND FUTURE RECOMMENDATIONS

This paper effectively illustrated how accurate LSTM is at predicting energy loads, which is especially useful in smart city settings with intricate linear and dynamic load patterns. The capacity of LSTM to detect long-term dependencies, especially those influenced by environmental conditions, is precious because conventional approaches frequently struggle with such complications. Load prediction is essential in energy optimization because it makes it easier to regulate demand management, energy distribution, and energy generation. Traditional forecasting techniques' ability to capture complex temporal patterns associated with several factors sometimes results in their failure in smart city scenarios. This investigation used LSTM to increase load prediction accuracy by utilizing various data sources, including historical load statistics and weather data. Moreover, the development of an intuitive GUI application facilitated practical implementation, enabling city officials, energy managers, and residents to use it with ease. This platform supports strategic energy planning and investment management, promoting well-informed decision-making and improving the efficiency of energy initiatives in smart cities.

Future studies might focus on integrating real-time data feeds, advanced optimization techniques, and demand-side management strategies to enhance the accuracy and effectiveness of load forecasting models and GUI applications.

REFERENCES

[1] D. Saba, F. Hajjej, O. Cheikhrouhou, Y. Sahli, A. Hadidi, H. Hamam, "Development of an intelligent solution for the optimization of hybrid energy systems," *Applied Sciences*, vol. 12, p. 83972022, doi: 10.3390/app12178397.

[2] S. Balouch, M. Abrar, H. Abdul-Muqeet, M. Shahzad, H. Jamil, M. Hamdi, A. Malik, H. Hamam, "Optimal scheduling of demand side load management of smart grid considering energy efficiency," *Energy Research*, vol. 10, 2022, doi: 10.3389/fenrg.2022.861571.

[3] K. Saleem, L. Wang, S. Bharany, K. Ouahada, A. Ur-Rehman, H. Hamam, "Intelligent multiagent model for energy-efficient communication in wireless sensor networks," *EURASIP Journal on Information Security*, vol. 9, 2024, doi: 10.1186/s13635-024-00155-6.

[4] Y. Ghadi, T. Mazhar, S. Shah, I. Haq, W. Ahmad, K. Ouahada, H. Hamam, "Integration of federated learning with IoT for smart cities applications, challenges, and solutions," *PeerJ Computer Science*, vol. 9, p. e.1657, 2023 doi: 10.7717/peerj-cs.1657.

[5] D. Costa, J. Peixoto, T. Jesus, P. Portugal, F. Vasques, E. Rangel, “A survey of emergencies management systems in smart cities,” *IEEE Access*, vol. 10, pp. 61843–61872, 2022, doi: 10.1109/ACCESS.2022.3180033.

[6] G. Stephen, "As the chorus of dumb city advocates increases, how do we define the truly smart city?," Data-Smart City Solutions, 2022, https://datasmart.hks.harvard.edu/chorus-dumb-city-advocates-increases-how-do-we-define-truly-smart-city.

[7] D. Mills, S. Pudney, P. Pevcin, J. Dvorak, "Evidence-based public policy decision-making in smart cities: does extant theory support achievement of city sustainability objectives?," *Sustainability*, vol. 14, no.3, doi: 10.3390/su14010003.

[8] *Building a Smart City, Equitable City – NYC Forward*, 2015, https://edc.nyc/industry/smart-cities.

[9] M. Deakin, H. Al-Waer, "From intelligent to smart cities," *Journal of Intelligent Buildings International*, vol.3, no. 3, pp. 140–152, 2011, doi: 10.1080/17508975.2011.586673

[10] A. Al-Qarafi, H. Alsolai, J. Alzahrani, N. Negm, L. Alharbi, M. Al-Duhayyim, H. Mohsen, M. Al-Shabi, F. Al-Wesabi, "Artificial jellyfish optimization with deep-learning-driven decision support system for energy management in smart cities," *Applied Sciences*, vol. 12, no. 15, p. 7457, 2022, doi: 10.3390/app12157457.

[11] E. Feinberg, D. Genethliou," Load forecasting," *Applied Mathematics for Restructured Electric Power Systems*, pp. 269–285, 2020, doi: 10.1007/0-387-23471-3_12.

[12] D. Park, M. EI-Sharkawi, R. Marks, L. Atlas, M. Damborg, ''Electric load forecasting using an artificial neural network,'' IEEE Transactions on Power System, vol. 6, no. 2, pp. 442-449, 1991.

[13] S. Kiartzis, A. Bakirtzis, V. Petridis, "Short-term load forecasting using neural networks," *Electric Power Systems Research*, vol. 33, no. 1, pp. 1-6, 1995, doi: 10.1016/0378-7796(95)00920-D.

[14] H. Hippert, C. Pedreira, R. Souza, "Neural networks for short-term load forecasting: a review and evaluation," *IEEE Transactions on Power Systems*, vol. 16, no. 1, 2001, doi: 10.1109/59.910780.

[15] P. Bacher, H. Madsen, H. Nielsen, "Online short-term solar power forecasting," *Solar Energy*, vol. 83, no. 10, pp. 1772–1783, 2009, doi: 10.1016/j.solener.2009.05.016.

[16] J. Kim, "Short-term electric load prediction using multiple linear regression method," *University of California San Diego*, a thesis submitted for the degree Master of Science in Electrical Engineering, University of California San Diego, 2020.

[17] A. Saber, T. Khandelwal," IoT based online load forecasting," IEEE Green Technologies Conference, 2020, doi: 10.1109/GreenTech.2017.34.

[18] R. Lu, S. Hong, M. Yu, "Demand response for home energy management using reinforcement learning and artificial neural network," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6629–6639, 2019, doi: 10.1109/TSG.2019.2909266.

[19] K. Chui, M. Lytras, A. Visvizi, "Energy sustainability in smart cities: Artificial intelligence, smart monitoring, and optimization of energy consumption," *Energies*, vol. 11, no. 11, 2018, doi: 10.3390/en11112869.

[20] M. Waseem, Z. Lin, L. Yang, "Data-driven load forecasting of air conditioners for demand response using Levenberg–Marquardt algorithm-based ANN," *Big Data and Cognitive Computing*, vol. 3, no. 3, pp. 1–17, 2019, doi: 10.3390/bdcc3030036.

[21] G. Hafeez, Z. Wadud, I. Khan, I Khan, Z. Shafiq, M. Usman, M. Khan, "Efficient energy management of IoT-enabled smart homes under price-based demand response program in smart grid," *Sensors*, vol. 20, no. 11, 2020, doi: 10.3390/s20113155.

[22] T. Saleem, M. Chishti, "Deep learning for the internet of things: Potential benefits and use-cases," *Digital Communications and Networks*, vol. 7, no. 4, pp. 526–542, 2021, doi: 10.1016/j.dcan.2020.12.002.

[23] H. Rotib, M. Nappu, Z. Tahir, A. Arief, M. Shiddiq, "Electric load forecasting for internet of things smart home using hybrid PCA and ARIMA algorithm," *International Journal of Electrical and Electronic Engineering and Telecommunications*, vol. 10, no. 6, pp. 425–430, 2021, doi: 10.18178/ijeetc.10.6.425-430.

[24] Y. Okubo, J. Dore, T. Ojasoo, J. Miquel, ''A multivariate analysis of publication trends in the 1980s with special reference to south-east Asia,'' *Scientometrics*, vol. 41, no. 3, pp. 273-289, 1998, doi: 10.1007/BF02459045.
[25] T. Vantuch, A. Vidal, A. Ramallo-González, A. Skarmeta, S. Misák, "Machine learning based electric load forecasting for short and long-term period," World Forum on Internet of Things, 2018, doi: 10.1109/WF-IoT.2018.8355123.
[26] J. Kim, J. Moon, E. Hwang, P. Kang, ''Recurrent inception convolution neural network for multi short-term load forecasting,'' *Energy Build*, vol. 194, pp. 328–41, 2019, doi: 10.1016/j.enbuild.2019.04.034
[27] T. Labeodan, W. Zeiler, G. Boxem, Y. Zhao, ''Occupancy measurement in commercial office buildings for demand-driven control applications—A survey and detection system evaluation,'' *Energy and Buildings*, vol. *93*, pp. 303-314, 2015, doi: 10.1016/j.enbuild.2015.02.028.
[28] R. Mubashar, M. Awan, A. Muhammad, Y. Awais, V. Singh, "Efficient residential load forecasting using deep learning approach," *International Journal of Computer Applications in Technology*, vol. 63, no. 3, pp. 205-214, 2022, doi: 10.1504/IJCAT.2022.124940
[29] B. Farsi, A. Manar, B. Nizar, E. Ursula, "On short-term load forecasting using machine learning techniques and a novel parallel deep LSTM-CNN approach," *IEEE Access*, vol. 9, pp. 31191-31212, 2021, doi: 10.1109/ACCESS.2021.3060290.
[30] U. Mumtahina, S. Alahakoon, P. Wolfs, '' Hyperparameter Tuning of Load-Forecasting Models Using Metaheuristic Optimization Algorithms—A Systematic Review,'' *Mathematics*, vol. 12, no. 21, p. 3353, 2024, doi: 10.3390/math12213353.
[31] O. Ignatius, A. Alayande, P. Orukpe, "Accurate probability distribution model determination and forecasting of peak load demand in Nigeria," *Jordan Journal of Electrical Engineering*, vol. 4, no. 4, pp. 186-194, 2018.
[32] S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural Computing*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735
[33] M. Elsaraiti, A. Merabet, ''A comparative analysis of the ARIMA and LSTM predictive models and their effectiveness for predicting wind speed,'' *Energies,* vol. 14, p. 6782, 2021, doi: 10.3390/en14206782.
[34] H. Lee, S. Kim, ''LSTM networks for improved load forecasting in smart city applications: A comparative study with SVM,'' *Energy Systems Journal*, vol. 32, no. 1, pp. 98-110, 2024.
[35] M. Bartouli, I. Hagui, A. Msolli, A. Helali, F. Hassen, ''Smart grid load forecasting models using recurrent neural network and long short-term memory,'' *Jordan Journal of Electrical Engineering*, in press, doi: 10.5455/jjee.204-1703066445, available online: September 25, 2024.
[36] H. Lee, S. Kim, "Temporal efficiency in load forecasting using LSTM networks: a comparative study with CNN models," *Journal of Smart City Energy Management*, vol. 12, no. 1, pp. 45-58, 2024.
[37] G. Vítor, P. Rito, S. Sargento, F. Pinto, ''A scalable approach for smart city data platform: support of real-time processing and data sharing,'' *Computer Networks*, vol. 123, p. 109027, 2022, doi: 10.1016/j.comnet.2022.109027.
[38] J. Lee, S. Park, "Flexible GUI applications for energy management in diverse urban infrastructures," *International Journal of Smart City Applications*, vol. 17, no. 3, 210-225, 2023.