SEVEN
publicações acadêmicas

POWER FLOW PREDICTION USING RECURRENT NEURAL NETWORKS WITH L.S.T.M. AND GENETIC ALGORITHM IN AN IEEE 30 BUS SYSTEM

PREDIÇÃO DE FLUXO DE POTÊNCIA USANDO REDES NEURAIS RECORRENTES COM L.S.T.M. E ALGORITMO GENÉTICO EM UM BARRAMENTO IEEE 30

PREDICCIÓN DEL FLUJO DE POTENCIA MEDIANTE REDES NEURONALES RECURRENTES CON LSTM Y ALGORITMO GENÉTICO EN UNA RED IEEE DE 30 BARRAS

https://doi.org/10.56238/sevened2025.036-051

João Victor Cesario Fernandes[1], Madeleine Rocio Medrano Castillo Albertini[2], Jaqueline Oliveira Rezende[3], Marcus Vinicius Borges Mendonça[4], Paulo Henrique Oliveira Rezende[5], Fabricio Augusto Matheus Moura[6], Arnaldo Jose Pereira Rosentino Junior[7], Antonio Cesar Costa Ferreira Rosa[8]

ABSTRACT
Accurate forecasting of active (P) and reactive (Q) power in electrical networks is fundamental to improving operational reliability and planning in modern power systems. This article proposes a forecasting model based on Long Short-Term Memory (LSTM) neural networks to estimate load demand in an IEEE 30-bus system. The model considers the active and reactive power supplied by the generators as input variables, while the power components of the loads are used as target variables for prediction. To improve performance, a Genetic Algorithm (GA) was used to optimize hyperparameters, which reduced the Mean Absolute Error (MAE) and increased the accuracy of the predictions. The results demonstrate that the proposed approach provides stable predictions of load behavior, highlighting its potential for application in smart grids and microgrid management systems.

Keywords: Power Flow Forecasting. LSTM. Genetic Algorithm. Active Power. Reactive Power. Smart Grids.

[1] Undergraduate student in Electrical Engineering. Instituto Federal do Triângulo Mineiro (IFTM).
E-mail: d202210285@uftm.edu.br Orcid: https://orcid.org/0009-0003-4515-4433
Lattes: http://lattes.cnpq.br/3845027875860177
[2] Post-doctorate researcher in Electrical Engineering. Universidade Federal do Triangulo Mineiro (UFTM).
E-mail: madeleine.albertini@uftm.edu.br Orcid: https://orcid.org/0000-0003-4692-9205
Lattes: http://lattes.cnpq.br/0849523409183491
[3] Dr. in Electrical Engineering. Instituto Federal de Goiás (IFG). E-mail: jaqueline.rezende@ifg.edu.br
Orcid: https://orcid.org/0000-0002-6697-1825 Lattes: http://lattes.cnpq.br/5450777657090154
[4] Dr. in Electrical Engineering. Instituto Federal de Goiás (IFG). E-mail: jaqueline.rezende@ifg.edu.br
Orcid: https://orcid.org/0000-0002-6697-1825 Lattes: http://lattes.cnpq.br/5450777657090154
[5] Dr. in Electrical Engineering. Universidade de Uberlandia (UFU). E-mail: paulohenrique16@gmail.com
Orcid: https://orcid.org/0000-0001-6156-7324 Lattes: http://lattes.cnpq.br/2338311941319569
[6] Dr. in Electrical Engineering. Universidade de Uberlandia (UFU). E-mail: fabricio.moura@ufu.br
Orcid: https://orcid.org/0000-0002-7425-8018 Lattes: http://lattes.cnpq.br/8081704801120219
[7] Dr. in Electrical Engineering. Universidade Federal do Triangulo Mineiro (UFTM).
E-mail: arnaldo.rosentino@uftm.edu.br Orcid: https://orcid.org/0000-0001-6713-8395
Lattes: http://lattes.cnpq.br/5190070034823798
[8] Master's degree. Instituto Federal do Triângulo Mineiro (IFTM). E-mail: antoniorosa@iftm.edu.br
Orcid: https://orcid.org/0009-0008-5464-8984 Lattes: http://lattes.cnpq.br/3845027875860177

RESUMO
A previsão precisa das potências ativas (P) e reativas (Q) em redes elétricas é fundamental para aprimorar a confiabilidade operacional e o planejamento nos sistemas de potência modernos. Este artigo propõe um modelo de previsão baseado em redes neurais do tipo Long Short-Term Memory (LSTM) para estimar a demanda de carga em um sistema IEEE de 30 barras. O modelo considera como variáveis de entrada as potências ativas e reativas fornecidas pelos geradores, enquanto os componentes de potência das cargas são utilizados como variáveis-alvo de previsão. Para aprimorar o desempenho, foi empregado um Algoritmo Genético (GA) na otimização de hiperparâmetros, o que reduziu o Erro Absoluto Médio (MAE) e aumentou a precisão das previsões. Os resultados demonstram que a abordagem proposta fornece previsões estáveis do comportamento das cargas, evidenciando seu potencial de aplicação em smart grids e em sistemas de gerenciamento de microrredes.

Palavras-chave: Previsão de Fluxo de Potência. LSTM. Algoritmo Genético. Potência Ativa. Potência Reativa. Smart Grids.

RESUMEN
La predicción precisa de la potencia activa (P) y reactiva (Q) en las redes eléctricas es fundamental para mejorar la confiabilidad operativa y la planificación en los sistemas de potencia modernos. Este artículo propone un modelo de predicción basado en redes neuronales LSTM (Long Short-Term Memory) para estimar la demanda de carga en un sistema IEEE de 30 barras. El modelo considera la potencia activa y reactiva suministrada por los generadores como variables de entrada, mientras que los componentes de potencia de las cargas se utilizan como variables objetivo para la predicción. Para mejorar el rendimiento, se empleó un algoritmo genético (AG) en la optimización de hiperparámetros, lo que redujo el error absoluto medio (MAE) y aumentó la precisión de las predicciones. Los resultados demuestran que el enfoque propuesto proporciona predicciones estables del comportamiento de la carga, lo que destaca su potencial aplicación en redes inteligentes y sistemas de gestión de microrredes.

Palabras clave: Predicción de Flujo de Potencia. LSTM. Algoritmo Genético. Potencia Activa. Potencia Reactiva. Redes Inteligentes.

1 INTRODUCTION

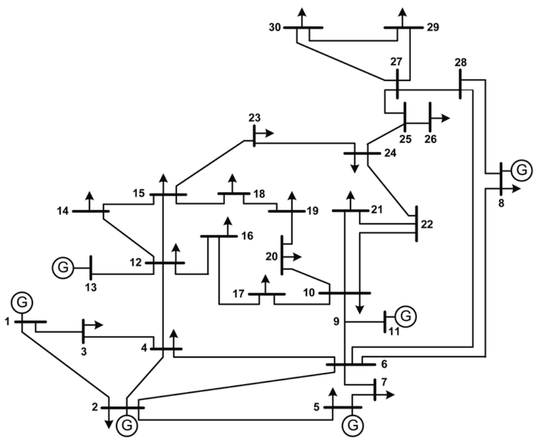
The increasing complexity of modern electrical systems, driven by increased demand and the integration of renewable energy sources, has intensified the need for the use of advanced computational tools for load forecasting and system analysis. Accurate estimation of active (P) and reactive (Q) power is essential to ensure reliable operation, system stability, and efficiency in energy planning.

Traditional forecasting approaches, based on statistical or linear methods, have limitations in dealing with the highly nonlinear and dynamic nature of power systems. In this context, Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) architectures, have stood out as effective alternatives due to their ability to capture long-term temporal dependencies in historical data.

In this work, it is proposed the use of an LSTM network to predict the active and reactive components of a load selected in the IEEE 30-bar test system, considering as input variables the active and reactive powers provided by the generators. To further improve the performance of the model, a Genetic Algorithm (GA) was used to optimize the hyperparameters of the LSTM, enabling more consistent results with lower values of Mean Absolute Error (MAE). Figure 1 presents the schematic representation of the IEEE 30-member system used as an experimental reference. This test system is widely adopted in the literature as a reference for the evaluation of forecasting and optimization methodologies in electric power grids.

Figure 1

Schematic representation of the IEEE system of 30

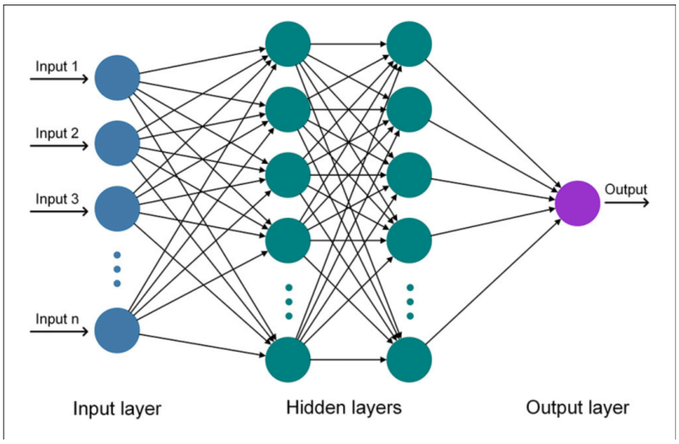


Source: Researchgate.

Figure 2 illustrates the general architecture of the proposed LSTM network, highlighting the temporal sequence of the inputs (active and reactive powers of the generators) and the outputs corresponding to the predictions of the active (P) and reactive (Q) powers of the selected load.

Figure 2

Diagram of the LSTM architecture

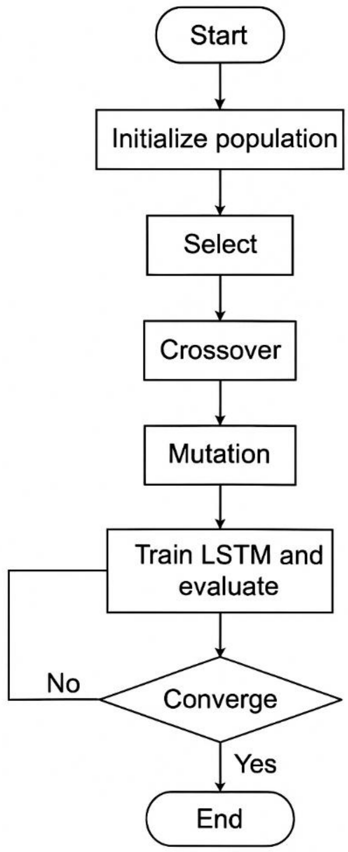


Source: Researchgate.

Finally, Figure 3 presents the flow of the optimization process conducted by the Genetic Algorithm. The GA acts on the hyperparameters of the LSTM (number of neurons, learning rate, number of layers, among others), evaluating performance based on the lowest MAE obtained during validation.

Figure 3

Optimization process flow conducted by the Genetic Algorithm



Source: The authors.

In this scenario, the present article proposes the development of an algorithm based on recurrent neural networks (RNN) with LSTM units for the prediction of the behavior of loads in the IEEE 30-bar system, using historical data of power injections in the busbars. LSTM-based RNNs stand out for their ability to capture complex temporal patterns in sequential data, increasing the accuracy of predictions even under highly dynamic operating conditions. Previous work has shown that accurate load forecasts directly impact the efficiency of the operation of electrical systems, especially when integrated into distributed generation structures, allowing the optimization of energy costs and the increase of the penetration of renewable sources in the grid [3].

To understand the state of the art and contextualize the technical challenges involved, a review of previous studies on time series forecasting in electrical systems and their impacts on operational planning was carried out. In our analysis, different prediction scenarios were tested using real power flow data combined with machine learning approaches. The learning capacity of LSTM proved to be highly effective in capturing temporal dependencies, showing promising results even when using a reduced set of input variables, such as the active and reactive powers of the generators, to predict the active and reactive demand of individual loads. The forecasts were updated every 10 moments of time, simulating a realistic operational horizon.

In the context of industrial and commercial applications, accurate load forecasting not only optimizes the operation of distributed resources, but also contributes to the stability of the power grid in regions with high penetration of renewable sources. Studies such as [4] highlight the potential of Artificial Neural Networks in predicting losses in distribution networks. Similarly, load forecasting with LSTM can act as a central element in reducing technical losses in distribution systems when integrated with advanced load management and energy storage strategies. This work advances over existing models by introducing innovative approaches to deal with the variability and complexity of load data. For this, a robust data set was built, contemplating various operating conditions of the IEEE 30-bar system. The data set used in this work was composed of the operating conditions of its generators and the 21 connected loads. Specifically, the input variables of the model corresponded to the active and reactive powers provided by the generators, while the output variables could, in principle, be the active and reactive powers of each of the 21 loads. However, for the experiments conducted in this study, only load 2 was selected as the prediction target. This choice aimed to reduce the initial complexity of the problem, allowing the methodology to be validated in a controlled scenario. Thus, even though the complete database includes all system loads, the forecast model was trained and evaluated considering exclusively the active and reactive demand of load 2. The data were pre-processed and normalized for training and validation of the prediction model.

Another novelty presented in this work was the use of a Genetic Algorithm (GA) to optimize the hyperparameters of the LSTM network, increasing the convergence speed and reducing the mean prediction error.

In conclusion, this study represents a significant contribution to the development of predictive techniques applied to the operation of electrical systems, with direct implications for the electricity sector and for sustainable energy planning. Future research may investigate the inclusion of additional operational variables, as well as the impact of extreme events and contingencies, with the aim of further improving the proposed predictive models.

2 MATERIAL AND METHOD

2.1 THEORETICAL FOUNDATION

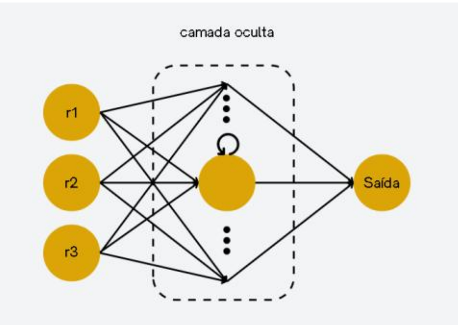
2.1.1 Rnn

Recurrent Neural Networks (RNNs) are a class of artificial neural networks specifically designed to handle sequential data. Unlike traditional *feedforward* networks , RNNs incorporate feedback loops that allow information from previous instants to influence the current output. This characteristic makes them particularly suitable for forecasting tasks, in which temporal dependencies and correlations play a key role [5].

However, traditional RNNs often face limitations, such as *vanishing* and *exploding gradients* when trained on long sequences. These problems reduce the grid's ability to capture long-term dependencies, which are essential for accurate forecasting in electrical systems.

Figure 4

Recurrent Neural Networks (RNN)



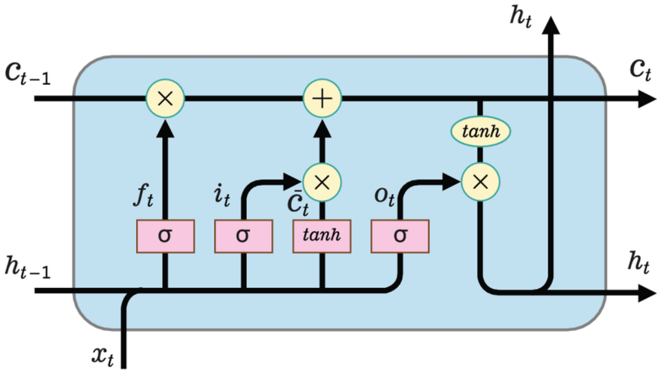
Source: The author.

2.1.1.1 LSTM

Each cell in the hidden layer corresponds to a Long Short-Term Memory (LSTM) unit, illustrated in Figure 4. This is a variation of recurrent neural networks designed to handle long-term dependencies on data sequences. Its main objective is to avoid problems such as the disappearance or explosion of gradients, common in traditional RNNs. To do this, LSTM uses a structured mechanism of gates that control the flow of information over time, allowing the network to decide what information to keep, discard, or update.

Figure 5

Long Short Term Memory (LSTM)



Source: Researchgate

In general, the LSTM cell maintains two main states: the cell state (Ct) and the hidden state (ht). The state of the cell is responsible for storing information over time, while the hidden state serves as the output at each instant. These states are dynamically updated through three main ports: oblivion port, inlet port, and outbound port.

2.1.1.1.1 Door of Oblivion (f):

Determines which previous state information (Ct-1) should be discarded. To do this, a sigmoid function is applied, which generates values between 0 (discard) and 1 (keep). These values are calculated from the combination of the previous hidden state (ht-1) and the current input (xt).

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

2.1.1.1.2 Gateway (i) and Candidate Information (C̃):

Defines what new information will be added to the cell state. First, a sigmoid function calculates the update coefficients. Next, candidate information (C̃t) is generated through the hyperbolic tangent function (tanh), which produces values between -1 and 1. These two steps ensure that only relevant information is incorporated.

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

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

2.1.1.1.3 Cell State Update (Ct):

The current state is updated by combining the previous state with the new candidate information. The oblivion port regulates how much of the previous state will be retained, while the gateway defines the proportion of candidate information that will be incorporated.

$$C_t = f_t \odot C_{t-1} + i_t \odot \overline{C}_t) \quad (4)$$

Here, $\odot$ represents element-by-element multiplication, ensuring a dynamic adjustment of the flow of information.

2.1.1.1.4 Exit Port (o):

Defines which current state information (Ct) will be used to generate the hidden state (ht). The combination of the previous hidden state and the current input goes through a sigmoid function, which controls the fraction released. Then, the cell state is activated with the tanh function and multiplied by the output port values:

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

2.1.1.1.5 Calculation of the Hidden State (ht):

Finally, the hidden state is calculated as the element-to-element product between the output of the output port (ot) and the activated version of the cell state (tanh(Ct)).

$$h_t = o_t \odot tanh(C_t) \quad (6)$$

2.1.1.2 HeatMap

HeatMap is a widely used data visualization tool to represent the values of a matrix or table through color variations. This technique makes it easier to identify patterns, trends, or correlations between variables in complex data sets. In the context of time series and machine learning, HeatMaps are often employed to explore relationships between different variables or to analyze the behavior of data over time.

The operation of a HeatMap is based on assigning colors to numerical values, usually following a continuous scale, such as gradients ranging from blue (for low values) to red (for high values). Each cell of the map represents the intersection between two variables, allowing for an intuitive visualization of their interactions. For example, in a correlation matrix, the HeatMap is used to display the strength and direction of linear relationships between pairs of variables.

Additionally, HeatMaps are useful for identifying anomalies or seasonal patterns in the data. In practical applications, such as time series forecasting, they can be used to analyze the relevance of input variables, identify periods of greater concentration of extreme values, or understand the influence of external factors on the analyzed data. HeatMaps can be implemented in tools such as *Seaborn* or *Matplotlib*, which allow the customization of color scales, labels and annotations, expanding the ability to interpret the results.

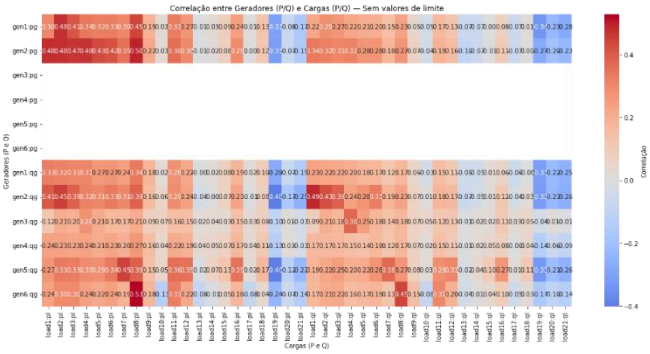
In this way, the use of HeatMaps provides a powerful visual approach to understand and communicate complex information, being an essential tool in studies that require the analysis of large volumes of data and their interrelationships.

The choice of input data in the RNN architecture, illustrated in Figure 4, was based on the analysis of the correlations between the system variables. The HeatMap, presented in Figure 6, shows the correlation relationships between different variables, making it possible to identify those with the greatest impact on the behavior of the system and, therefore, more relevant to the model.

Thus, the power sources were chosen as the input variables for the training, and one of the loads (in this case, Load 2) was selected as the grid output variable.

Figure 6

HeatMap



Source: The author

2.1.1.3 TensorFlow and Keras

TensorFlow and Keras are essential tools for developing *deep learning* models, especially for time-series tasks such as predicting power flow variables in power grids.

TensorFlow, powered by Google, is a highly scalable and efficient open-source library designed to perform complex mathematical operations and support model training on high-performance hardware such as GPUs and TPUs. Keras, integrated as TensorFlow's high-level API, provides a modular, easy-to-use interface for building neural networks, allowing researchers to focus on the conceptual and experimental aspects of modeling.

In the model developed in this work, several features of TensorFlow and Keras were explored for their simplicity and efficiency in implementation. The architecture was built using the Sequential class, which organizes the layers in a linear fashion and is suitable for neural networks that process sequential data, such as the LSTMs employed in this study. The LSTM layers formed the backbone of the model, designed to capture temporal patterns and long-term dependencies on the input data. This capability is critical in time series analysis in power systems, as current load values are strongly influenced by past variations of active and reactive powers.

To enhance the learning process, a Bidirectional layer was added, which allows LSTM to analyze the data both in the direct and inverse direction of time. This approach is particularly advantageous in power flow forecasting, where future operating conditions can also be influenced by past interactions between generation and demand.

In addition to the LSTM layers, Dense layers were included, responsible for performing linear and nonlinear transformations in the internal representations, connecting the LSTM outputs to the final predictions. The output layer was configured with two neurons, corresponding to the active (P) and reactive (Q) powers of the selected load, ensuring that the model produced predictions aligned with the physical quantities of interest.

For the training, the model was compiled using the Adam optimizer, which is known for its efficiency in *deep learning tasks* due to adaptive adjustment of the learning rate. The loss function chosen was the Mean Absolute Error (MAE), which is suitable for regression problems and provides a direct interpretation of the mean of the differences between predicted and actual power values.

After training, the predictions were generated with the Keras predict function, integrated into the processing flow. Overall, TensorFlow and Keras stood out not only for their ease of use, but also for offering advanced solutions to complex problems, such as capturing temporal dependencies in the operation of electrical systems. These frameworks were instrumental in ensuring the efficiency, flexibility, and scientific rigor of the proposed forecasting model.

2.1.1.4 Genetic Algorithm for Hyperparameter Optimization

Genetic Algorithms (GAs) are a class of stochastic optimization techniques inspired by the principles of natural selection and genetics. Initially proposed by John Holland in the 1970s, AGs are widely used to solve optimization problems where the search space is complex, nonlinear, or difficult to approach by traditional gradient-based methods. Their robustness and adaptability make them suitable for a wide variety of engineering applications, including machine learning and power system analysis.

At their core, AGs operate on a population of candidate solutions, called individuals, that are represented by chromosomes (often encoded as binary chains, real value vectors, or more complex structures, depending on the problem). Each chromosome corresponds to a potential solution, the quality of which is evaluated through a fitness function that measures how well it meets the optimization objective.

The process of a GA follows the following key steps:

- Initialization: A population of candidate solutions is randomly generated within the defined search space. In the context of hyperparameter optimization in neural networks, these candidates can represent configurations such as number of LSTM cells, *dropout* rates, learning rate, and number of training epochs.
- Selection: Individuals are selected based on their fitness scores, favoring those who perform best. Strategies such as roulette, tournament, or ranked selection ensure that the strongest individuals are more likely to pass on their traits to the next generation.
- Crossover: Pairs of selected individuals exchange parts of their chromosomal representations to generate offspring. This process mimics biological reproduction, promoting the combination of advantageous characteristics of different solutions.
- Mutation: Random changes are introduced into the chromosomes of offspring with a small probability. The mutation ensures diversity in the population, avoiding premature convergence to suboptimal solutions.
- Replacement: A new population is formed, usually combining the best individuals from the previous generation with the newly generated offspring. The cycle of evaluation, selection, crossover, and mutation is repeated for a set number of generations or until the convergence criteria are met.
-

2.1.1.4.1 Application in this work

In the present study, the Genetic Algorithm was employed to optimize the hyperparameters of the LSTM-based prediction model developed for the IEEE 30-bar system. Instead of manually selecting parameters through trial and error, AG performed an automatic search for the best combination of hyperparameters that would minimize prediction error.

The following hyperparameters were encoded in the AG chromosomes:

- Number of LSTM units per layer: AG explored different configurations of hidden neurons, balancing the complexity of the model and its ability to generalize.
- Number of training seasons: By optimizing this parameter, AG ensured efficient convergence without causing *overfitting* or *underfitting*.
- Dropout rate: AG adjusted the level of regularization to avoid *overfitting*, which is particularly important in sequential models with high representation capacity.

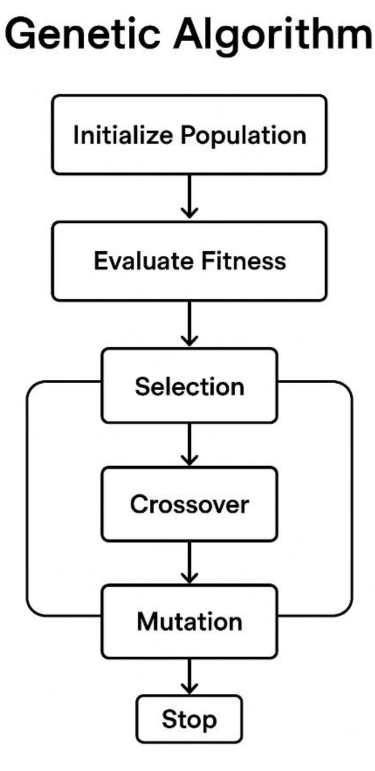
- Learning rate: although it has been fixed in part of the experiments, this parameter can also be optimized by the AG to refine the speed of convergence and the stability of the training.

The fitness function was defined as the Mean Absolute Error (MAE) between the predicted and actual load values (active and reactive powers), after training the model with a certain set of hyperparameters. Lower MAE values indicated better performance, guiding the evolutionary search towards more suitable configurations.

With the integration of the AG into the modeling process, the model achieved a final MAE of 0.0507, a result higher than that of reference models with manually adjusted hyperparameters. This performance highlights the efficiency of AGs in exploring large search spaces and their ability to adaptively identify optimal solutions to complex problems, such as time series forecasting in electrical systems.

Figure 7

Genetic Algorithm Schematic



Source: the authors

3 CASE STUDY AND RESULTS OBTAINED

Once the methodology for the implementation of power flow forecasting was established, its validation necessarily depended on the comparison of the model's performance in relation to a reference case derived from a real distribution feeder.

The code was implemented in Python to predict the active (P) and reactive (Q) powers of a selected load, using as inputs the generation data and auxiliary variables of the system. This approach allows the model to capture temporal and operational patterns inherent in the dynamics of power systems, ensuring accurate short-term forecasts. Such forecasts are crucial for the planning and operation of modern electrical systems, especially in scenarios with distributed generation and fluctuating demand. The procedure adopted and the considerations on the results obtained are presented below.

3.1 SPLITTING THE DATASET

Splitting the dataset is a critical step in ensuring that the model learns reliable patterns and is able to generalize to unseen data. In this study, a temporal division was adopted, allocating 80% of the samples for training and 20% for testing. This division preserves the chronological order of events, avoiding data leakage, a fundamental aspect in time series forecasting. Unlike random splits, which are often used in other machine learning tasks, time sequence maintenance better reflects the model's real-world applications in operational scenarios.

3.2 CONSTRUCTION OF THE SEQUENCES

Since LSTM networks are designed to process sequential data, the code employed a sliding window strategy to construct the temporal sequences. Specifically, each input sequence was composed of 20 consecutive instants of system operation, while the output corresponded to the active and reactive power values of the target load at the subsequent instant. This configuration provides the model with enough historical context to learn the temporal dependencies on the payload's behavior.

3.3 NEURAL NETWORK ARCHITECTURE

The neural network architecture is designed to capture both the temporal and nonlinear dependencies present in the power flow data. It is composed of stacked layers of Bidirectional LSTM, fully connected dense layers, and a linear output layer. Unlike fixed architectures, the configuration of this model — such as the number of LSTM units per layer and the number of training epochs — was not chosen arbitrarily. Instead, a Genetic Algorithm (GA

Bidirectional LSTM layers: two bidirectional layers with different unit numbers (e.g., 64, 128) were tested, allowing the network to learn temporal dependencies in both forward and reverse. This is especially relevant in electrical systems, where current conditions may depend on both past and evolving states.

Dropout regularization: a *dropout rate* of between 1% and 5% was applied in order to mitigate *overfitting* and ensure that the model generalized well to unseen operating conditions.

Dense Layer: after LSTM, a dense layer of variable size (optimized by the AG) was included to refine the representations learned before the final output.

Output Layer: the final layer contained two neurons, corresponding to the active (P) and reactive (Q) power predictions of the selected load.

3.4 OPTIMIZATION OF HYPERPARAMETERS WITH GENETIC ALGORITHM

The Genetic Algorithm played a central role in choosing the optimal configuration of the model. The candidate solutions (chromosomes) encoded hyperparameters such as the number of LSTM cells per layer and the number of training times. AG evolved these solutions over the generations, through the processes of selection, *crossover* , and mutation, converging on architectures that minimized validation error. This evolutionary approach proved to be efficient in balancing the complexity of the model and its predictive accuracy.

It is important to highlight that the hyperparameters obtained represent the best configuration for the dataset and operational conditions of this study. However, different datasets or electrical system scenarios may require alternative configurations, reinforcing the flexibility of combining LSTM networks and evolutionary optimization techniques such as Genetic Algorithms.

3.5 MODEL COMPILATION AND CONFIGURATION

After defining the architecture, the model was compiled with the following elements:

- Loss Function: the Mean Absolute Error (MAE) was used. Unlike Mean Square Error (MSE), which penalizes large deviations more severely, MAE provides a more direct interpretation metric in the context of power flow, representing the average magnitude of forecast errors in both active and reactive powers.

- Adam Optimizer: Adam (Adaptive Moment Estimation) was adopted, which combines advantages of RMSProp and Stochastic Gradient (SGD). Adam dynamically adjusts the learning rate of each parameter based on first- and second-order estimates (mean and variance), accelerating convergence and ensuring stability in the training of complex sequential data.

3.6 MODEL TRAINING

The training was carried out over 15 seasons, with a batch size of 32 samples. During the training, the following strategies were applied:

Adaptive Learning Rate: although the ReduceLROnPlateau *callback* was not explicitly used , the learning rate was adjusted empirically and also by the Genetic Algorithm, ensuring convergence without significant oscillations.

Loss Function Monitoring: training and validation losses were tracked at each time in order to detect signs of *overfitting* or *underfitting*. The evolution of these losses is shown in Figure 7 (Training and Validation Loss Graph), confirming the stable learning process of the model.

3.7 MODEL EVALUATION

After training, the model was evaluated in the test set to measure its generalizability. The evaluation involved the following steps:

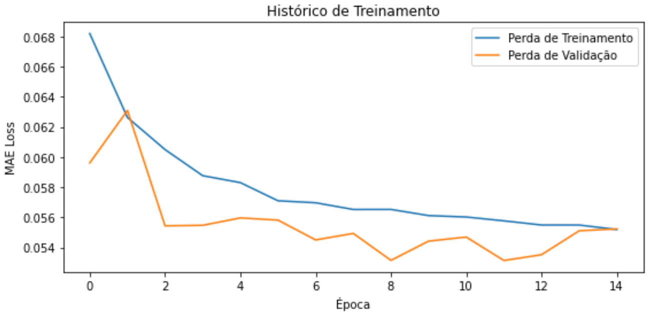
Denormalization: Both the predicted and actual values have been rescaled to their original units using the adjusted *MinMaxScaler* objects . This ensured that the evaluation metrics were interpretable within the physical context of the power flow.

Mean Absolute Error (MAE): the final MAE obtained was 0.0507, reflecting a high degree of accuracy considering the active and reactive power scale in the IEEE 30-bar system. This low error demonstrates the robustness of the model and its ability to capture complex temporal and nonlinear relationships.

Visualization of Forecasts: comparative graphs were generated between actual and predicted values for the active and reactive powers of the target load. These graphs, presented in Figures 8 (Active Power – P2) and 9 (Reactive Power – Q2), highlight the model's ability to follow trends and fluctuations, confirming its effectiveness in short-term prediction of load behavior.

Figure 8

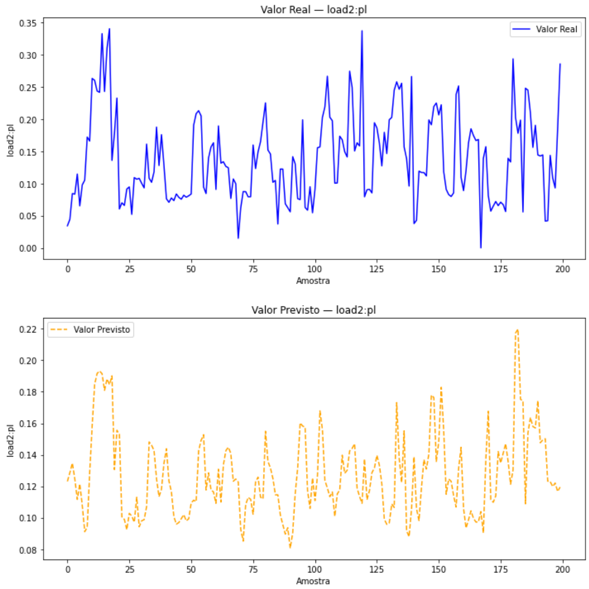
Loss chart



Source: the authors.

Figure 9

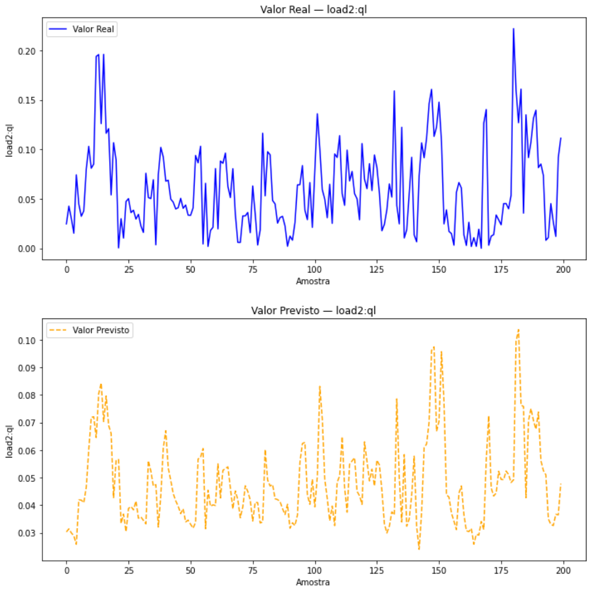
Comparison of actual and forecasted data from P2



Source: the authors.

Figure 10

Comparison of actual and forecasted data from Q2



Source: the authors.

4 STRENGTHS AND LIMITATIONS OF THE STUDY

Most studies that analyze predictions in electrical power systems do not combine recurrent neural network techniques with long-term memory units (LSTM) and Genetic Algorithms (AG) for hyperparameter optimization in load prediction problems. As a strong point of our study, we highlight precisely the use of this integrated approach, which allowed an automatic adaptation of the neural network and reduced subjectivity in the choice of training parameters. In addition, we emphasize that the analysis was performed using data derived from a standard system widely accepted in the literature (IEEE 30 bars), ensuring the reproducibility and comparability of the results. Another positive aspect is that the results achieved a relatively low mean absolute error (MAE), which reinforces the robustness of the proposed model and its practical usefulness for forecasting applications in the electricity sector. Thus, we believe that the results of this study are of great relevance, as they offer initial evidence on the applicability of deep learning techniques combined with evolutionary algorithms in power flow scenarios.

However, we recognize that our study also has limitations. First, it was conducted considering only the prediction of the active and reactive power of a specific load, which restricts the generalization of the results to the entire system. In addition, although the results were satisfactory, the use of simulated data limits the external validity, since noise, measurement failures or contingencies typical of real systems were not considered. Another limiting point is the computational cost associated with the Genetic Algorithm, which can become high in real-time applications or in larger systems. Other limiting factors include the reduced temporality of the analyzed database and the absence of testing under extreme event scenarios or anomalous operating conditions. Although the model has shown predictive capacity under normal conditions, it has not yet been validated in situations of greater variability or electrical disturbances, which are common in real networks.

Therefore, additional research is suggested, especially those involving larger systems and real operation data, to assess the robustness of the model under more challenging conditions. It is also recommended to investigate other optimization metaheuristics, as well as to expand the scope to multiple simultaneous loads, in order to increase the applicability and generalization of the proposed method.

5 CONCLUSIONS

The results obtained in this study demonstrate the effectiveness of Recurrent Neural Network (RNN) models, with Long Short-Term Memory (LSTM) units, for the prediction of load power (active and reactive components) in the IEEE 30-bar system. The proposed model, optimized by means of a Genetic Algorithm for the selection of hyperparameters, achieved a Mean Absolute Error (MAE) of 0.0507, which indicates a high level of accuracy in capturing temporal patterns from historical power flow data.

While the model is designed to predict the active and reactive power of a single load, the methodology can be easily extended to multiple loads and more complex grid configurations. The use of the active and reactive powers of the generators as input variables proved to be sufficient to model the dynamics of the selected load, reducing computational complexity and, at the same time, maintaining a robust predictive performance.

The integration of the Genetic Algorithm was particularly important, as it enabled an automatic search for optimal hyperparameters — such as the number of LSTM cells per layer and the number of training epochs — resulting in higher convergence speed and lower prediction error compared to manually tuned models.

This work represents a significant contribution to the development of intelligent forecasting techniques applied to power systems, offering a solid foundation to improve operational planning, reliability and efficiency in smart grids. Future research may explore the application of this approach to all loads of the IEEE 30-bar system, the inclusion of stochastic renewable sources, and the use of hybrid optimization techniques, with the aim of further refining the predictive capacity of neural network-based models in applications in the electricity sector.

REFERENCES

Alves, F. A. (n.d.). Uma análise das smart grids aplicadas às microrredes inteligentes. Universidade Federal do Ceará.

Bissiriou, A. O. S. (2023). Contribuições para o gerenciamento de energia de microrredes CA monofásicas usadas em comunidades isoladas. Universidade Federal do Rio Grande do Norte.

Correia, V. T. (2021). Sistema de gerenciamento de energia para microrredes. Universidade Federal do Paraná.

Cunha, M. V. da. (2016). Estratégias de gerenciamento pelo lado da demanda aplicadas aos consumidores de BT considerando a tarifa branca e a geração distribuída. Universidade Federal de Santa Maria.

Dronio. (2019). Solar Energy Dataset. Kaggle. https://www.kaggle.com/datasets/dronio/SolarEnergy

Lara Filho, M. O. de. (2021). Modelo robusto orientado a dados para programação diária de operação de microrredes considerando recursos energéticos distribuídos sob incertezas.

Leal, A. G. (2006). Sistema para determinação de perdas em redes de distribuição de energia elétrica utilizando curvas de demanda típicas de consumidores e redes neurais artificiais [Tese de doutorado, Escola Politécnica da Universidade de São Paulo].

Passini, A. F. C., Niz, I. V. G., Munaretto, L. F., Borba, W. F., & Rodrigues, A. C. (2023). Energia solar no Brasil: Oportunidades e desafios. Instituto Brasileiro de Estudos Ambientais.

Rauber, T. W. (n.d.). Redes neurais artificiais. Universidade Federal do Espírito Santo.

Saúde, L. M. S. (2018). Análise comparativa entre os métodos autoregressivo, integrado de médias móveis e rede neural artificial para previsão de séries temporais.