BMBAD BJESR Mühendislik Bilimleri ve Araştırmaları Dergisi
Journal of Engineering Sciences and Researches

Artificial Intelligence and Optimization Based Output Power Forecasting in Power Plants

Enerji Santrallerinde Yapay Zekâ ve Optimizasyon Temelli Çıkış Gücü Kestirimi

[1]Murat TEKİN, [2]Serhat Berat EFE

[1,2]*Bandırma Onyedi Eylül Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Elektrik Mühendisliği Bölümü Bandırma/Balıkesir, Türkiye*

[1]mtekin496@gmail.com, [2]sefe@bandirma.edu.tr

Araştırma Makalesi/Research Article

ARTICLE INFO

Article history

Received : 5 May 2025
Accepted : 4 July 2025

Keywords:

LSTM, PSO, Power Plants, Power Forecasting

ABSTRACT

The global demand for energy has been rapidly increasing due to industrialization and population growth. This surge necessitates making energy production processes more efficient, sustainable, and predictable. In this context, forecasting models based on artificial intelligence and heuristic optimization techniques have become a crucial component of decision support systems in the energy sector. In this study, a forecasting model based on Particle Swarm Optimization (PSO) was developed by optimizing the hyperparameters of a Long Short-Term Memory (LSTM) network using PSO. Operational data from a power plant was used during the training and testing phases. The model's performance was evaluated using statistical error metrics such as the coefficient of determination (R^2), root mean square error (RMSE), mean squared error (MSE), and mean absolute error (MAE). The results indicate that the proposed PSO-based optimization approach provides high accuracy in energy production forecasting and offers a strong alternative to traditional methods.

MAKALE BİLGİSİ

Makale Tarihleri

Gönderim : 5 Mayıs 2025
Kabul : 4 Temmuz 2025

Anahtar Kelimeler:

LSTM, PSO, Enerji Santralleri, Güç Kestirimi

ÖZET

Sanayileşme ve nüfus artışına bağlı olarak enerjiye olan küresel talep hızla artmaktadır. Bu artış, enerji üretim süreçlerinin daha verimli, sürdürülebilir ve öngörülebilir hale getirilmesini zorunlu kılmaktadır. Bu bağlamda, yapay zekâ ve sezgisel optimizasyon tekniklerine dayalı tahmin modelleri, enerji sektöründeki karar destek sistemlerinin önemli bir bileşeni haline gelmiştir. Bu çalışmada, Uzun Kısa Süreli Bellek (LSTM) modelinin hiperparametreleri Parçacık Sürü Optimizasyonu (PSO) algoritmasıyla optimize edilerek PSO tabanlı bir tahmin modeli geliştirilmiştir. Modelin eğitim ve test aşamalarında, bir enerji santraline ait operasyonel veriler kullanılmıştır. Model performansı, belirleme katsayısı (R^2), kök ortalama kare hata (RMSE), ortalama kare hata (MSE) ve ortalama mutlak hata (MAE) gibi istatistiksel hata metrikleriyle değerlendirilmiştir. Sonuçlar, önerilen PSO tabanlı optimizasyon yaklaşımının enerji üretim tahmininde yüksek doğruluk sunduğunu ve geleneksel yöntemlere kıyasla güçlü bir alternatif oluşturduğunu ortaya koymaktadır.

ORCID ID: [1]0000-0002-8447-7847
[2]0000-0001-6076-4166

1. INTRODUCTION

Today, the rapid increase in global energy demand, driven by population growth and industrial development, presents significant challenges in terms of not only energy supply but also its sustainability, efficiency, and predictability. Power plants remain at the core of energy infrastructure due to their ability to deliver large-scale, uninterrupted electricity generation. However, their operational performance is highly sensitive to environmental parameters such as atmospheric pressure, ambient temperature, relative humidity, and condenser vacuum. These variables directly influence turbine behavior and subsequently affect net power output. Therefore, accurately modeling the impact of environmental conditions on plant performance is crucial for improving the reliability and operational forecasting of energy systems [1], [2].

The accurate forecasting of net power generation in energy plants is not only essential for production scheduling but also constitutes a critical factor in maintaining grid stability, managing supply-demand equilibrium, and optimizing operational expenditures. Nevertheless, the underlying dynamics of power generation are inherently complex, driven by nonlinear interactions among environmental parameters that evolve over time. As such, conventional statistical or regression-based models often fall short in capturing these intricacies [3]. In recent years, artificial intelligence techniques—particularly Long Short-Term Memory (LSTM) neural networks—have gained considerable traction in energy forecasting tasks due to their capacity to model sequential data and learn temporal dependencies [4], [5]. Unlike traditional RNNs, the LSTM architecture incorporates memory gates that enable the model to retain long-term temporal patterns, thereby improving its ability to reflect real-world operational fluctuations in predictive modeling.

Although LSTM architectures are highly effective in modeling time series data, the overall performance of these models heavily depends on the proper configuration of hyperparameters. Key parameters such as the number of layers, the number of units per layer, learning rate, activation function, dropout rate, and batch size directly influence the model's learning capacity and generalization ability [4].

If these parameters are selected arbitrarily or through trial-and-error methods, the model may suffer from overfitting or underfitting, significantly reducing its predictive performance [6].

To overcome these challenges, heuristic optimization algorithms have become increasingly popular for hyperparameter tuning. In this regard, Particle Swarm Optimization (PSO) has emerged as a powerful technique for improving the training process of artificial neural networks [7].

Studies in the literature report that LSTM models optimized with PSO achieve lower error rates, faster convergence, and more stable prediction outputs compared to those using conventional tuning strategies [6], [8].

Moreover, it has been shown that PSO not only enhances accuracy but also improves computational efficiency by reducing training time when integrated with LSTM models [3].

In the literature, numerous studies have demonstrated the effectiveness of LSTM models optimized with Particle Swarm Optimization (PSO) in energy forecasting and other complex predictive tasks.

For example, in [3], a PSO-LSTM model was proposed for short-term load forecasting, where various activation functions and network depths were explored to improve prediction accuracy. In [8], a similar PSO-supported LSTM approach was employed for solar irradiance forecasting, with results showing higher R^2 and lower MSE compared to standard LSTM, particularly after applying normalization techniques. In [5], a multi-step prediction model for short-term electrical load was introduced, where PSO improved the performance stability in long-range forecasting, reducing cumulative error propagation significantly.

In [9], a deep learning-based electric load forecasting model was implemented, in which PSO was used to optimize hyperparameters, leading to reduced computational cost and more stable outputs. The structure in [9] is comparable to that in [3] though it focuses more on training efficiency and runtime optimization.

In [6], PSO was combined with attention-enhanced LSTM architectures, resulting in improved training performance and lower validation loss, and demonstrating the synergy between attention mechanisms and heuristic optimization.

Two other studies, similar to the present work, focused on predicting the power output of a power plant using the same target variable, but applied different modeling techniques.

In [10], the performance of several regression-based machine learning models—including linear regression, ridge regression, lasso, elastic net, random forest, and gradient boosting was evaluated in forecasting hourly electrical power output from a combined cycle power plant. Input features included ambient temperature (AT), relative humidity (RH), atmospheric pressure (AP), and exhaust vacuum (V). The results in [10] indicated that even relatively simple models can achieve acceptable accuracy when applied to clean and well-structured datasets.

In [11], a novel metaheuristic optimization algorithm, the Waterwheel Plant Algorithm (WWPA), was proposed and integrated with Recurrent Neural Networks (RNN) to improve net power prediction accuracy in combined cycle power plants. The model was trained using key operational parameters such as ambient temperature, relative humidity, atmospheric pressure, and exhaust vacuum. The results revealed that optimized neural networks can effectively capture nonlinear dependencies in complex energy systems.

Applications outside of energy systems have also benefited from the PSO-LSTM framework. For instance, in [12], PSO-optimized deep learning networks were applied to air pollution forecasting, achieving high accuracy across multiple urban locations. In [13], an LSTM-based prediction model was optimized using Swarm Optimization

(PSO) for short-term ship motion forecasting. The study emphasized the limitations of traditional gradient-based optimizers in escaping local minima, especially in nonlinear and noisy time series data. The hybrid LSTM-PSO model demonstrated superior performance in terms of prediction accuracy and mean squared error (MSE) compared to standard optimization techniques.
Finally, in [14], the PSO-LSTM approach was employed for wind power bidding optimization, demonstrating improved decision-making efficiency and highlighting the model's flexibility beyond traditional forecasting tasks. Although the studies vary in application domain and input configurations, they collectively emphasize the robustness of PSO in enhancing LSTM-based forecasting models. In contrast, the current study utilizes real-time environmental variables from a thermal power plant (AT, RH, AP, V). It evaluates prediction performance not only through traditional error metrics but also via a detailed analysis of training-validation loss curves.

2. MATERIAL-METHOD

In this section, the data preparation process, the methods employed, and the error metrics used to evaluate their performance are discussed.

2.1. Dataset

In this study, the net power generation (P) [MW] was selected as the target output variable to be predicted based on data obtained from a power plant. The original data used in the study are openly available in the UCI Machine Learning Repository. The input variables include ambient temperature (T) [°C], condenser vacuum (V) [cm Hg], relative humidity (RH) [%], and ambient pressure (AP) [Mbar]. Data preprocessing is a critical initial step in developing and training the models. First, all variables were consolidated and organized into a single Excel file in the appropriate format. Subsequently, the dataset was processed using the Python programming language and prepared for modeling. The data were then split into training (80%) and testing (20%) sets to evaluate model accuracy.
In power generation systems, external environmental variables have a direct impact on plant performance. In particular, factors such as ambient temperature, humidity, and pressure significantly influence turbine efficiency and, consequently, net power output. In this study, optimization techniques were employed to model these variables and improve the accuracy of net power prediction. To enhance the model's predictive performance, the data were normalized using the StandardScaler method, minimizing the impact of varying scales among input variables. The dataset used in this study comprises parameters critical to making optimal operational decisions in power plants. Therefore, it is anticipated that the model outputs will play a vital role in energy production forecasting and management [1], [8], [15].
The general flow diagram illustrating the applied method is presented in Figure 2.1.

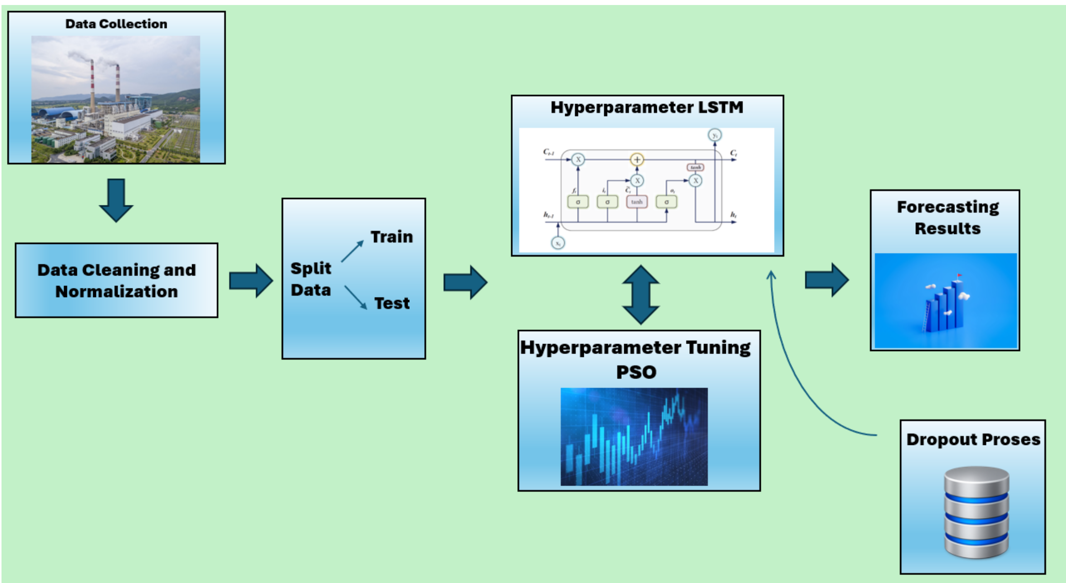

Figure 2.1. Flow diagram.

2.2. Principle of the Power plant and the Impact of Data Variables on the Process

The general energy conversion processes of a conventional thermal power plant are schematically illustrated as follows. The system begins with the combustion of fuel in the boiler, where the resulting thermal energy converts water into steam. This high-pressure steam is then directed to the steam turbine. Within the turbine, the steam energy is transformed into mechanical energy, which is subsequently converted into net electrical power through a generator. The low-pressure steam exiting the turbine is condensed back into liquid form in the condenser unit,

allowing the cycle to restart. This process operates based on the principles of the Rankine cycle, and the overall performance of the system varies significantly depending on environmental conditions [2], [16], [17].
In the dataset used for this study, environmental parameters such as ambient temperature (AT, °C), condenser vacuum (V, cm Hg), atmospheric pressure (AP, Mbar), and relative humidity (RH, %) were employed as input variables for the model [1], [17], [18], [19].

- AT variable (ambient temperature) indirectly influences heat losses in the boiler and the efficiency of the combustion process.
- V variable (condenser vacuum) affects mechanical work output by determining the pressure difference between the turbine outlet and the condenser.
- AP (atmospheric pressure) and RH (relative humidity) variables are critical for the condensation capacity of the condenser and the performance of the cooling systems.

Net power output (PE, MW) is the target variable in the dataset, representing the overall performance of the power plant resulting from the combined effect of the four environmental parameters. This flow structure clearly illustrates the relationship established between physical processes and data variables in data-driven modeling and forecasting applications [1], [17], [18], [19].
The overall energy conversion process and the operational role of the variables mentioned above are illustrated in Figure 2.2 and summarized in Table 2.1, respectively.

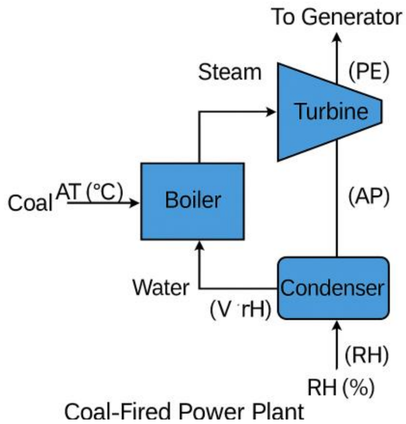

Figure 2.2. Power plant modeling.

Table 2.1. Operational role of environmental and process variables in power generation.

Variable	Location/Stage	Affected Proses
AT	Boiler Input	Efficiency of combustion and steam generation
V	Turbine output- Condenser	Turbine output power
AP	Cooling Air	Condensation efficiency
RH	Cooling Air	Condenser performance
PE	Generator Output	Final electricity generation (predicted value)

2.3. Methodology Employed

In this study, a forecasting model based on Particle Swarm Optimization (PSO) was developed to enhance the accuracy of power generation prediction. Within this framework, a Long Short-Term Memory (LSTM) neural network was employed as the prediction tool, and its hyperparameters were optimized using the PSO algorithm. PSO played a central role by systematically tuning critical hyperparameters such as learning rate, batch size, number of epochs, and dropout rate, thereby improving the model's overall performance. The results clearly demonstrate that the PSO-based optimization approach, combined with the temporal modeling capabilities of LSTM architecture, provides an effective and reliable method for forecasting energy production in power plants.

2.3.1. Particle Swarm Optimization (PSO)

The Particle Swarm Optimization (PSO) algorithm was developed by Kennedy and Eberhart in 1995. This algorithm is a stochastic optimization technique based on the principle of swarm intelligence and has yielded successful results in a wide range of engineering and scientific problems. In PSO, each potential solution is represented as a member of a population of individuals called particles. These particles continuously update their positions within the solution space in pursuit of the optimal result. During this process, they determine their direction by drawing on both their own best experiences and the collective experience of the swarm.
In the PSO algorithm, the position and velocity of the particles are the fundamental components of the optimization process. The position of a particle represents a solution point in the search space, while its velocity determines the

direction and magnitude of its movement in the next iteration. The updates of particle positions and velocities are carried out according to specific mathematical rules, which incorporate both stochastic and deterministic components. A particle's new position is obtained using a velocity vector calculated based on its previous position and experiences. Numerous studies in the literature have demonstrated that the PSO algorithm is an effective and efficient method, particularly in solving complex and high-dimensional optimization problems [20], [21].

2.3.2. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a specialized architecture developed to overcome the limitations of traditional Recurrent Neural Networks (RNNs) in learning long-term dependencies. LSTM models are widely used in various domains such as speech recognition, language modeling, energy production forecasting, demand analysis, and financial time series modeling.

The LSTM architecture incorporates three fundamental gating mechanisms to regulate information flow and preserve important past information: the forget gate, the input gate, and the output gate. These gates determine how much of the cell state should be retained, how much should be updated, and which information should be passed on as output.

This architecture enables LSTM to more effectively capture long-term dependencies often present in time series data. In applications where high forecasting accuracy is essential, such as in energy production prediction, this capability offers a significant performance advantage, making LSTM a powerful modeling tool in complex temporal analyses [3], [4].

As illustrated in Figure 2.3, the forget gate removes irrelevant past information, while the input gate adds new candidate information created via a tanh activation. The updated cell state is then used to generate the output (h_t) through the output gate. This architecture makes LSTM particularly effective in modeling sequential data, such as in energy production forecasting.

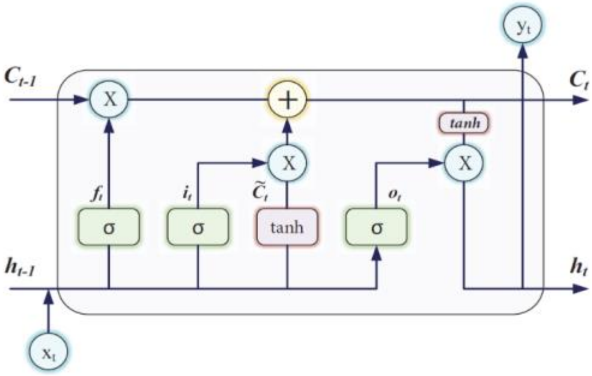

Figure 2.3. LSTM model.

2.4. Error Measures

Error measures represent fundamental statistical tools for evaluating the performance of forecasting models. Commonly used metrics include Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the Coefficient of Determination (R^2). These indicators provide insights into the model's overall predictive performance and goodness of fit, enabling researchers to assess how closely the model's predictions correspond to the actual observed values [3].

2.4.1. Mean Squared Error (MSE)

Mean Squared Error (MSE) is a commonly used metric that quantifies the average squared difference between the predicted values and the actual observations [8], [22], [23].

It is computed by taking the mean of the squared residuals and is expressed by the following formula (Eq. 1):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - x_i^*)^2 \quad (1)$$

- x_i actual value,
- x_i^* predicted value,
- N total number of samples.

MSE squares the errors, it is more sensitive to large errors. Therefore, it should be used with caution in models that are likely to produce large errors [8], [22], [23].

2.4.2. Mean Absolute Error (MAE)

Mean Absolute Error (MAE) represents the average of the absolute differences between the predicted and actual values. It is calculated using the following formula (Eq. 2): [8], [23], [24].

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|x_i - x_i^*| \tag{2}$$

MAE measures the closeness of predicted values to the actual values and does not penalize large errors as heavily as MSE. Therefore, it is often preferred in situations where a more balanced evaluation of prediction errors is desired [8].

2.4.3. Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) is computed by taking the square root of the Mean Squared Error (MSE), and it provides a more accurate reflection of the magnitude of prediction errors (Eq. 3): [24].

$$\sqrt{RMSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - x_i^*)^2} \tag{3}$$

RMSE provides a measure analogous to the standard deviation of prediction errors and is commonly used to assess how closely the predicted values match the actual observations [24].

2.4.4. Coefficient Of Determination (R^2)

The coefficient of determination (R-squared or R^2) indicates the proportion of variance in the dependent variable that is explained by the independent variables. R^2 is used to assess the goodness of fit of a model and ranges between 0 and 1, with values closer to 1 indicating a better explanatory power of the model (Eq. 4): [23].

$$R^2 = \frac{\Sigma(x_i^* - \bar{x}^*)(x_i - \bar{x})}{\Sigma(x_i^* - \bar{x}^*)^2\,\Sigma(x_i - \bar{x})^2} \tag{4}$$

- $\bar{x}$ average of the actual values,
- $\bar{x}^*$ average of the predicted values.

As the R^2 value approaches 1, the model demonstrates a better fit to the data and higher explanatory power. Conversely, an R^2 value close to 0 indicates weak predictive performance.

3. EXPERIMENTAL RESULTS

In In this study, the hyperparameter optimization of the LSTM-based prediction model was carried out using the Particle Swarm Optimization (PSO) algorithm. The model was implemented using the Keras library, with the default tanh activation function applied in the LSTM layers, and a linear activation function used in the output layer, which is suitable for regression problems. The Adam optimization algorithm was employed throughout the training process.

Within this scope, four key hyperparameters of the LSTM model — learning rate, batch size, number of epochs, and dropout rate — were optimized using the PSO algorithm. The search ranges for these hyperparameters were defined based on both similar LSTM-based energy forecasting studies in the literature and the need to balance model accuracy with training efficiency. Accordingly, the following ranges were set: [0.0001–0.1] for the learning rate, [16–128] for the batch size, [5–10] for the number of epochs, and [0.1–0.5] for the dropout rate. These values were directly defined as the lower and upper bounds in the PSO algorithm, allowing it to search for the optimal combination within this space. This configuration aimed to improve the prediction performance of the model while keeping computational cost under control.

The number of LSTM layers was comparatively evaluated using both 2-layer and 3-layer structures. In addition, PSO parameters such as swarm size (20 or 30) and maximum number of iterations (10 or 20) were also varied to analyze their impact on model performance.

All model configurations were evaluated using commonly used error metrics including Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the coefficient of determination (R^2). According to the results, the best performance was achieved with a 3-layer LSTM structure, a swarm size of 20, and 10 iterations, where the R^2 score reached 0.9454. A detailed performance comparison of different configurations is presented in Table 3.1.

Table 3.1. Model performance metrics.

Layer Count	Swarm Size	Max Iter	MAE	MSE	RMSE	R^2
3 Layer	20	10	3.1247	15.9699	3.9962	0.9454
3 Layer	30	20	3.2616	17.5889	4.1939	0.9399
2 Layer	20	10	3.2846	17.1567	4.1421	0.9413
2 Layer	30	20	3.4835	20.1869	4.493	0.931

Table 3.2 summarizes the key hyperparameters used in the PSO and LSTM models and explains how increasing or decreasing these values may affect model performance.

Table 3.2. PSO and LSTM hyperparameter effects.

Category	Parameter	Description	Impact Of Increase	Impact Of Decrease
PSO	SwarmSize	Number of particles simultaneously exploring the solution space	Broader search space, better global optima chances, slower convergence	Faster computation, risk of suboptimal solutions
PSO	MaxIter	Maximum iterations	More optimization attempts, better accuracy	Risk of premature convergence, insufficient optimization
LSTM	Units	Number of neurons in each LSTM layer	Greater predictive power, higher computational cost and risk of overfitting	Lower complexity, reduced accuracy but faster training
LSTM	Dropout Rate	Randomly drops units during training to prevent overfitting	Prevents overfitting; too high may lead to underfitting	Higher risk of overfitting; model may memorize instead of generalizing

Overall, it was concluded that enhancing model performance requires not only architectural depth but also careful tuning of the parameters of the optimization algorithm used. The findings indicate that simpler architectures, when combined with carefully selected PSO parameters, can provide optimal solutions in terms of both accuracy and computational efficiency. In this context, it can be stated that lower SwarmSize and MaxIter values contribute to better generalization performance by preventing overfitting.
In this section of the study, the model configuration that achieved the highest performance was identified as the one with a 3-layer LSTM architecture and PSO parameters set to SwarmSize = 20 and MaxIter = 10. This configuration not only produced low error metric values but also demonstrated strong generalization capability, as evidenced by its high coefficient of determination.
Therefore, in the subsequent stages of the study, detailed analyses were carried out using only this best-performing model configuration. Within the scope of these analyses:

- In section 3.1, the changes in training and validation losses across epochs were examined to evaluate the model's performance during the training process.
- In section 3.2, the predicted values were compared with the actual production data to visually assess the model’s forecasting capability.

These two analytical steps allow for a comprehensive evaluation of the model not only in terms of numerical performance but also in terms of learning behavior and prediction stability.

3.1. Analysis of Training and Validation Losses Across Epochs

Figure 3.1 presents the variation of training loss and validation loss values observed during the training of the deep learning model, based on the number of epochs. The graph aims to illustrate how well the model performs on the training data throughout the learning process, as well as its generalization capability on the validation data. While the training loss represents the error on data directly seen by the model, the validation loss indicates how successfully the model generalizes to unseen data. This figure serves as an important visual tool for assessing the model’s dynamics and identifying potential issues such as overfitting or underfitting [3], [25].
An examination of the training and validation loss curves in the graph reveals that the model's learning process proceeded in a stable manner. The training loss, which was relatively high in the first epoch, decreased rapidly within the following few epochs, indicating that the model was able to learn the fundamental patterns in a short period of time. Around the third epoch, the training and validation loss values converge and continue to follow a similar trend in subsequent epochs. This suggests that the model did not exhibit signs of overfitting, and that the learned patterns were successfully generalized to the validation data.
Moreover, the low degree of fluctuation in the validation loss indicates that the model's performance on the validation set remained stable, and that the training process did not experience high variance learning. The absence of a significant increase in validation loss despite the growing number of epochs also suggests that the learning curve profile is well-suited for the implementation of an early stopping strategy. In conclusion, the graph presented in the figure demonstrates that the model achieved a sufficient level of learning, exhibited strong generalization capability, and with the PSO-optimized hyperparameters contributed positively to this process.

3.2. Comparison of Actual and Predicted Values

Figure 3.2 presents comparative plots of the actual values and the predicted values generated by the developed model, aiming to evaluate its prediction performance. The upper part of the figure illustrates the overall distribution of actual values (blue line) and predicted values (red dashed line) across all test samples. This graph provides a broad perspective for assessing the model’s accuracy and prediction capability. The two separate plots in the lower part of the figure focus on specific sample intervals (approximately 300–700 and 1100–1300), offering a closer examination. These zoomed-in views allow for a more detailed observation of the model’s prediction precision during certain time periods. Such a multi-scale visualization approach is important not only for evaluating the model’s overall performance but also for analyzing its behavior at a local level [8], [25].

Figure 3.1. Model training and validation loss.

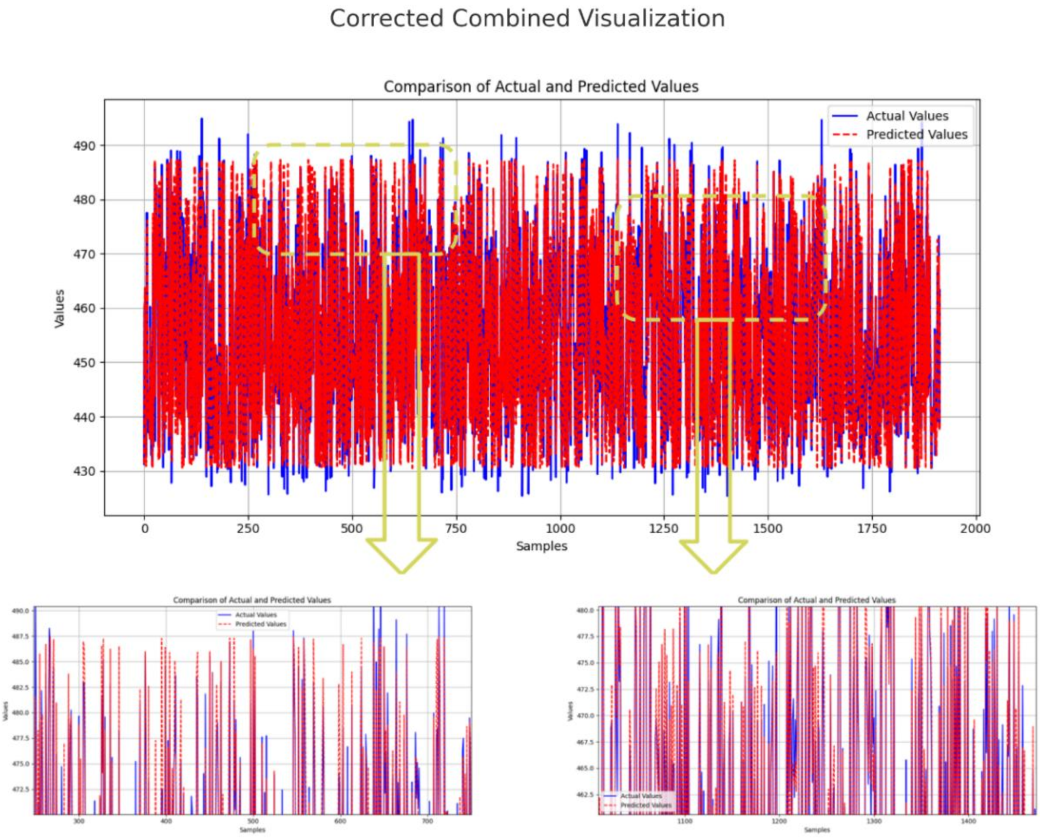

Figure 3.2. Comparison of actual and predicted values.

The graph presented in the figure reveals compelling results in terms of evaluating the model's predictive capability. In the overall view, it can be observed that the predicted values (red dashed line) largely overlap with the actual values (blue line). This indicates that the model was able to successfully apply the patterns learned during training to the test data, demonstrating strong generalization ability.

The zoomed-in plots in the lower section provide a more detailed perspective on the model's performance over specific time intervals. In these sections, the model is shown to respond even to short-term fluctuations and produce values that closely follow the actual data. However, at certain points of sudden change, there are minor deviations between the predictions and the actual values. These discrepancies may be due to abrupt changes in the data or insufficient representation of such patterns in the model inputs.

Nevertheless, the predicted curve effectively follows the overall trend, highlighting the model's reliability in time series data with trend-driven structures such as electricity generation. The visual success observed in the graph is also supported by numerical metrics (MAE, RMSE, R^2), confirming that the model's predictive performance is satisfactory from a practical standpoint.

3.3. Model Performance Comparison

Figure 3.3 presents a comparative analysis of the prediction performances of different model configurations, created using varying numbers of LSTM layers and PSO parameters (SwarmSize and MaxIter). The comparison is made using four different evaluation metrics: MAE, MSE, RMSE, and R^2. The horizontal axis represents the model configurations, while the vertical axis visualizes the values obtained for each metric [17].

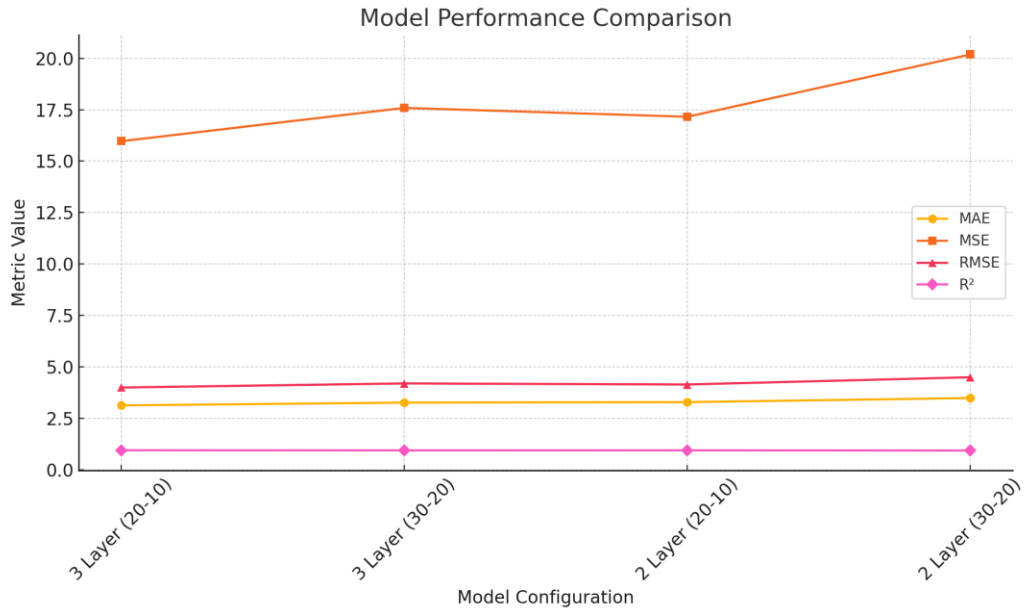

Figure 3.3. Model performance comparison.

An examination of the figure reveals clear differences in the prediction performance of the various model configurations. The configuration with the lowest error values was the 3-layer (SwarmSize=20, MaxIter=10) model. The fact that both MAE and RMSE values are at their minimum in this configuration indicates that the model is capable of producing highly accurate predictions. Additionally, the R^2 score being the highest among all configurations demonstrates that this model explains the variance in the data most effectively.

In contrast, the 2-layer (SwarmSize=30, MaxIter=20) model exhibited significant increases in all error metrics. In particular, the sharp rise in the MSE value suggests an increase in prediction error magnitude and a weakened generalization capability. Furthermore, the drop in R^2 to its lowest level implies that the model struggled to capture the underlying nonlinear relationships in the data.

These results indicate that model performance depends not only on the number of layers, but also directly on the tuning of the hyperparameters of the optimization algorithm. Configurations using fewer iterations and smaller swarm sizes (SwarmSize) tend to reduce the risk of overfitting, resulting in more balanced and generalizable models.

4. CONCLUSIONS

In this study, a forecasting model was developed to improve prediction accuracy in power generation processes by utilizing a Long Short-Term Memory (LSTM) network enhanced with hyperparameter optimization based on Particle Swarm Optimization (PSO). While the core structure of the model is based on the LSTM architecture, key hyperparameters such as the number of layers, units per layer, and learning rate were optimized using the PSO algorithm to enhance overall model performance.

The model was evaluated using hourly operational data obtained from a real-world thermal power plant. Its performance was assessed through commonly used error metrics, including the coefficient of determination (R^2), Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). The results indicated that the PSO-based LSTM model achieved high predictive accuracy, with an R^2 value exceeding 0.94, demonstrating that the predicted net power values were closely aligned with the actual observed data. These findings suggest that PSO-based optimization offers a more reliable and effective alternative compared to conventional forecasting methods.

For future studies, it is recommended to conduct comparative analyses using other metaheuristic algorithms (such as Genetic Algorithm, Artificial Bee Colony, or Simulated Annealing) to further enhance the model's performance. Additionally, testing the model on broader and more diverse datasets would provide insights into its generalizability. The integration of PSO with other machine learning techniques such as Support Vector Machines (SVM), Convolutional Neural Networks (CNN), or Gated Recurrent Units (GRU) may also improve prediction performance and represents a promising direction for further research.

In conclusion, this study demonstrates that hyperparameter optimization using PSO can be successfully applied to LSTM-based forecasting models in energy production processes, offering a robust and practical approach to energy management. Future research may focus on adapting this model to different types of energy sources and operational conditions, thereby contributing to the development of more comprehensive decision support systems in the energy sector.

Author's Contributions

All authors contributed equally.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

REFERENCES

[1] P. Tüfekci, "Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods," International Journal of Electrical Power and Energy Systems, vol. 60, pp. 126–140, 2014.

[2] Y. S. H. Najjar and A. Abu-Shamleh, "Performance evaluation of a large-scale thermal power plant based on the best industrial practices," Sci Rep, vol. 10, no. 1, Dec. 2020.

[3] N. Isık Delibalta, "Zaman Serileri Tahmininde Melez Bir Yaklaşim," 2023.

[4] C. Korkmaz and İ. Kacar, "Zaman Serisinin Kestirimi İçin Uzun-Kısa Süreli Bellek Ağı Yaklaşımı," Çukurova Üniversitesi Mühendislik Fakültesi Dergisi, vol. 39, no. 4, pp. 1053–1066, Dec. 2024.

[5] Y. M. Zhu, L. Pan, J. Shen, and J. Zhang, "Multi-Step Prediction of Short-Term Electrical Load Using The Multi-Network Model Based on PSO-LSTM Algorithm," Proceedings - 2022 4th International Conference on Electrical Engineering and Control Technologies, CEECT 2022, pp. 118–123, 2022.

[6] A. Pranolo et al., "Exploring LSTM-based Attention Mechanisms with PSO and Grid Search under Different Normalization Techniques for Energy demands Time Series Forecasting," Knowledge Engineering and Data Science, vol. 7, no. 1, p. 1, Apr. 2024.

[7] A. Sharma, A. Sharma, J. K. Pandey, and M. Ram, "Swarm intelligence: Foundation, principles, and engineering applications," Swarm Intelligence: Foundation, Principles, and Engineering Applications, pp. 1–140, Feb. 2022.

[8] V. A. G. Raju, J. Nayak, P. B. Dash, and M. Mishra, "Short-term solar irradiance forecasting model based on hyper-parameter tuned LSTM via chaotic particle swarm optimization algorithm," Case Studies in Thermal Engineering, vol. 69, p. 105999, May 2025.

[9] L. P. Dai, "Performance analysis of deep learning-based electric load forecasting model with particle swarm optimization," Heliyon, vol. 10, no. 16, p. e35273, Aug. 2024.

[10] N. S. Santarisi and S. S. F. Doctor, "Predıctıon Of Combıned Cycle Power Plant Electrıcal Output Power Usıng Machıne Learnıng Regressıon Algorıthms," Eastern-European Journal of Enterprise Technologies, vol. 6, no. 8(114), pp. 16–26, 2021.

[11] M. A. Saeed et al., "Electrical power output prediction of combined cycle power plants using a recurrent neural network optimized by waterwheel plant algorithm," Front Energy Res, vol. 11, 2023.

[12] M. Chen, P. Xu, Z. Liu, F. Liu, H. Zhang, and S. Miao, "Air pollution prediction based on optimized deep learning neural networks: PSO-LSTM," Atmos Pollut Res, vol. 16, no. 3, p. 102413, Mar. 2025.

[13] Y. Yao, L. Han, and J. Wang, "LSTM-PSO: Long Short-Term Memory Ship Motion Prediction Based on Particle Swarm Optimization," chine, Aug. 2018, pp. 1–5.

[14] V. A. Truong, N. S. Dinh, T. L. Duong, N. T. Le, C. D. Truong, and L. T. Nguyen, "Hybrid LSTM-PSO optimization techniques for enhancing wind power bidding efficiency in electricity markets," Ain Shams Engineering Journal, vol. 16, no. 2, p. 103285, Feb. 2025.

[15] A. Bazi, F. Basim Ismail, A. Al-Bazi, R. Hikmat Al-Hadeethi, and D. Singh, "Application of Intelligent Computational Techniques in Power Plants: A Review," Advances In Industrial Engineering And Management, vol. 10, no. 1, pp. 10–21, Oct. 2021.

[16] M. B. Ozdemir, T. Menlik, H. İ. Variyenli, and L. Sevin, "Politeknik Dergisi Journal of Polytechnic," pp. 971–978, 2017.

[17] Y. Song, J. Park, M. S. Suh, and C. Kim, "Prediction of Full-Load Electrical Power Output of Combined Cycle Power Plant Using a Super Learner Ensemble," Applied Sciences (Switzerland), vol. 14, no. 24, Dec. 2024.

[18] M. A. Saeed et al., "Electrical power output prediction of combined cycle power plants using a recurrent neural network optimized by waterwheel plant algorithm," Front Energy Res, vol. 11, 2023.

[19] H. Kaya and P. Tufekci, "Local and GlobalLearning Methods for Predicting Power of a Combined Gas & Steam Turbine," Mar. 2012.

[20] R. Eberhart and J. Kennedy, "A New Optimizer Using Particle Swarm Theory," 1995.

[21] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of ICNN'95 - International Conference on Neural Networks, vol. 4, pp. 1942–1948, 1995.

[22] M. Jacob, C. Neves, and D. Vukadinović Greetham, "Short Term Load Forecasting," pp. 15–37, 2020.

[23] G. Memarzadeh and F. Keynia, "Short-term electricity load and price forecasting by a new optimal LSTM-NN based prediction algorithm," Electric Power Systems Research, vol. 192, p. 106995, Mar. 2021.

[24] R. Zhou and X. Zhang, "Short-term power load forecasting based on ARIMA-LSTM," J Phys Conf Ser, vol. 2803, no. 1, 2024.

[25] M. Milanova et al., "Automatic Evaluation of Neural Network Training Results," Computers 2023, Vol. 12, Page 26, vol. 12, no. 2, p. 26, Jan. 2023.