Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 7, 2025

SUSTAINABLE ENERGY MANAGEMENT IN SMART GRIDS USING MACHINE LEARNING

Abdur Rahman Khan[1], Ihsan Ul Haq[2], Muhammad Farooq[3], Bilal Ur Rehman[*4], Kifayat Ullah[5], Muhammad Amir[6], Muhammad Kashif Khan[7], Muhammad Iftikhar Khan[8]

[1,3, *4,5,6,7,8]*Department of Electrical Engineering, University of Engineering and Technology, Peshawar, KPK, Pakistan*
[2]*Energy Engineering, University of Hull, United Kingdom*

DOI: https://doi.org/10.5281/zenodo.16720300

Keywords
Smart Grid, Renewable Energy Forecasting, LSTM, Sustainable Energy, Energy Management, Time-Series Prediction

Article History
Received: 30 April, 2025
Accepted: 16 June, 2025
Published: 31 July, 2025

Corresponding Author: *
Bilal Ur Rehman

Abstract
This research paper investigates an approach for sustainable energy management in smart grids using machine learning by forecasting renewable energy generation using Long Short-Term Memory (LSTM) networks. An energy management system is developed that collects real-world time-series data from renewable sources (solar and hydro) and then applies data pre-processing techniques, including dropout regularization and normalization, to build an LSTM model that predicts generation trends. To validate the accuracy and sustainability impact of the model, performance metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) are determined. This specific model enables switching between renewable sources and conventional grid supply, which is based on predicted generation demand. The results indicate enhanced resource utilization, reduced dependency on fossil fuels, and support for smart grid automation.

INTRODUCTION

There has been a significant shift in global energy consumption, driven by growing energy demands and an increasing need for environmental sustainability. Due to concerns about carbon emissions and the finite resources of fossil fuels, there has been a growing concern that has led to a movement towards cleaner, renewable sources of energy, such as solar, hydro, wind, and biomass [1]. Since renewable technologies are more economical and environmentally friendly, they help meet both climate goals and energy security [2]. Smart grids offer a promising solution for handling multiple renewable energy sources while maintaining reliability, efficiency, and resilience [3]. They have emerged as the backbone of modern energy systems by integrating advanced automation, bidirectional communication, real-time monitoring, and decentralized control. Smart grids have enabled the strategic balancing of supply and demand, facilitating consumers' stated needs through active participation in demand response programs and the integration of distributed energy resources [4]. However, despite these advancements, efficiently managing renewable energy through smart grids remains a challenging task due to the variable nature of sources such as solar and hydropower [5]. To address these issues, a machine learning data-driven technique is employed to predict generation patterns and inform real-time operational decisions [6]. Long Short-Term Memory (LSTM) networks are among the techniques; they are a specialized form of Recurrent Neural Network (RNN) that has shown remarkable success, particularly in time-series forecasting, due to their ability to capture temporal dynamics and long-term dependencies [7].

This research study proposes a practical approach to utilizing a robust LSTM-based framework for forecasting renewable energy generation, which enables dynamic switching between renewable and conventional power sources based on scenarios of supply and demand. This model supports intelligent energy transfer by accurately predicting renewable energy availability, resulting in grid stability, optimized resource utilization, and reduced dependency on fossil fuels [8]. This research provides practical insights into the implementation of machine learning for real-time, sustainable energy management, building upon a growing body of literature in smart grids [9].

LITERATURE REVIEW

There has been a significant transformation in the modern electric grid, from a fossil-fuel-based infrastructure to a more sophisticated, sustainable, and intelligent framework. This framework is also known as the smart grid, which utilizes advanced communication, real-time control technologies, and automation. Numerous researchers have employed data-driven methods, particularly machine learning, to enhance forecasting and ensure efficient energy management. This is because the integration of renewable energy sources, particularly solar and wind, which are naturally variable, is complex; therefore, managing such energy generation and consumption can become increasingly complex [10].

A one-way energy flow model was used in traditional power systems, where energy flowed from generators to consumers. The bi-directional energy flows became necessary with the introduction of renewable energy sources, electric vehicles and distributed generation [11]. To facilitate the optimal integration of green energy, avoid grid instability, and enable accurate and efficient load scheduling, it is critical to have an accurate forecast of both supply and demand [12]. Auto-Regressive Integrated Moving Average (ARIMA), linear regression and exponential smoothing were grounded in statistical techniques. Although they were effective in capturing simple trends, these models struggled with nonlinear and stochastic behaviour that is quite common in modern energy systems [13]. Unlike conventional models, machine learning has emerged as a transformative tool in innovative grid applications. It can capture nonlinear patterns from large datasets and adapt to changing input dynamics [14].

Artificial Neural Networks (ANNs) were developed by Awais et al., who created a short-term load forecasting model that outperformed linear regression in modelling daily and hourly consumption variations [15]. Furthermore, Ahmed et al. proposed a hybrid machine learning-based approach that utilized Support Vector Machines (SVM) and Decision Trees for wind energy forecasting, highlighting how learning techniques can enhance forecasting accuracy exponentially under variable meteorological conditions [16].

Random Forest (RF), Gradient Boosting Machines (GBM), and k-Nearest Neighbours (kNN) were among the other models that gained traction. Zhang et al. evaluated the performance of tree-based models, such as Random Forest and Gradient Boosting Machines. He utilized smart meter data and observed superior results compared to linear baselines, especially when working with high-dimensional feature sets [17].

Deep learning (DL) models, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, have become essential for time-series data, including energy generation and demand [18]. LSTM networks, with their memory cell and gated mechanism, can be adapted to learn temporal dependencies over long sequences. Khan et al. achieved a significant reduction in Mean Squared Error (MSE) compared to SVM and ANN baselines when they implemented an LSTM model for solar power forecasting [19]. A bi-directional LSTM (Bi-LSTM) was proposed by Liu et al. for short-term load forecasting, enabling the model to access both past and future contexts. Resulting in more robust peak demand predictions [20]. CNNs are used to extract local temporal features, and the LSTM models extract sequential dependencies. The hybrid deep learning models have been developed by combining CNNs and LSTMs. When dealing with multivariate inputs, such as combined historical load, temperature, humidity, and solar irradiance data, these models are very effective.

Weather conditions have a significant impact on energy consumption and renewable energy generation. The performance of solar panels and wind turbines is greatly affected by solar irradiance, temperature, wind speed, and humidity. Therefore, it is crucial to integrate meteorological data into forecasting models to enhance their predictive capabilities [21].

Morales et al. addressed these challenges by using Quantile Regression Neural Networks (QRNN) to create forecasts of solar output based on probability. QRNN models provide prediction intervals rather than producing a single-point estimate, offering valuable insights into forecasting uncertainty [22]. Moreover, spatial forecasting methods that integrate Geographic Information Systems (GIS) are employed to enhance the model's applicability across different regions.

This research presents a comprehensive proposal and implementation of a machine learning-based framework for sustainable energy management in smart grids, focusing on the prediction and optimization of power generation from diverse energy sources. The LSTM model aligns closely with real energy generation profiles to accurately predict and capture daily trends in the predicted 24-hour generation.

METHODOLOGY

A. Data Pre-processing

Data pre-processing is one of the most crucial phases in any data-driven system, particularly in time-series forecasting, where noise, temporal integrity, scale, and completeness have a significant impact on model performance. Careful pre-processing was required to clean and structure the data effectively for this research, which integrates both energy consumption (load) and meteorological data for use with deep learning models, such as LSTM. In most cases, real-world datasets are rarely complete due to sensor faults, which create gaps in load consumption records. Weather datasets retrieved from sources like NOAA may contain missing entries due to poor signal quality or inadequate equipment maintenance. A forecasting model requires synchronized load and weather inputs. Load data is recorded on an hourly basis, whereas the weather data is available at varying intervals (hourly or every 3 hours, depending on the station and parameter). To merge these datasets, Resampling is used to standardize all features to an hourly frequency, and the weather data is interpolated to match the timestamps of the load dataset. Once aligned, these features are merged using inner joins on the timestamp columns; as a result, a dataset is created that preserves time ordering and consistency required for sequential modeling.

To improve model performance, the calendar features are used to encode the day of the week and the hour of the day, thereby pinpointing whether the day was a weekend or a holiday. These variables helped capture periodic trends and anomalies in load consumption. For the average, rolling statistics were used, along with tolling standard deviations for variables such as load and temperature, over intervals of 3, 6, 12, and 24 hours. These features gave short-term trend information to the model. Interaction features were utilized where some features were synthesized based on domain knowledge, such as interaction terms between wind speed and time of day, to enhance the photovoltaic generation patterns.

B. Data Splitting

For time-series problems, one of the most important components is data splitting, as time-series data must preserve chronological order to simulate real-world forecasting conditions, unlike random splits that are commonly used in traditional machine learning. This research highlights the strategy employed to divide the dataset into training, validation, and test sets. The datasets were divided into three non-overlapping chronological segments because LSTM models learn from temporal dependencies; therefore, shuffling the data before splitting would result in data leakage.

Since we had a temporal dependency between observations, a standard k-fold cross-validation technique could not be used on time-series data; therefore, a more suitable approach was to use a walk-forward validation or expanding window approach. Therefore, we simulated the effects of full walk-forward validation by repeating training with different cut-off dates and verifying consistency in performance. The dataset is resampled depending on the forecasting objectives (e.g., one hour ahead, 24 hours ahead). For 1-hour ahead forecasting, the sequences of 24 hours predicted the next hour. A larger period (72 hours) was used to predict hourly targets for the next day. The splitting strategy remained constant, even though each configuration resulted in a different input-output mapping. The line plots of load and weather features were generated for each subset to ensure uniform distribution.

C. Data Cleaning

To build a robust machine learning model, it is vital to perform effective data cleaning, especially in domains such as time-series forecasting, like renewable resource prediction and energy consumption. In this study, we combined environmental parameters from NOAA and load consumption data from ERCOT. These datasets had various inconsistencies, although they were rich in information; for instance, they contained missing values and outliers. Therefore, a comprehensive data cleaning process was carried out to ensure high-quality inputs for our LSTM-based forecasting model in several phases.

Missing and null values was one of the most common issues faced during dataset analysis as they appeared in both the environmental data (humidity, temperature, wind speed) and the load consumption data (mostly in hourly readings). These missing values would disrupt the sequential integrity of the data and create significant problems in the learning process if left untreated. To locate and handle the missing values, we used detection, forward fill, backward fill, and interpolation. Furthermore, a column or row was removed if it had more than 30% missing values in the sequence to ensure minimal distortion of data quality. Therefore, by applying these methods, we ensured that the continuity of the dataset was preserved without compromising its temporal structure.

D. Handling Duplicates, Timestamp Normalization, and Outlier Detection

In the load data, we encountered certain hours that were logged more than once due to a system error, which was identified as duplicate timestamps. Duplicates can distort sequential learning, which can be detrimental to a time-series model that is sensitive to chronological order. To counter this, we used the 'drop duplicates' function (subset='timestamp', keep='first'). We retained the first instance or averaged the timestamp when its duplicates had varying values, depending on the consistency of neighboring entries. To align the weather data (from NOAA) and load data (from ERCOT), both required consistent timestamps. To achieve this, we had to address two significant challenges: different time zones (some in UTC, others in local time) and various date and time formats (e.g., MM/DD/YYYY HH: MM vs. ISO 8601). To counter this, we standardized all timestamps to UTC and converted them into datetime objects. This allowed us to merge datasets seamlessly, resulting in the accurate generation of lag features. In the raw data, some features were represented as the day of the week and the hour of the day. For compatibility with the downstream model input, these were converted to integer types or encoded using Label Encoder. For example, days were considered as integers from 0 (Monday) to 6 (Sunday). Categorical weather conditions were one-hot encoded, which helped reduce model complexity and memory usage.

E. Normalization and Scaling

Neural networks are sensitive to the scale of input features. LSTM models contain variables with diverse units, so normalization is essential. For this reason, Min-Max Scaling was applied to rescale all numeric features into the [0, 1] range. Training data was scaled using scalers (e.g., MinMaxScaler) to prevent data leakage, and the same transformation was applied to the validation/test data. This improved model convergence, reduced training time, and ensured numerical stability. LSTM models require 3D input (e.g., samples, time steps, features) to build their structure. Each sample has the previous 24 hours of load, calendar indicators, and corresponding weather variables. These steps allowed the pre-processed tabular dataset to be transformed into a supervised learning format, which is required for sequential models.

SIMULATION RESULTS

Dataset Selection:

Table 1: Dataset showing the multiple means of power production

For smart grid forecasting, where temporal accuracy, environmental factors, and load variability play significant roles, it is crucial to ensure the highest quality of input data. This research focused on a real-world dataset, as shown in Table 1, spanning the period from 1 January 2020 to 31 December 2022. Generation is provided for both by production type (natural gas, geothermal, solar, etc.) and in total.

A. Visual Imbalance and Feature Suppression in un-normalized Data

Table 2 highlights a multi-variant time series visualization where each feature is plotted over time. This plot exhibits overlapping and flat-lining because the values remain compressed near the bottom of the graph, relative to a more dominant signal. There is also a visual imbalance, which highlights how un-normalised data can mislead both visual analytics and machine learning model during training. Therefore, if this dataset is fed to an LSTM model, it could cause the optimization process to favor features with higher magnitudes, and as a result, we could experience slower convergence and potentially higher training error.

Table 2: Dataset before Normalization

Date_Time	Total (MWh)	Natural Gas	Dammed Hydro	Lignite	River	Import Coal	Wind	Solar	Fuel Oil	Geothermal	Asphaltite Coal	Black Coal	Biomass	Naphta	LNG	Import-Export	Waste Heat	Consumption (MWh)
01.01.2020 00:00	28125.46	6166.4	3761.62	5780.09	1545.26	7302.54	770.5	0	35.5	1157.82	293.65	572.31	396.65	0	0	278	65.12	28125.46
01.01.2020 01:00	26548.41	5161.03	3079.97	5792.66	1548.51	7286.46	776.18	0	35.8	1164.57	273.78	575.65	392.46	0	0	394	67.34	26548.41
01.01.2020 02:00	25287.99	5149.52	2160.79	5755.22	1460.39	7281.67	787.75	0	36.3	1167.74	273.78	585.67	390.22	0	0	172	66.94	25287.99
01.01.2020 03:00	24311.53	3860.16	2292.47	5793.94	1591.92	7299.57	815.19	0	36.6	1168.91	271.57	569.56	392.99	0	0	154	64.65	24311.53
01.01.2020 04:00	23730.48	3437.43	2470.96	5626.99	1653.69	7299.74	939.26	0	33	1167.93	273.78	374.9	401.09	0	0	-13	64.71	23730.48
01.01.2020 05:00	23535.52	3343.25	2394.24	5449.72	1624.99	7299.25	1185.05	0	36	1170.88	271.57	264.53	420.11	0	0	8	67.93	23535.52
01.01.2020 06:00	23774.24	3398.91	2532.43	5195.1	1634.98	7297.46	1337.51	0	35.6	1163.9	271.57	269.99	423.99	0	0	140	72.8	23774.24
01.01.2020 07:00	24112	3608.24	2445.22	5257.4	1654.75	7249.22	1477.6	0.06	35.7	1162.54	275.99	295.06	423.07	0	0	154	73.15	24112
01.01.2020 08:00	24196.77	3284.36	2646.02	5263.06	1665.65	7196.76	1788.91	1.98	34.4	1164.08	273.78	280.02	425.79	0	0	99	72.96	24196.77

B. Enhanced Visibility and Pattern Recognition through Normalization

According to Table 3, all features are now scaled into the [0, 1] interval, and the visual structure of the time series becomes more interpretable, allowing patterns and periodic trends between variables to be more easily discerned. Most importantly, there is no feature compression, and the range of variability for each signal is preserved. Through normalization, we achieved a uniform contribution of features during training and faster convergence of gradients, which resulted in stable learning dynamics in the LSTM model. There is better detection of multi-variant dependencies in models that rely on a weighted combination of inputs, and we have enhanced forecasting accuracy as the network can more effectively learn temporal patterns without scale bias.

Typically, the dataset is split in a ratio of around 70% for training and 30% for testing.

Table 3: Dataset after Normalization

Date_Time	Total (MWh)	Natural Gas	Dammed Hydro	Lignite	River	Import Coal	Wind	Solar	Fuel Oil	Geothermal	Asphaltite Coal	Black Coal	Biomass	Naphta	LNG	Import-Export	Waste Heat	Consumption (MWh)
01.01.2020 00:00	0.355808334	0.299084401	0.222180827	0.851908437	0.202280284	0.69111864	0.068648386	8.53E-05	0.335539	0.71248955	0.796403775	0.771003845	0.042039	0	0	0.62906059	0.319352385	0.317883033
01.01.2020 01:00	0.311943378	0.246050138	0.178471557	0.85489048	0.202849043	0.685367481	0.069266585	8.53E-05	0.338374	0.722199	0.742514645	0.777361148	0.035508	0	0	0.651302605	0.343474954	0.278693605
01.01.2020 02:00	0.276885348	0.245442974	0.129531211	0.846008403	0.187427811	0.688845836	0.070525839	8.53E-05	0.3431	0.726768237	0.742514645	0.796433058	0.032016	0	0	0.585598059	0.339128545	0.247372379
01.01.2020 03:00	0.24972554	0.177427957	0.127974895	0.855194141	0.210445907	0.690795198	0.07351235	8.53E-05	0.345936	0.728454675	0.736520937	0.765769538	0.036334	0	0	0.584594987	0.314245355	0.223107511
01.01.2020 04:00	0.233563888	0.15512853	0.139420164	0.815587762	0.222125819	0.690813712	0.087015862	8.53E-05	0.311909	0.727042103	0.742514645	0.395256767	0.04896	0	0	0.538177603	0.314897316	0.208660516
01.01.2020 05:00	0.228041161	0.150160442	0.134500667	0.773513117	0.216233243	0.690760349	0.113767117	8.53E-05	0.340265	0.731294233	0.736520937	0.18518025	0.078606	0	0	0.54401452	0.349865907	0.203823793
01.01.2020 06:00	0.234783053	0.153096562	0.143361789	0.713128346	0.21798152	0.690565413	0.130360536	8.53E-05	0.336484	0.721233262	0.736520937	0.195572728	0.084654	0	0	0.58070371	0.402803434	0.209755945
01.01.2020 07:00	0.2441757	0.164138927	0.137769644	0.727908086	0.225441322	0.685311936	0.145607631	0.000132	0.337429	0.719272958	0.748508353	0.243290571	0.08322	0	0	0.584584987	0.406606541	0.218349224
01.01.2020 08:00	0.246533541	0.147053936	0.150645492	0.729250836	0.22334885	0.679598889	0.179489942	0.00162	0.325142	0.721492714	0.742514645	0.214663672	0.08746	0	0	0.569307825	0.404541997	0.220255745
01.01.2020 09:00	0.280058267	0.149173397	0.219515345	0.691888699	0.219957299	0.674187583	0.209828493	0.007559	0.302457	0.715237038	0.736520937	0.194202292	0.088925	0	0	0.580815503	0.418015864	0.250207993
01.01.2020 10:00	0.318620542	0.176779115	0.24337821	0.664497548	0.233208498	0.686567587	0.244954003	0.015846	0.362949	0.708217421	0.742514645	0.212284442	0.069659	0	0	0.609054447	0.401934152	0.284659055

C. Box plots

Box plots, also known as box and whisker plots, are a graphical method used for visualizing the distribution, spread, and skewness of continuous numerical data. It provides us with a five-number summary of a dataset, which includes the minimum, first quartile (Q1), second quartile, third quartile (Q3), and maximum. Outliers are also displayed explicitly.

In the following research, box plots were used to understand the underlying distribution of energy consumption across different time frames, including hourly, daily, and monthly, as part of the exploratory data analysis (EDA) phase. These visualizations helped to understand outliers, temporal trends, and variations that are critical for accurate time-series modeling in the smart grid context. For instance, box plots of daily, monthly, and yearly energy consumption are shown in Figures 1, 2, and 3, respectively.

D. Box Plot Visualizations for Temporal Energy Patterns

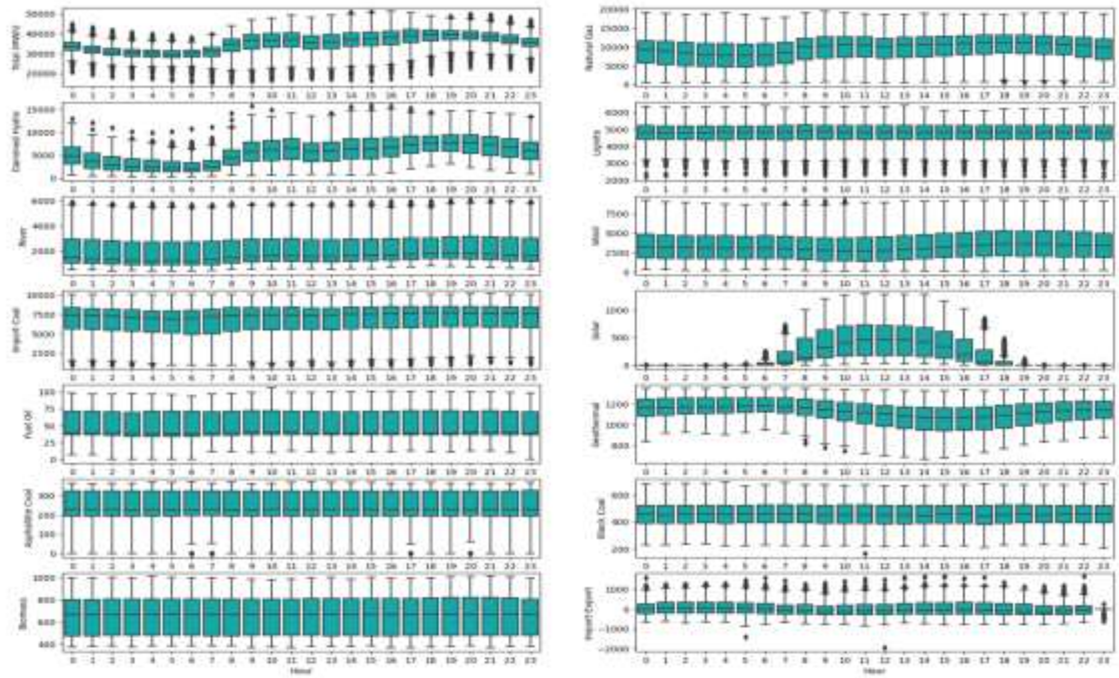

Figure 1: Daily Average Energy Generation

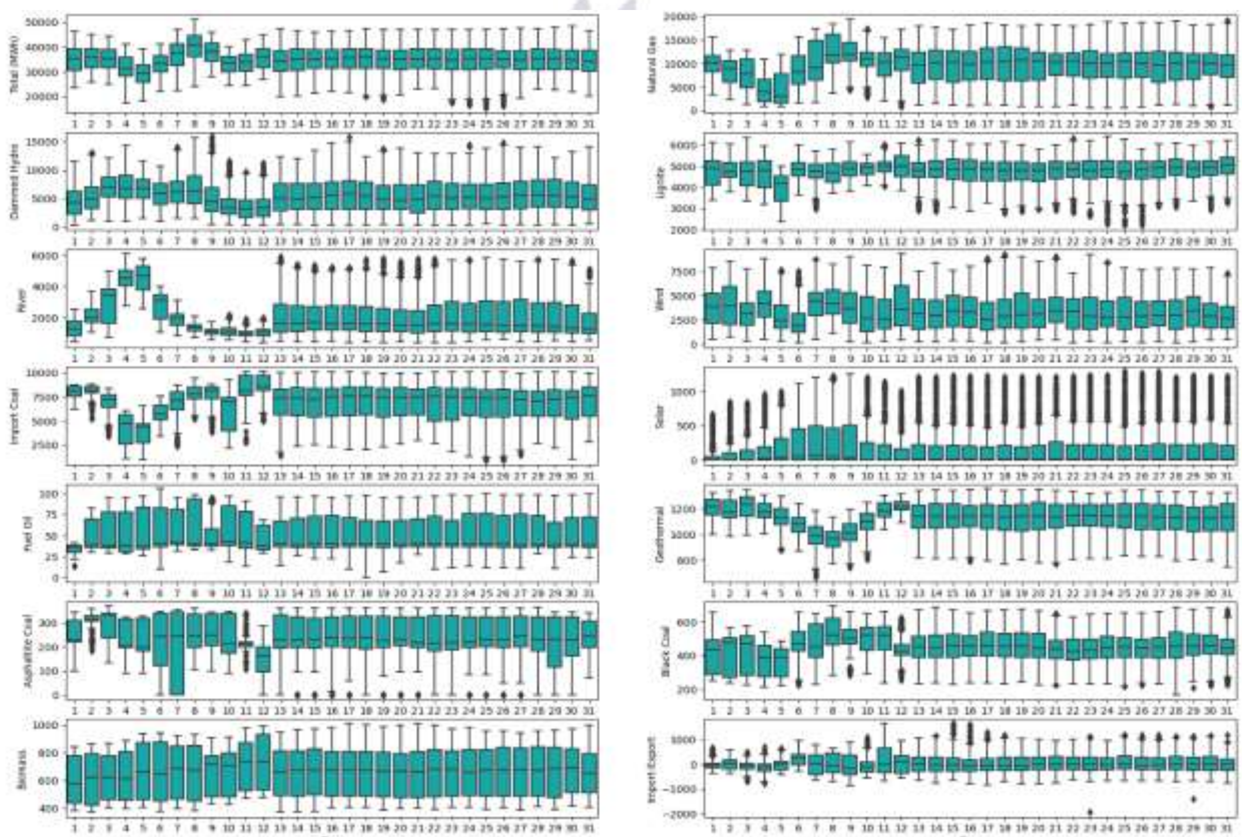

Figure 2: Monthly Average Energy Consumption

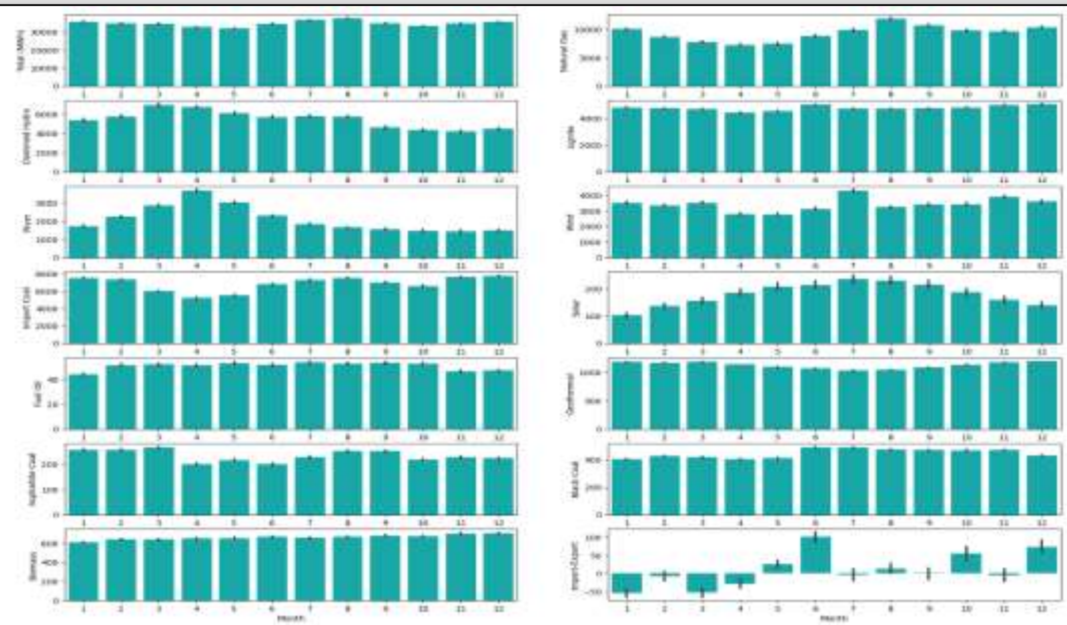

Figure 3: Yearly Average Energy Consumption

OBSERVATIONS

A. Central Tendency of Daily Load

The box plot of daily energy consumption, shown in Figure 1, clearly indicates that the median line is situated around the center of most boxes, representing the typical daily load and serving as a baseline for forecasting models. This validates the presence of short-term volatility, and therefore, it is necessary to use memory-based models like LSTM. In Figure 2, lower median values are observed during the late night and early morning hours (1:00 am to 5:00 am), indicating minimal activity. For Figure 3, the median energy consumption varied significantly across months, indicating a high electricity demand.

B. Daily variability and Interquartile Spread

The interquartile range (IQR) in Figure 1 highlights the spread of daily consumption values. A relatively wide IQR suggested significant day-to-day variability, which could stem from changes in consumer behaviour or weather. Furthermore, the whiskers are also extended from the boxes to show the minimum and maximum daily consumption values within 1.5 × IQR. These whiskers captured normal fluctuations in consumption. According to Figure 2, the highest medians and widest IQRs appeared during business hours, between 7:00 am and 9:00 pm, which relates to typical waking hours and peak load times. There has also been a sharp increase around 7:00 am and 6:00 pm, which suggests that morning start-ups and returning home in the evening are more prevalent in residential areas; however, commercial zones show a flatter pattern during work hours. In the case of Figure 3, there were Peak medians in July and August, as well as in January and December, which reflect increased usage due to cooling and heating loads, respectively. On the other hand, lower consumption medians in April, May, and October suggest milder temperatures and thus less HVAC usage.

C. Detection of Outliers

Several outliers were outside the whiskers, as shown in Figure 1, particularly on holidays, weekends, or during extreme weather conditions. These outliers were necessary to identify, as they could indicate anomalies that needed to be removed for training the model. Therefore, this allowed us to develop robust pre-processing routines to handle daily-level anomalies and ensure the model could generalize across diverse usage situations. According to Figure 2, in the early evenings, we observed extended whiskers and frequent outliers, which highlight unpredictable usage, primarily due to climate-based appliances (e.g., air conditioners during the summer). Figure 3 clearly shows outliers in certain months, likely due to blackouts or unusually hot or cold years.

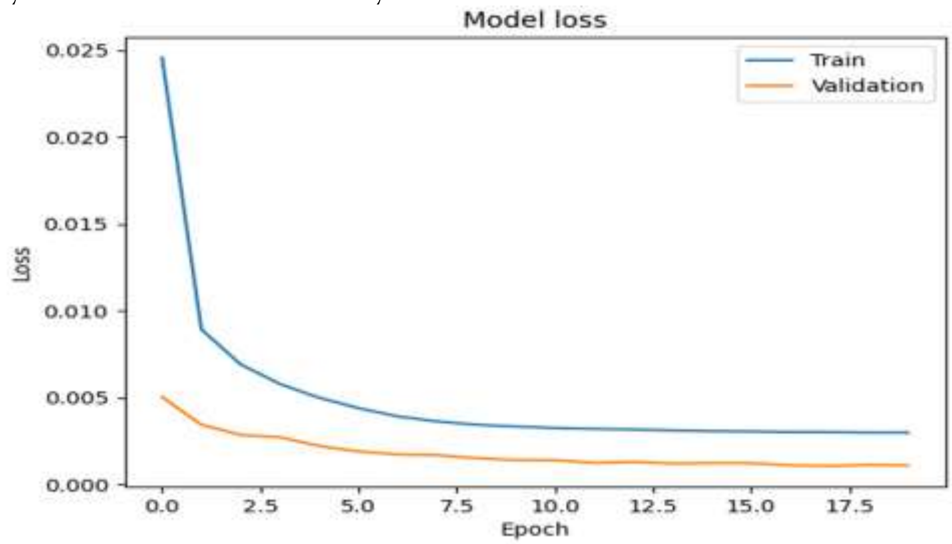

Figure 4: Train validation curve before optimizations

So, we used a multivariate nature of the dataset where each source acts as an input feature, and using both renewable and non-renewable energy sources, we can train the machine learning model to forecast the availability of green energy, determine switching points between sustainable and conventional sources, and optimize grid decisions based on reliability and variability of sources. Before optimization, we can get crucial information through the training validation curve about how the LSTM model behaves when trained without regularization techniques, as shown in Figure 4.

During this phase, the model is trained using loosely configured settings such as higher learning rates and insufficient regularization. Our main goal was to observe how this model learns from past time-series data (e.g., power consumption and generation) and to analyze its generalization capability on unseen validation data. Before optimization, the LSTM model may have had too many hidden units, as LSTM networks can model very fine-grained aspects of the training data, including random noise, which may lead to poor generalization, as seen in Figure 4. In the pre-optimized form, the model uses low dropout rates. Dropout is a regularization method that randomly deactivates neurons during training, making the network learn more robust features. In its absence, the network might experience overfitting. For the overfitting issue, various strategies were employed to enhance the model's ability to generalize. The goal was to achieve convergence between the training and validation losses, thereby maintaining low error on both curves.

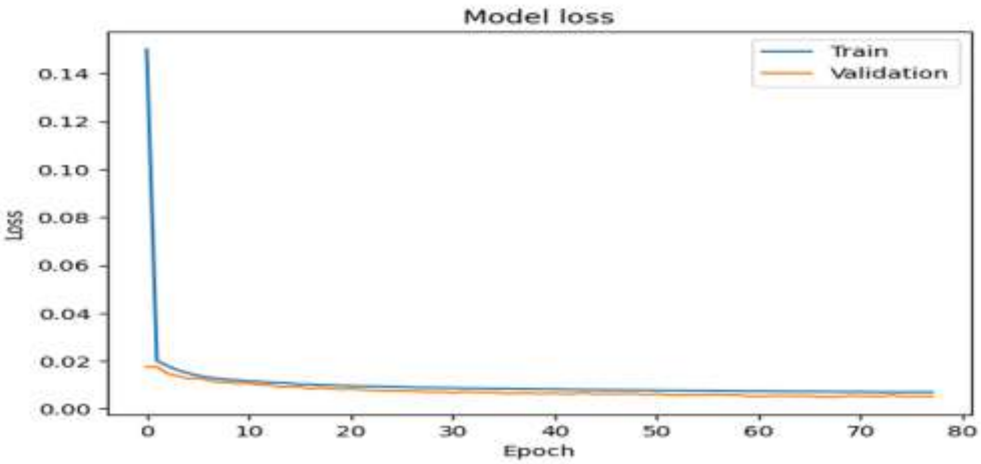

Figure 5: Train validation curve after optimizations (i.e. underfitting)

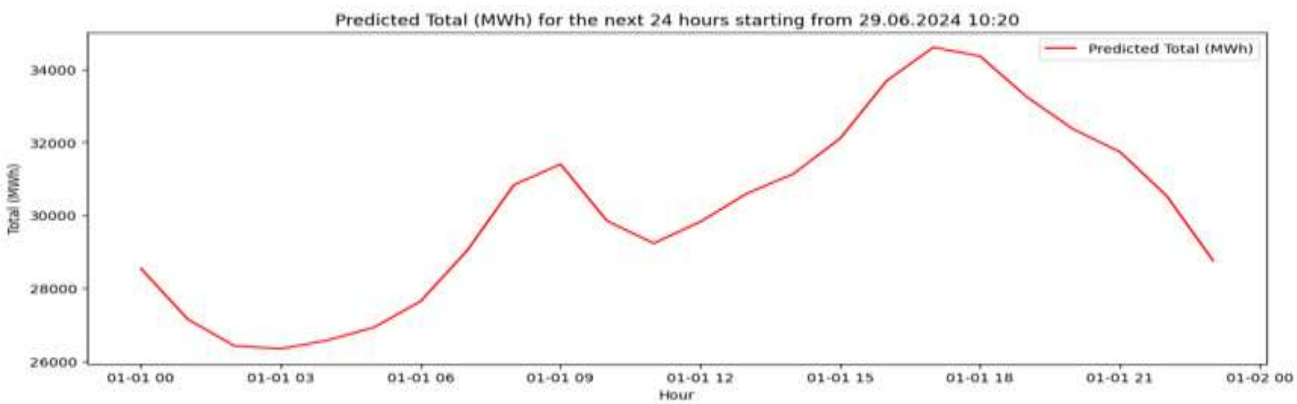

Figure 6: Predicted 24 hours of Total MWh starting from 29.06.24 10:20 AM

After optimization, both the training and validation losses remain close together, with minimal reduction in error, which suggests that the overfitting issue has been resolved, as shown in Figure 5. Figure 6 shows the predicted total MWh for the next 24 hours. To assess model performance, Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) were used, which help detect extreme deviations in energy predictions, express error as a percentage, and interpret performance relative to the actual energy demand. The visualization of predicted versus actual energy generation values shows that the model adapts well to seasonal variations. The simulation results validate the model's potential to forecast demand-supply gaps in real-time and prioritize renewable energy usage when availability is high. During periods of low green energy, it switches to backup (non-renewable) sources, which helps minimize carbon emissions and improve the sustainability index of the grid, supporting autonomous decision-making systems within modern smart grids.

CONCLUSION

This research study presents a comprehensive proposal and implementation of a machine learning-based framework for sustainable energy management in smart grids, focusing on the prediction and optimization of power generation from diverse energy sources. According to the simulation results, the LSTM model aligns closely with real energy generation profiles and accurately captures the predicted 24-hour generation curves, reflecting daily trends. The convergence of training and validation loss curves further validated the reliability of the optimization techniques employed. In the future generation of sustainable power systems, the proposed framework can play an important role. Hence, to further enhance the accuracy of forecasting load inputs, implement the IoT-based sensor networks to capture real-time weather data.

REFERENCES

[1] Khan, R., et al., "Solar energy forecasting using LSTM networks," Applied Energy.
[2] Wang, J., et al., "Review on wind power forecasting methods," Renewable and Sustainable Energy Reviews.
[3] Siano, P., et al., "Machine learning for energy management systems: A review," Electric Power Systems Research.
[4] Fan, S., & Hyndman, R. J., "Short-term load forecasting based on a semi-parametric additive model," IEEE Transactions on Power Systems.
[5] Hong, T., & Fan, S., "Probabilistic electric load forecasting: A tutorial review," International Journal of Forecasting.
[6] He, M., & Yan, J., "Short-term electric load forecasting using deep neural networks," Journal of Modern Power Systems and Clean Energy.
[7] Kong, W., et al., "Short-term residential load forecasting based on LSTM recurrent neural network," IEEE Transactions on Smart Grid.
[8] Marino, D. L., et al., "Building energy load forecasting using deep neural networks," IECON.
[9] Yang, F., et al., "Weather-aware forecasting for smart grid using gradient boosting," Energy Systems.
[10] Ribeiro, M. T., et al., "Model interpretability for black-box energy forecasting," Neural Computing and Applications.
[11] Zhang, G., et al., "Forecasting with artificial neural networks: The state of the art," International Journal of Forecasting.
[12] Paraschiv, F., et al., "Electricity price forecasting using autoregressive models," Energy Economics.
[13] Morales, J. M., et al., "Probabilistic forecasting of renewable energy generation," IEEE Transactions on Sustainable Energy.
[14] Dong, X., et al., "Wind power forecasting using extreme learning machine," Renewable Energy.
[15] Awais, M., et al., "Short-term load forecasting using artificial neural networks in smart grids," Journal of Energy Engineering.
[16] Ahmed, S., et al., "Hybrid SVM and decision tree models for wind energy prediction," Renewable Energy.
[17] Zhang, L., et al., "Tree-based ensemble methods for energy forecasting," IEEE Transactions on Smart Grid.
[18] Shi, Y., et al., "Deep learning for household load forecasting—A novel pooling deep RNN," IEEE Transactions on Smart Grid.
[19] Khan, M. J., et al., "Improving solar generation forecasts using deep LSTM networks," Energy Reports.
[20] Liu, Y., et al., "Bi-directional LSTM for electricity demand forecasting," Energy Reports.
[21] Rolnick, D., et al., "Tackling climate change with machine learning," ACM Computing Surveys.
[22] Ma, T., et al., "Ensemble learning for energy consumption prediction," IEEE Access.