www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 15(4), Nov 2024, 19-41. ISSN: 2007-1558.
https://doi.org/10.61467/2007.1558.2024.v15i4.493

Optimization of Marginal Price Forecasting in Mexico through applying Machine Learning Models

Marcos Fidel Guzmán Escobar[1], *Alberto Alfonso Aguilar Lasserre*[1], *Marco Julio Del Moral Argumedo*[1], *Nicasio Hernández Flores*[2], *Gustavo Arroyo Figueroa*[2]
[1] National Technological Institute of Mexico / Technological Institute of Orizaba, Orizaba, Veracruz, México.
[2] National Institute of Electricity and Clean Energies (INEEL), Cuernavaca, Morelos, México.
E-mails: fidel.mfge@gmail.com, albertoaal@hotmail.com, marcojulioarg@gmail.com, nicasio.hernandez@ineel.mx, garroyo@ineel.mx

Abstract. The Local Marginal Price (LMP) represents the value of energy at a specific moment and location, and its proper management is crucial for the development of the country's strategic sectors. This study compares the ADR, RPSG, SARIMA, and LSTM-H models for predicting the LMP, achieving an approximate effectiveness of 88%. By implementing it in 28 nodes of the three interconnection systems (SIN, BCA, and BCS) in Mexico, the results of the enhanced LSTM network analysis are presented through sensitivity analysis and an ensemble with Prophet, yielding the following metrics: MAE: 0.0189, MSE: 0.0101, RMSE: 0.1007, and MAPE: 12.18, at node 05PAR-115 in Hidalgo del Parral, Chihuahua. This model can construct tree diagrams (ADR) that identify the critical variables for predicting the LMP of any node, significantly contributing to the accuracy of predictive analysis models.

Keywords: Recurrent Neural Networks, Statistical Analysis, Local Marginal Price Forecast, Python code.

Article Info
Received Jul 12, 2024
Accepted Sep 11, 2024

1 Introduction

The Wholesale Electricity Market (MEM) was created as a stipulation of the Electricity Reform in Mexico (Alonso & Gabriel, 2017). With the MEM, it is ensured to have all the inputs for the optimal operation of the National Electric System (SEN), which is divided into three systems:

- National Interconnected System (SIN).
- Baja California System (BCA).
- Baja California Sur System (BCS).

All this system is under the responsibility of the National Energy Control Center (CENACE), whose mission is to manage energy resources in pursuit of economic growth. The Local Marginal Price (LMP) is the energy price at a specific node in the SEN for a defined period, calculated by CENACE for the SCPM (Short-Term Electricity Market). The LMP is the result of three components:

- **Marginal Congestion Component** represents the marginal congestion cost at each Node.
- **Marginal Energy Component,** which represents the marginal energy cost at the reference node of the corresponding Interconnected System; and,
- **Marginal Losses Component** represents the marginal cost of losses at each Node.

The LMP is calculated by adding three components: LMP = Energy Component + Congestion Component + Loss Component. Each component can assume positive, negative, or even null values.

In developing a machine learning model to forecast the LMP, it is crucial to integrate weather variables into the model to enrich the database. This results in a more comprehensive and robust model that includes wind speed, temperature, precipitation, and solar radiation. These variables in the context of energy demand and generation in Mexico are especially relevant due to the complexity of the characteristics of the country's nodes, which amount to more than 2,300 and are distributed throughout the Mexican Republic; their inclusion provides an additional and critical perspective to improve the accuracy and relevance of the forecast model (Alonso & Gabriel, 2017).

Although models based on recurrent neural networks, such as LSTM (Long Short-Term Memory), can offer accurate predictions in many cases, they also face significant challenges; one of the essential challenges lies in adequately selecting the model architecture and optimizing hyperparameters. Due to these circumstances, a comprehensive determination of the structure of an LSTM network and its hyperparameters is carried out to improve predictions significantly. Additionally, the LSTM network approach has been efficiently integrated with another forecasting algorithm, in this case, Prophet. The hybrid LSTM model has been configured to generate predictions 48 hours into the future using a time-distributed neuron. Furthermore, during the Prophet training, the algorithm was instructed to consider the predictions generated by LSTM as part of the input data, allowing both approaches to merge synergistically. This combined approach improves the accuracy of predictions and provides a 95% confidence interval, providing greater confidence in the forecasted results.

2 Literature Review

2.1 Artificial Neural Networks

Two of the most commonly used topologies, according to differences in the way connections are made (Ponce, 2010), are:

a) **Feed-forward networks.** The flow of information from inputs to outputs is exclusively forward, extending through multiple layers of units, but there is no feedback connection. Fig. 1 shows a feed-forward network with four layers, with "x" inputs and "y" outputs.

b) **Recurrent Neural Networks.** They contain feedback connections, which can result in a process of evolution towards a stable state where there are no changes in the activation state of neurons, allowing the network to process data sequences and capture temporal dependencies in the input data. See Fig 2

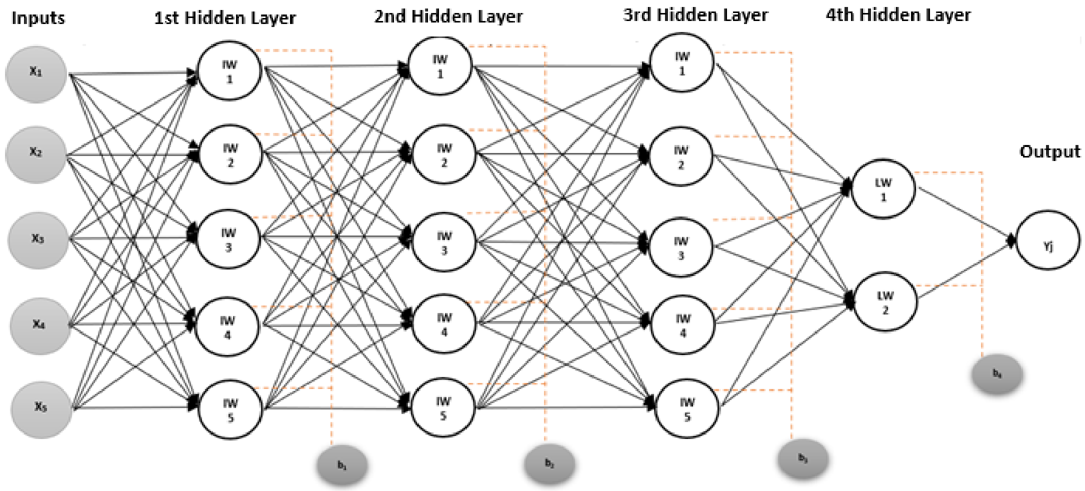


Fig. 1 ANN Feedforward Neural Network.

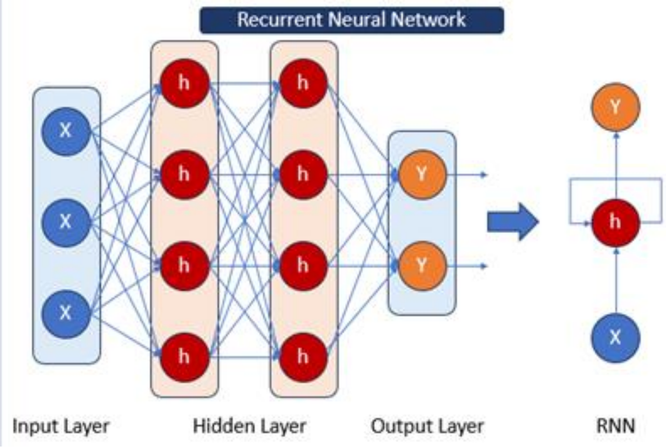


Fig 2. RNN (Recurrent Neural Network), where the propagation forms a cycle.

2.2 Literature Review

The study of using artificial intelligence (AI) techniques has expanded in recent years due to the many satisfactory results achieved with them. Below is a literature review exploring the application of various methods to address forecasting challenges related to the energy sector, starting with those in Artificial Intelligence, such as Machine Learning and Deep Learning, and then complementing them with approaches from the traditional statistical field.

2.2.1 Machine Learning (ML)

Kyung Keun Yun, Sang Won Yoon, and Won (2022) propose an algorithm that utilises two independent input features: internal technical indicators and external market prices. This bilateral forecasting scheme undergoes a two-stage feature selection process, including feature set expansion, a hybrid genetic algorithm, machine learning regressions to identify essential features, and importance score filtering to select the optimal features. Additionally, they employ Savitzky-Golay smoothing to enhance local interpretability through optimal piecewise curve fitting and examine potential clusters of external features.

On the other hand, applying machine learning techniques has tackled uncertainty. For example, the document (Jincheng Zhang, Xiaowei Zhao, Siya Jin, & Greaves, 2022) focuses on developing a new wave prediction method for energy generation but with a probabilistic approach. The Bayesian Machine Learning (BML) approach for predicting resolved-phase waves is employed, which can leverage ML's capability to address nonlinear problems while accounting for different types of uncertainties through the Bayesian framework. Since inferring the actual posterior distribution in the Bayesian Neural Network (BNN) model is generally infeasible due to its extreme computational complexity, various approximation methods have been utilized, such as Monte Carlo dropout (MC) and modern ML hardware development (e.g., GPUs). The proposed method considers both random uncertainty (i.e., uncertainty of resolved-phase wave information) and epistemic uncertainty (uncertainty due to model capacity). According to the authors, this is the first time that real-time nonlinear wave prediction with quantified uncertainty, including epistemic and random uncertainties, has been achieved.

Livas-García, Tzuc, May, Rasikh Tariq, and Torres (2022) analyse the scope of AI in the LMP using a database of 11 independent variables (seasonal, environmental, operational, and economic factors) to forecast the LMP. The study preprocesses the data through cleaning filters, smoothing, and normalisation. Subsequently, they train a model based on Artificial Neural Networks (ANN) and another based on fuzzy logic. Finally, they conduct a Global Sensitivity Analysis (GSA), identifying that the day of the year has the most significant impact on the loss and congestion components, while fuel prices significantly influence the energy component in the Yucatan Peninsula, Mexico.

2.2.2 Deep Learning (DL)

In another study, Tovar Rosas (2020) addresses the challenge of integrating intermittent renewable energy sources (RES) into the electrical grid through the use of battery energy storage systems (BESS) in Baja California. The study proposes precise charging and discharging schedules for two BESS systems powered by photovoltaic solar energy and wind energy in the Baja California Sur (BCS) region of Mexico. These schedules are based on RES and electric demand predictions generated by a hybrid model of convolutional neural network long short-term memory (CNN-LSTM). The results demonstrate that this integration can effectively mitigate peaks in electric demand at the studied location.

2.2.3 Statistical Models (SM)

Santiago López and García (2018) describe and characterise the temporal evolution of the hourly-daily LMP time series resulting from the MECP in Mexico, while simultaneously analysing their volatility over different periods for each energy market (MDA and MTR) and for each interconnected system (SIN, BCA, and BCS). The non-stationary nature of the time series is attributed to patterns associated with weekly frequency, and the analysis focuses on historically analysed data.

Ramos, Cortés, Latorre, and Cerisola (2006) analyse the relationship between marginal electricity market price and demand using mathematically defined clusters. They conclude that electricity price fundamentally depends on demand and available power, which varies over time.

Singhal and Swarup (2011) use historical data and estimated future factors to adjust and extrapolate prices and quantities. They develop a neural network method to forecast market clearing prices (MCP) for daily energy markets. The neural network structure consists of a three-layer backpropagation (BP) network, which incorporates fuel prices and other factors, using a classic backpropagation neural network model.

2.2.4 Conclusion of the State of the Art

The literature related to the electricity price forecasting area and other studies address artificial intelligence (AI) models based on artificial neural networks (ANN), statistical time series, and approaches adopted by the research community to tackle the problem of electricity price prediction. Often, acceptable results have been obtained using statistical models such as ARIMA (Autoregressive Integrated Moving Average). However, ANNs have technical advantages such as managing nonlinearity complexity, searching for behavioral patterns, and learning from the database.

On the other hand, various studies have compared different architectures to determine which offers the best accuracy in modeling and deep learning techniques. However, it is essential to note that variables commonly used in training these models often focus on historical electricity and oil prices or electric demand and energy supply (other variables besides those reported in the literature are limited or reserved).

To date, most studies have focused on predicting the value of the Local Marginal Price (LMP) in real-time, with a notable absence of research that individually analyzes energy components constituting the fundamental elements of energy price. This research gap represents a significant opportunity, especially considering the crucial implications that congestion, energy, and loss components can have in the detailed analysis of LMP. Therefore, the present project aims to improve the understanding and forecast of the Local Marginal Price (LMP) in a replicable manner across nodes throughout the country; to achieve this purpose, a hybrid approach combining machine learning models with statistical analysis will be employed, incorporating key climatological variables into cutting-edge models and providing a more comprehensive and efficient approach to LMP forecasting, capable of predicting both real-time and future values.

3 Methodology

AAlthough recurrent neural networks are robust in capturing temporal patterns, they can also be susceptible to overfitting if not adequately tuned; choosing the right amount of layers, units, and activation functions and configuring other parameters can be an iterative process that requires experience and specialized knowledge. Through a sensitivity analysis, the appropriate LSTM network architecture for forecasting more than 24 hours into the future is deciphered, and successful validation is implemented with walk forward validation.

For the construction of the models, where the objective is to forecast the Local Marginal Price of the nodes in the Mexican Republic, mainly, a node was randomly chosen from among the more than 2,300 interconnected nodes (CENANCE, 2022). To demonstrate the results, a case study is implemented at node 05PAR-115 in Hidalgo del Parral, Chihuahua. The methodology employed is as follows (see Figure 3); starting with the goal of forecasting the local marginal price, the first step is the collection and preprocessing of the databases. Next, a forecasting model is constructed, which will then be subjected to a sensitivity analysis of its hyperparameters to determine the appropriate ones for the forecast. The prediction error metrics are compared to measure the model's performance, and once the model's predictive capability is accepted, it is implemented, achieving the forecast results. Each stage of the methodology is then described in detail.

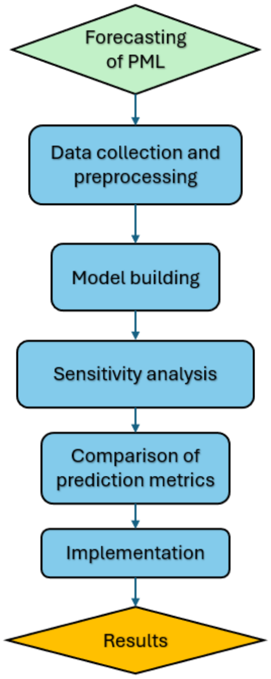
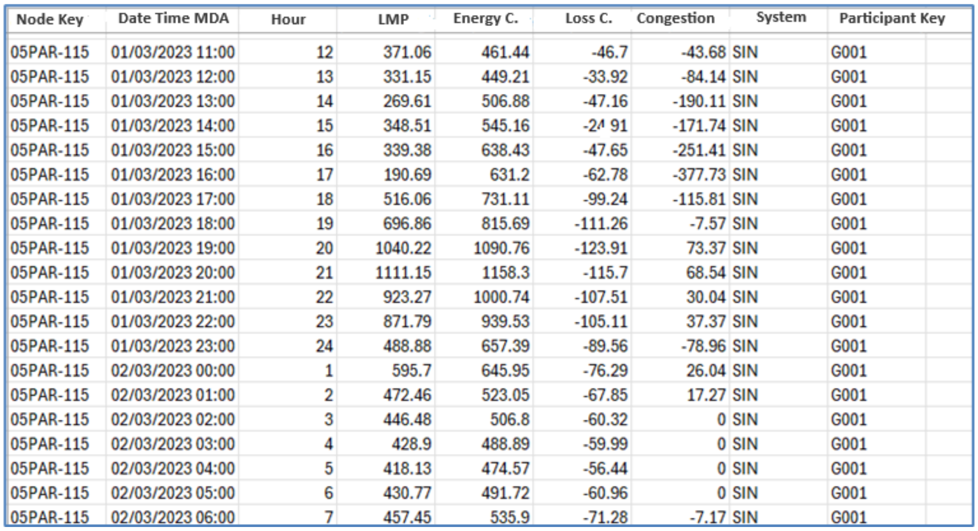


Fig. 3 Methodolgy.

3.1 Data Collection and Preprocessing

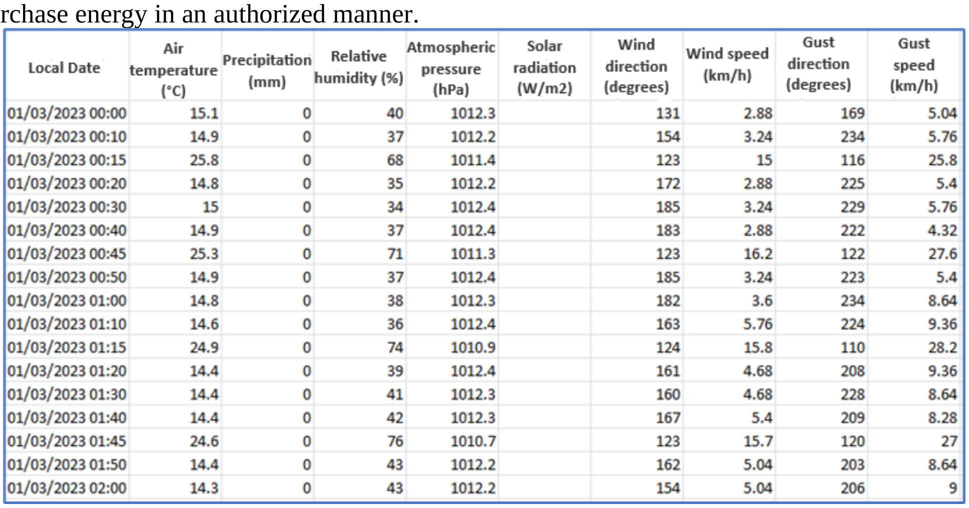
Data is collected from the National Energy Control Center (CENACE) (CENANCE, 2022)(see Figure 4) and the National Water Commission (CONAGUA) (CONAGUA, 2022)(see Figure 5), whose records contain hourly data from the year 2019 to 2023 for CENACE, while for CONAGUA the records are ten minutes, which should be averaged to convert them to hourly data. Additionally, having very old or atypical data is not helpful or recommended, so it is advisable to establish a period of statistical analysis first. Since we aim to predict a forward marginal price in the future, we need to analyze the most recent data within a reasonable period, so data from a month is more than enough for graphs and calculations as we have 720 records (24 records * 30 days), in this case from March 2023. See Figure 4.



Node Key	Date Time MDA	Hour	LMP	Energy C.	Loss C.	Congestion	System	Participant Key
05PAR-115	01/03/2023 11:00	12	371.06	461.44	-46.7	-43.68	SIN	G001
05PAR-115	01/03/2023 12:00	13	331.15	449.21	-33.92	-84.14	SIN	G001
05PAR-115	01/03/2023 13:00	14	269.61	506.88	-47.16	-190.11	SIN	G001
05PAR-115	01/03/2023 14:00	15	348.51	545.16	-24 91	-171.74	SIN	G001
05PAR-115	01/03/2023 15:00	16	339.38	638.43	-47.65	-251.41	SIN	G001
05PAR-115	01/03/2023 16:00	17	190.69	631.2	-62.78	-377.73	SIN	G001
05PAR-115	01/03/2023 17:00	18	516.06	731.11	-99.24	-115.81	SIN	G001
05PAR-115	01/03/2023 18:00	19	696.86	815.69	-111.26	-7.57	SIN	G001
05PAR-115	01/03/2023 19:00	20	1040.22	1090.76	-123.91	73.37	SIN	G001
05PAR-115	01/03/2023 20:00	21	1111.15	1158.3	-115.7	68.54	SIN	G001
05PAR-115	01/03/2023 21:00	22	923.27	1000.74	-107.51	30.04	SIN	G001
05PAR-115	01/03/2023 22:00	23	871.79	939.53	-105.11	37.37	SIN	G001
05PAR-115	01/03/2023 23:00	24	488.88	657.39	-89.56	-78.96	SIN	G001
05PAR-115	02/03/2023 00:00	1	595.7	645.95	-76.29	26.04	SIN	G001
05PAR-115	02/03/2023 01:00	2	472.46	523.05	-67.85	17.27	SIN	G001
05PAR-115	02/03/2023 02:00	3	446.48	506.8	-60.32	0	SIN	G001
05PAR-115	02/03/2023 03:00	4	428.9	488.89	-59.99	0	SIN	G001
05PAR-115	02/03/2023 04:00	5	418.13	474.57	-56.44	0	SIN	G001
05PAR-115	02/03/2023 05:00	6	430.77	491.72	-60.96	0	SIN	G001
05PAR-115	02/03/2023 06:00	7	457.45	535.9	-71.28	-7.17	SIN	G001

Fig. 3 Original Data from CENACE.

From Figure 4 (PML data), the node key is a name assigned to the node and is an alphanumeric key consisting of initials and the voltage the node handles. The Date Time MDA is the registration date in the format "yyyy-mm-dd". The hour corresponds to the time of day when the data was recorded. The "Local Marginal Price" is the target variable we will define as the dependent variable in the statistical analysis and forecast. The variables energy component, loss component, and congestion component are independent in our analysis and forecast; the energy component is the price of generating electricity at that node, the loss component corresponds to the price of energy lost due to distribution and other leaks, and finally the congestion component is the cost or price of electricity that, given the emergency or limited capacity of transmission lines, increases the price due to transmission network congestion at that node. The system variable corresponds to the acronyms of one of the interconnection systems, such as SIN (National Interconnected System), BCA (Baja California Norte system), and BCS (Baja California Sur system). Finally, the Participant Key variable corresponds to the qualified user within the MEM (Wholesale Electricity Market) system who can purchase energy in an authorized manner.



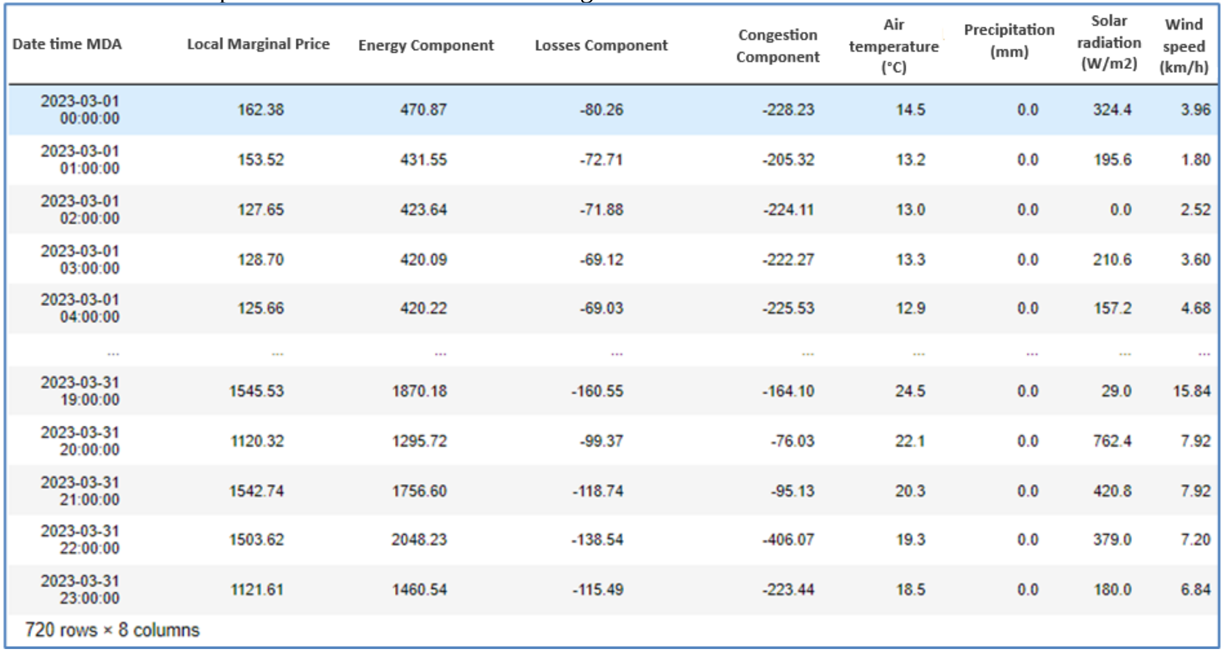
Local Date	Air temperature (°C)	Precipitation (mm)	Relative humidity (%)	Atmospheric pressure (hPa)	Solar radiation (W/m2)	Wind direction (degrees)	Wind speed (km/h)	Gust direction (degrees)	Gust speed (km/h)
01/03/2023 00:00	15.1	0	40	1012.3		131	2.88	169	5.04
01/03/2023 00:10	14.9	0	37	1012.2		154	3.24	234	5.76
01/03/2023 00:15	25.8	0	68	1011.4		123	15	116	25.8
01/03/2023 00:20	14.8	0	35	1012.2		172	2.88	225	5.4
01/03/2023 00:30	15	0	34	1012.4		185	3.24	229	5.76
01/03/2023 00:40	14.9	0	37	1012.4		183	2.88	222	4.32
01/03/2023 00:45	25.3	0	71	1011.3		123	16.2	122	27.6
01/03/2023 00:50	14.9	0	37	1012.4		185	3.24	223	5.4
01/03/2023 01:00	14.8	0	38	1012.3		182	3.6	234	8.64
01/03/2023 01:10	14.6	0	36	1012.4		163	5.76	224	9.36
01/03/2023 01:15	24.9	0	74	1010.9		124	15.8	110	28.2
01/03/2023 01:20	14.4	0	39	1012.4		161	4.68	208	9.36
01/03/2023 01:30	14.4	0	41	1012.3		160	4.68	228	8.64
01/03/2023 01:40	14.4	0	42	1012.3		167	5.4	209	8.28
01/03/2023 01:45	24.6	0	76	1010.7		123	15.7	120	27
01/03/2023 01:50	14.4	0	43	1012.2		162	5.04	203	8.64
01/03/2023 02:00	14.3	0	43	1012.2		154	5.04	206	9

Fig. 4 Original Data from CONAGUA.

Regarding Figure 5 (weather data), we have the following meteorological variables: Air temperature (°C), Precipitation (mm), Relative humidity (%), Atmospheric pressure (hPa), Solar radiation (W/m^2) Wind direction (degrees), Wind speed (km/h), Gust direction (degrees), Gust speed (km/h). The corresponding task is to process the data for proper management of its information, analyzing its characteristics to apply forecasts with Machine Learning correctly. Below is the variable selection process for this study, justifying the inclusion or exclusion of certain variables based on their relevance to predicting the LPM and their unity with Mexico's energy context. Here is a brief description of how each variable selected for the models could influence the local marginal energy price:

- **Solar radiation (W/m^2):** Mexico has excellent solar potential, and solar radiation directly affects energy production in photovoltaic solar plants. Higher solar radiation would imply higher solar energy production, which could influence energy prices.
- **Air temperature (°C):** Temperature can influence energy demand. For instance, scorching days may increase the usage of air conditioners, leading to a rise in electricity demand. Conversely, on rare cold days, there might be an increase in demand for heating in certain regions. This fluctuation in demand can influence energy prices.
- **Precipitation (mm):** Rainfall can have a dual effect. On one hand, it may decrease the solar radiation reaching solar panels, affecting solar energy production. On the other hand, if hydroelectric production exists in the region, rainfall can influence water storage and flow, affecting energy generation.
- **Wind Speed (km/h):** Including wind speed is essential. Although this region is not predominantly known for large-scale wind generation like other areas of Mexico, wind speed remains a relevant indicator.
- Regarding the LPM data, where we have the components of LPM (energy, congestion, and loss), it's known that these elements together make up the LPM of any node. Here's a description of the reasons:
- **Energy Component:** Represents the cost of generating electricity and is the main component of LPM. In La Paz, this includes local generation and imported energy, with the mix of energy sources used being relevant, such as solar, thermal, and, to a lesser extent, wind.
- **Congestion Component:** This component reflects the costs associated with transmission capacity limitations. In areas like La Paz, which are relatively isolated from the National Interconnected System, congestion can be a significant factor, especially if local generation is insufficient to meet demand and more energy import is required.
- **Loss Component:** This refers to the energy lost during transmission and distribution from the generation point to the node. Since La Paz is a geographically isolated region, losses may be higher than those of nodes more connected to the central system, thus affecting LPM.

Therefore, the other variables such as nodeKey, dateHourMDA, hour, system, participantKey, Wind Direction (degrees), Gust Direction (degrees), Gust Speed (km/h), Relative Humidity (%), Atmospheric Pressure (hpa), and Local Date should be discarded, as their values are known to have little to no relationship with LPM other than serving as hourly references for some. In the case of meteorological data, only the most relevant variables for the technology of generation plants were selected. Now, the database has a dimension of 720 rows by eight columns. This configuration provides ample information to conduct detailed analysis and build accurate predictive models of LPM. See Figure 6.



Date time MDA	Local Marginal Price	Energy Component	Losses Component	Congestion Component	Air temperature (°C)	Precipitation (mm)	Solar radiation (W/m2)	Wind speed (km/h)
2023-03-01 00:00:00	162.38	470.87	-80.26	-228.23	14.5	0.0	324.4	3.96
2023-03-01 01:00:00	153.52	431.55	-72.71	-205.32	13.2	0.0	195.6	1.80
2023-03-01 02:00:00	127.65	423.64	-71.88	-224.11	13.0	0.0	0.0	2.52
2023-03-01 03:00:00	128.70	420.09	-69.12	-222.27	13.3	0.0	210.6	3.60
2023-03-01 04:00:00	125.66	420.22	-69.03	-225.53	12.9	0.0	157.2	4.68
...	...	...	...	...	...	...	...	...
2023-03-31 19:00:00	1545.53	1870.18	-160.55	-164.10	24.5	0.0	29.0	15.84
2023-03-31 20:00:00	1120.32	1295.72	-99.37	-76.03	22.1	0.0	762.4	7.92
2023-03-31 21:00:00	1542.74	1756.60	-118.74	-95.13	20.3	0.0	420.8	7.92
2023-03-31 22:00:00	1503.62	2048.23	-138.54	-406.07	19.3	0.0	379.0	7.20
2023-03-31 23:00:00	1121.61	1460.54	-115.49	-223.44	18.5	0.0	180.0	6.84

720 rows × 8 columns

Fig. 5 Dataset of node 05-PAR-115.

3.2 Model Construction

Several machine learning models have been developed, including ADR, RPSG, SARIMA, and LSTM-H. These models are designed to be adaptable to various data patterns. They are trained using an 80% subset of data and validated with 10%, leaving 10% for testing. Standard forecast error metrics are employed, with walk-forward validation used to evaluate model performance:

Model 1: The Decision Tree Regression model is implemented using Python's Scikit-learn library. The process begins by selecting input and target variables from the dataset. Next, the dataset is split into training, validation, and test sets, with 80%, 10%, and 10% of the data, respectively. Once the dataset is prepared, the model is trained, tuned, and validated. Subsequently, the model is evaluated using the test set to determine its performance on previously unseen data.

Model 2: Second-degree Polynomial Regression is implemented using Python's Scikit-learn library. It extends linear regression to capture nonlinear relationships between variables, especially when data suggests a quadratic trend. Similar to the Decision Tree, variables are selected, and the dataset is split into training, validation, and test sets. However, in this case, feature transformation is performed to fit the polynomial model. Predictions are made for the test set after training the model with the transformed training set.

Model 3: The "Seasonal" component of the SARIMA model allows for considering seasonality in the data, which is helpful for recurring patterns such as those occurring every 24 hours or every week. Proper selection of parameters (p, d, q, P, D, Q, s) is crucial and sometimes requires iterations. It is ideal for data with seasonal patterns and dependencies on past values. The "stats models" library in Python is used to implement it. Auto-ARIMA automates parameter selection, saving time and providing an objective way to choose the best model based on criteria such as AIC. The process involves identifying the best SARIMA model, considering seasonality, differentiation orders, and periodicity. The model coefficients, AIC, and other statistics help evaluate its performance. Finally, the SARIMA model is trained with the training data, predictions are made on the test data, and its performance is evaluated using metrics such as MAPE.

Model 4: When combining LSTM with Prophet, we adopt a sequential approach (seq2seq) to model complete sequences of future points rather than just a single point in time. This process involves several key steps:

1. **Data Preparation in seq2seq Format:** Data is organized into input-output pairs to enable the model to predict future sequences instead of individual values. This data structure is fundamental for the seq2seq approach.
2. **Utilization of an LSTM Layer followed by a Time Distributed Layer:** The LSTM model captures complex temporal dependencies in the data. The Time Distributed layer allows for applying a dense layer to each time point in the output sequence, facilitating the generation of sequential predictions.
3. **Use of Time Series Split for Temporal Data Splitting:** Temporal data splitting with Time Series Split ensures that the model is trained and appropriately validated, avoiding leakage of future information into the past during the training process.

Once the LSTM model has captured the temporal dependencies, Prophet comes into play to model trends and seasonality in the data. Combining predictions from both models provides a more robust and accurate forecast.

3.3 Sensitivity Analysis

Two main architectures are identified for this analysis: "1 + 1" and "2 + 1". The first consists of an LSTM layer with 128 units that returns sequences instead of a single value, followed by a Time Distributed layer to apply the dense layer to each time step. The second architecture is more complex and includes two LSTM layers (128 and 64 units, respectively), Batch Normalization and Dropout layers. It uses the Repeat Vector technique to repeat the output before feeding it to another LSTM layer. A Time-distributed layer is also applied to the production to ensure consistent prediction for each step. Including Batch Normalization and Dropout layers in the "2 + 1" architecture aims to improve the model's robustness by stabilizing training and reducing overfitting. The Repeat Vector technique provides additional contextual information to the model, while the Time Distributed layer ensures coherent sequential output. Sensitivity analysis involves exploring 20 combinations of parameters and conditions, 10 for each of the two architectures. The results are recorded and analyzed to assess the impact of changes on prediction accuracy. Graphs display the actual and forecasted values and MAE and MAPE metrics for each explored configuration.

3.4 Prediction Metrics Comparison

In each of the algorithms, training and cross-validation (walkforward) are crucial, as well as experimentation, to optimize their performance and obtain the metrics for their corresponding comparison. Both temporal validation techniques with time series split and walkforward validation were used for rigorous evaluation. These techniques divided the data into temporally coherent training and test sets, allowing the model's performance to be evaluated under realistic time series conditions. The results of standard metrics (MAE, MSE, RMSE, MAPE) and temporal validation techniques ensured a comprehensive and rigorous assessment of the performance of the developed models, providing a complete understanding of their ability to predict future values of the time series accurately.

3.5 Implementation

The obtained model is replicated at node 05PAR-115 in Hidalgo del Parral, Chihuahua, where its performance is also evaluated at several strategic nodes of the interconnected systems throughout the country. This detailed approach provides confidence in the replicability of the model in a broader deployment in the national electrical system.

3.6 Results

The comparison of 4 machine learning algorithms provides comprehensive results and ensures predictive accuracy by understanding the behavioral patterns of data and replicating them nationwide through Python code. This code handles temporal data exploration, statistical analysis, and accurate forecasting following machine learning guidelines, thus offering state-of-the-art prediction capabilities. The hybrid LSTM model efficiently captures the temporal behavioral component, marking a significant advancement in predictive modeling.

4 Results

4.1 Database Explorer

A data selector and visualization tool are created using the Tkinter library, providing an interactive interface for selecting and visualizing data over specific periods. Users can choose the period they wish to view the LPM data, as shown in Figure 7:

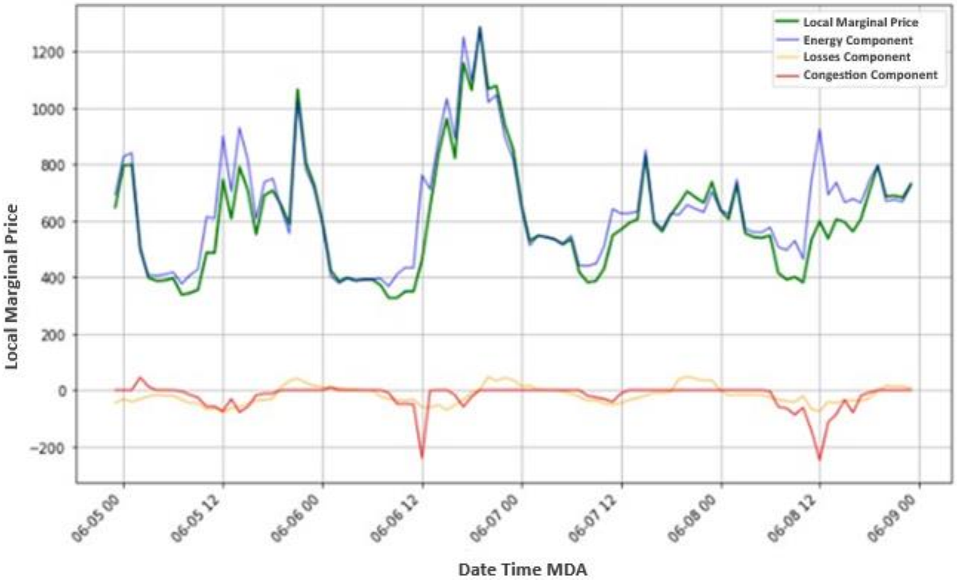


Fig 6. LPM Database Exploration (node 05PAR-115).

Figure 7 allows users to explore the prices of node 05PAR-115 in Hidalgo del Parral, Chihuahua, using the database explorer. The temporal window is adjustable, allowing users to zoom in on the graph at their discretion.

4.2 Data Preprocessing

Immediately after the explorer, the forecasting code is executed, consisting of the four machine learning models that unify the LPM databases with weather variables. Here, the Local marginal price is the target variable, and the three components integrate the LPM in the corresponding node and hour, along with air temperature, precipitation, solar radiation, and wind speed. The model conducts the machine learning statistical analysis to identify their correlation and significance with the LPM.

4.3 Statistical Analysis

The statistical analysis and correlation of the variables are carried out through a Python code block, which prepares and visualizes an analysis of the numeric columns of the 'PML_recent' data frame after removing specific columns irrelevant to the study. This analysis will provide information about the data's central tendency, dispersion, and correlation, allowing for a deeper understanding of the distribution and behavior of the data and the relationships between the variables. Beginning with a boxplot of the data at this particular node in Figure 8:

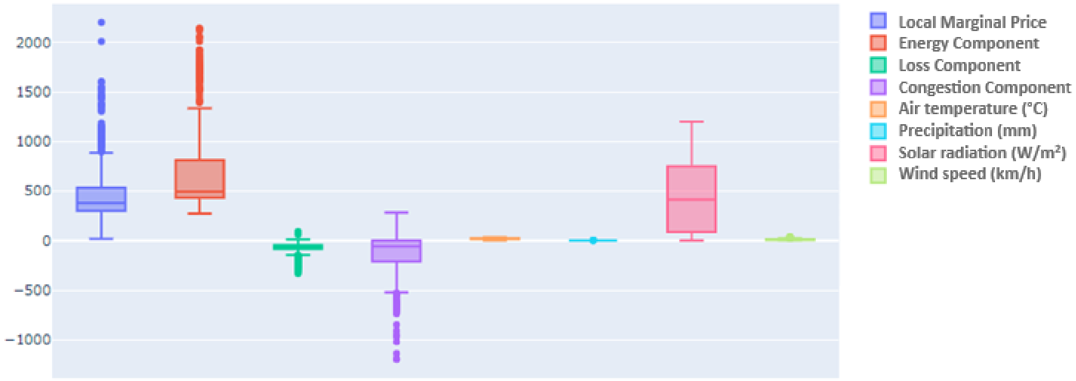


Fig. 7 Boxplot of Node 05PAR-115.

Figure 8 shows the seven independent variables concerning the dependent variable (LPM), and we observe a box plot that, with the Python environment, is interactive because it represents the statistics of each variable by simply hovering over each one. In general terms, we can observe that the LPM of the Parral node is relatively constant in its values within 0 to 900 pesos per megawatt-hour. Still, the points outside the quartiles indicate the registered peaks that exceed the normal limits of the variable. This behavior is present in the LPM variables (congestion, energy, and loss) but not in the climatological variables. These outliers represent a challenge in the forecast sought in the future. Once their statistics are identified, a multiple correlation of the variables is carried out, see Figure 9:

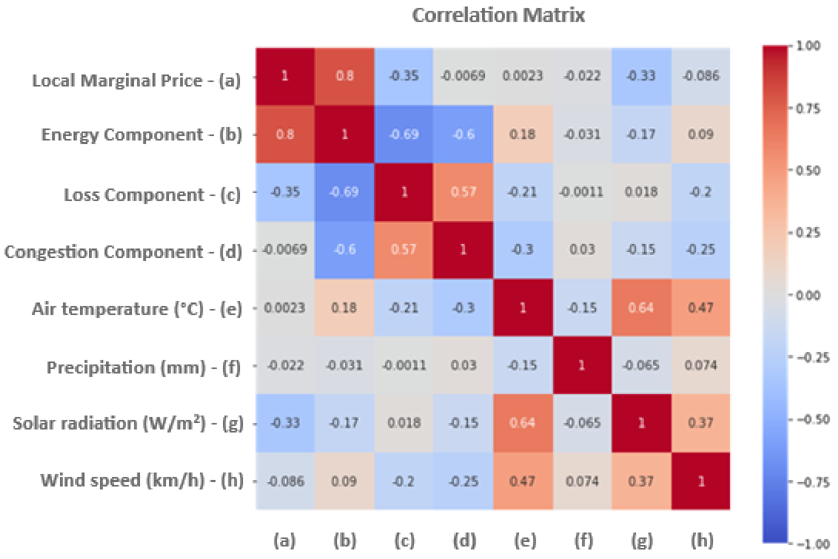


Fig. 8 Multiple Correlation Diagram of Node 05PAR-115.

The correlation matrix (Figure 9) shows the linear relationships between the variables included in the analysis. Correlation values range from -1 to 1, where 1 indicates a perfect positive correlation, -1 a perfect negative correlation, and 0 an absence of correlation:

- Local Marginal Price shows a moderate positive correlation with Energy Component (0.802) and negative correlations with Loss Component (-0.354) and Solar Radiation (-0.311).
- Energy Component exhibits a moderate positive correlation with Local Marginal Price (0.802) and negative correlations with Loss Component (-0.690) and Congestion Component (-0.599).
- Loss Component has a moderate negative correlation with the Energy Component (-0.690) and a positive correlation with the Congestion Component (0.571).
- Congestion Component shows moderate negative correlations with Energy Component (-0.599) and Air Temperature (-0.308).
- Air Temperature has a moderate positive correlation with Solar Radiation (0.620).
- Solar Radiation" exhibits a moderate negative correlation with Loss Component (-0.164).

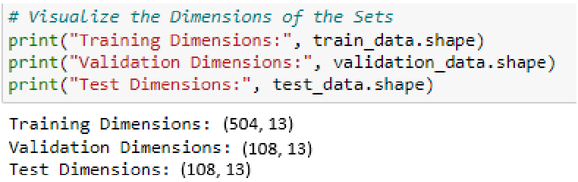
These correlations provide insights into how these variables may mutually influence each other in the context of analysis. Due to the sensitivity of the models, data imputation should be performed following the statistical analysis, such as feature engineering in the case of LSTM networks. This may involve applying a sine function for the continuity of cyclic daily data and normalization to control the outliers observed in the Boxplot.

4.4 Data Imputation

Following the statistical analysis, data imputation was performed on the dataset of node 05PAR-115 in Hidalgo del Parral, Chihuahua, which is a crucial step before feeding the data into the machine learning forecasting model, as missing data can impact the forecast. During this process, various imputation strategies were applied tailored to the nature of the variables. For precipitation, missing values were replaced with 0.0, considering the absence of precipitation during those periods as a reasonable assumption in the meteorological time series. The air temperature was imputed with the mean of the recorded temperatures, considering that this variable tends to vary around an average with seasonal and daily fluctuations. Additionally, the median of wind speed was used as an imputation for missing values in this variable, given its robustness against extreme values. During nighttime hours, missing solar radiation data were assigned values of 0.0 since there was no solar radiation during this period. For missing values, the KNN (K-Nearest Neighbors) imputation method was implemented with five as the number of nearest neighbors to capture complex relationships in the data. After the imputation process, the integrity of the dataset was verified to validate the effectiveness of the process, resulting in a dataset prepared for further analysis.

4.4 Decision Tree for Regression (ADR)

The preprocessing step is crucial for Machine Learning models, and the results obtained in the Decision Trees algorithm have a significant impact on this research due to its contribution not only to future forecasting but also to its logic of data division, subjecting them to a decision criterion based on the forecast of the PML. Its implementation has allowed for a deeper and more accurate understanding of the forecast, adding a layer of detail in this context that had not been explored before. The cardinality of each subset is visualized in Figure 10; in this case, the dataset is divided into a training set with 504 samples, a validation set with 108 samples, and a test set with 108 samples (504+108+108=720). The number 13 in each corresponding set denotes the 13 features (columns).



```
# Visualize the Dimensions of the Sets
print("Training Dimensions:", train_data.shape)
print("Validation Dimensions:", validation_data.shape)
print("Test Dimensions:", test_data.shape)
```

```
Training Dimensions: (504, 13)
Validation Dimensions: (108, 13)
Test Dimensions: (108, 13)
```

Fig. 10 Cardinality of the subsets: training, validation, and testing.

Once the model is compiled, the generated tree is saved as an image. The graph of significant variables for the forecast determined by the decision tree (using the condition of which variables have the lowest MAE to approach the LPM variable) achieves the bar graph of all the variables, which complements and supports the correlation diagram shown. See Figure 11:

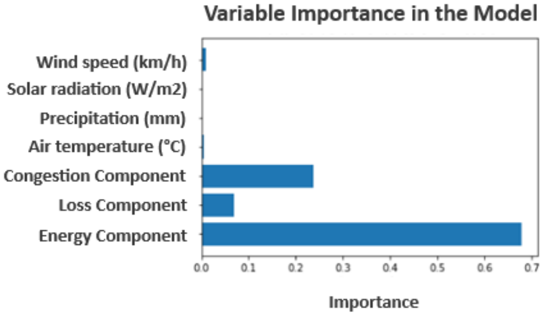


Fig. 11 Variables are ranked in order of importance by the ADR model.

The results from the graph of essential variables within the decision tree algorithm rank the energy component variable as the most significant in the forecast. It is worth noting that there are five other variables of relative importance also detected by the model: congestion component, loss component, wind speed, and air temperature, albeit to a lesser extent. This is highly useful as it allows for the application of neural networks to predict and evaluate simultaneously, significantly contributing to what the decision tree algorithm achieves.

4.5 Second Degree Polynomial Regression (RPSG)

The regression algorithm presents the model that captures nonlinear patterns in the data most effectively without falling into overfitting by excessively raising values within the model; it is the RPSG model that achieves metrics so close to 0.0 (See Figure 12) that it is necessary to apply model validation with time series split in addition to walkforward validation; Both approaches, time series split and walkforward validation, offer valuable insights. Time series split evaluates the model in different temporal divisions to obtain a more comprehensive view of performance over time. On the other hand, walk-forward validation simulates the network's ability to continuously adapt to new data, emulating a dynamic production environment. It uses training and validation windows that progress over time, thus representing a more dynamic simulation of real life. Retraining in each iteration and adjusting the network to more recent data, this approach aims to maintain the network's predictive capability in changing or "never seen before" situations. The same techniques were applied to all models for reliable metric validation. Transformer = Polynomial Features (degree=2): An instance of "Polynomial Features" is created to transform the features. A polynomial degree of 2 is specified, implying that features will be squared and interactions will be made between them. See Figure 12 for more details.:

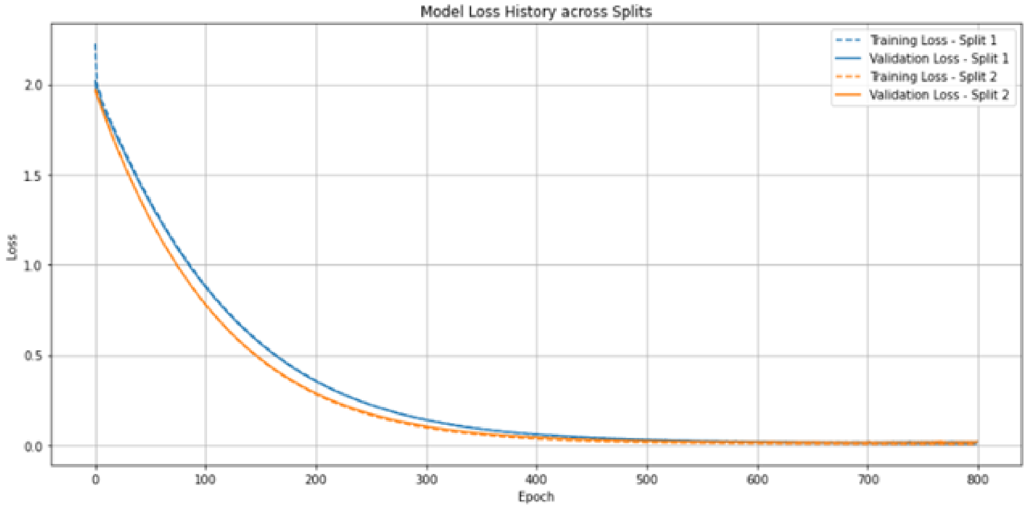
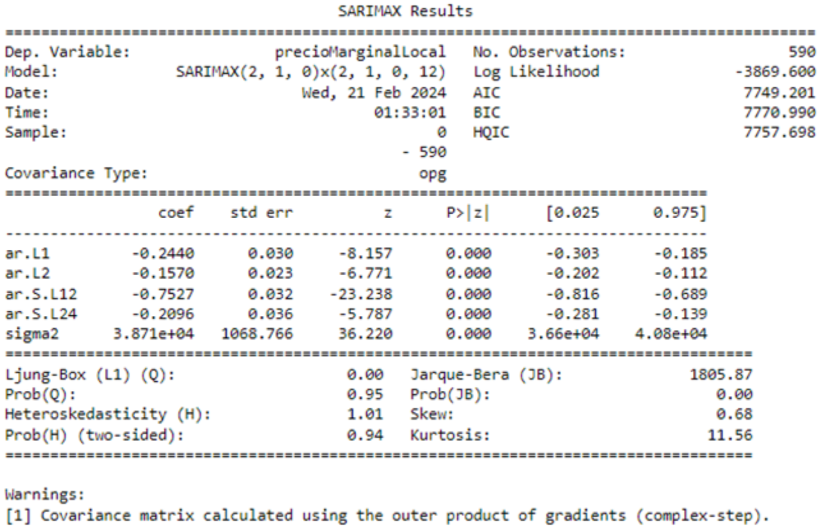


Fig. 12 Validation of the Polynomial Regression Model using time series split.

4.6 SARIMA

The SARIMA model includes automatic hyperparameter tuning and a rapid and comprehensive statistical analysis after compiling it and utilizing the stats model and Panda's libraries (see Figure 13).



```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                preciolMarginalLocal   No. Observations:                  590
Model:             SARIMAX(2, 1, 0)x(2, 1, 0, 12)   Log Likelihood               -3869.600
Date:                            Wed, 21 Feb 2024   AIC                           7749.201
Time:                                    01:33:01   BIC                           7770.990
Sample:                                         0   HQIC                          7757.698
                                            - 590
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.2440      0.030     -8.157      0.000      -0.303      -0.185
ar.L2         -0.1570      0.023     -6.771      0.000      -0.202      -0.112
ar.S.L12      -0.7527      0.032    -23.238      0.000      -0.816      -0.689
ar.S.L24      -0.2096      0.036     -5.787      0.000      -0.281      -0.139
sigma2      3.871e+04   1068.766     36.220      0.000    3.66e+04    4.08e+04
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):              1805.87
Prob(Q):                              0.95   Prob(JB):                         0.00
Heteroskedasticity (H):               1.01   Skew:                             0.68
Prob(H) (two-sided):                  0.94   Kurtosis:                        11.56
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

Fig. 13 Statistical Analysis of the SARIMA Model.

This analysis presents the results of the SARIMAX (2, 1, 0) (2, 1, 0, 12) model applied to the local marginal price data series. This indicates:

ARIMA (2,1,0):

- AR (Autoregressive) = 2: The model utilizes three autoregressive terms.
- I (Integrated) = 1: There is differencing to make the time series stationary.
- MA (Moving Average) = 0: No moving average terms are used.

Seasonal Component (2,1,0) [12]:

- Seasonal AR = 2: There are two autoregressive terms in the seasonal part of the model.
- Seasonal Differencing = 1: Seasonal differencing is performed.
- Seasonal MA = 0: No moving average terms are used in the seasonal part.
- [12]: The seasonal pattern occurs every 12 hours.

1. **AR and SAR Coefficients**: The AR (Autoregressive) and SAR (Seasonal Autoregressive) coefficients indicate the linear relationship between observed values and their past values, both at regular and seasonal levels. In this case, the AR(L1) and AR(L2) coefficients are significant and negative, suggesting an inverse relationship between past and present values. The SAR(L12) and SAR(L24) coefficients are also negative and significant, indicating a seasonal influence on the marginal prices.
2. **Sigma2**: This value represents the model's residual variance, i.e., the difference between observed values and values predicted by the model. In this case, the residual variance is approximately 3.871e+04, indicating that the model explains significant variability in the data. However, there may still be some level of unexplained error.
3. **Goodness-of-Fit Tests**: The Ljung-Box and Jarque-Bera tests evaluate the model's goodness of fit. In this case, the Ljung-Box test's Q value suggests no significant autocorrelation in the model residuals, indicating a good fit. However, the Jarque-Bera test value implies that the distribution of residuals may not be expected, which could indicate some deviation from the normality assumption in the data.
4. **Heteroscedasticity and Skewness**: Heteroscedasticity refers to the non-constant variation of errors over time, while skewness relates to the asymmetry of the error distribution. In this case, heteroscedasticity and skewness appear insignificant, suggesting that errors may be relatively constant and symmetrically distributed.

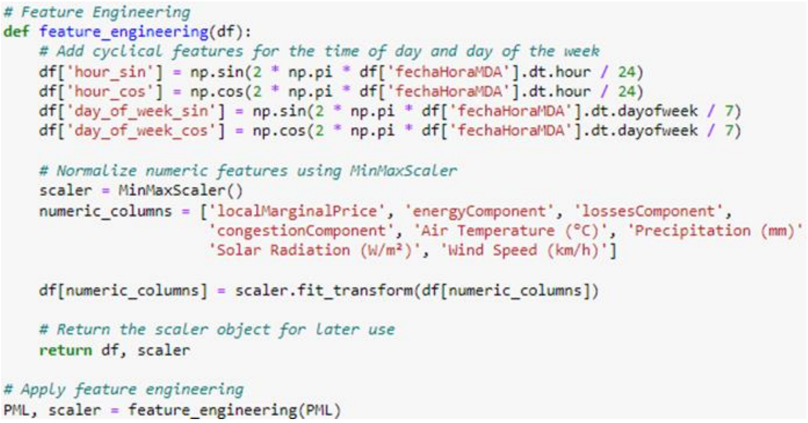
4.7 Hybrid LSTM (LSTM-H)

Additionally, when building a forecasting model, it is beneficial to employ feature engineering techniques to capture nonlinear relationships and interactions between variables. Applying transformations such as sine and cosine to temporal features is a common feature engineering strategy aimed at identifying cyclic or sinusoidal patterns in the data. The rationale behind this is that many of these temporal features have a cyclical nature. By applying the sine and cosine functions to the time of day, we transform a linear variable into two new features that better capture the periodicity and cyclical continuity of the data. For example, on a 24-hour time - scale, the jump from 23 to 0 creates a discontinuity that does not reflect the cyclical reality of time. The sine and cosine functions smooth out this jump, representing 23:00 and 00:00 as points close in the cycle, thus providing two new features that capture the daily cycle of the original hour. The transformations can be expressed as:

$$hour\ \sin = \sin(\frac{2\pi * hour}{24}) . \tag{1}$$

$$hour\ \cos = \cos(\frac{2\pi * hour}{24}) . \tag{2}$$

These transformed features will have values ranging from -1 to 1 and describe a unit circle as the day progresses. See Figure 14.



```
# Feature Engineering
def feature_engineering(df):
    # Add cyclical features for the time of day and day of the week
    df['hour_sin'] = np.sin(2 * np.pi * df['fechaHoraMDA'].dt.hour / 24)
    df['hour_cos'] = np.cos(2 * np.pi * df['fechaHoraMDA'].dt.hour / 24)
    df['day_of_week_sin'] = np.sin(2 * np.pi * df['fechaHoraMDA'].dt.dayofweek / 7)
    df['day_of_week_cos'] = np.cos(2 * np.pi * df['fechaHoraMDA'].dt.dayofweek / 7)

    # Normalize numeric features using MinMaxScaler
    scaler = MinMaxScaler()
    numeric_columns = ['localMarginalPrice', 'energyComponent', 'lossesComponent',
                       'congestionComponent', 'Air Temperature (°C)', 'Precipitation (mm)'
                       'Solar Radiation (W/m²)', 'Wind Speed (km/h)']

    df[numeric_columns] = scaler.fit_transform(df[numeric_columns])

    # Return the scaler object for later use
    return df, scaler

# Apply feature engineering
PML, scaler = feature_engineering(PML)
```

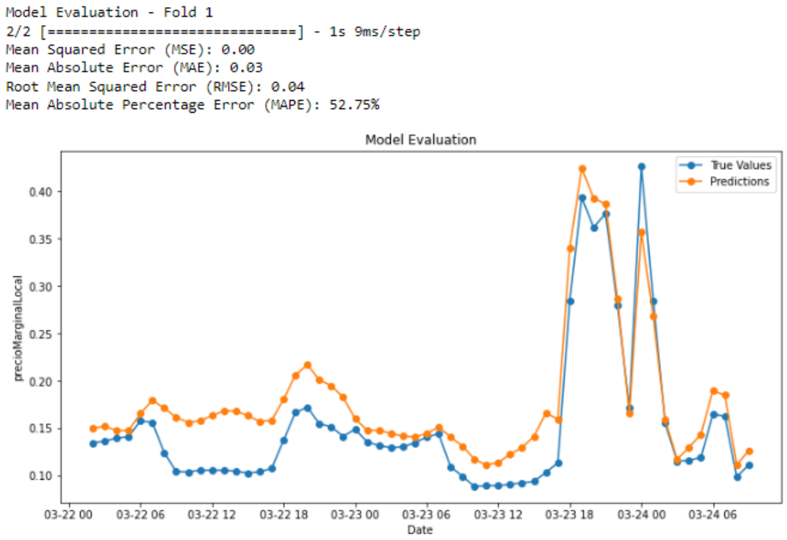
Fig. 14 Feature Engineering Application Commands to the LPM Dataset.

Additionally, MinMaxScaler is applied for data normalization between 0 and 1, as it helps ensure that all features contribute equally to the model without one feature dominating due to its more extensive scale. The normalization process follows the formula:

$$X_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} . \tag{3}$$

These transformations can lead to improved performance of the models we use to forecast PML energy, but it is always important to validate the impact of these transformations on model performance through cross-validation tests for time series.

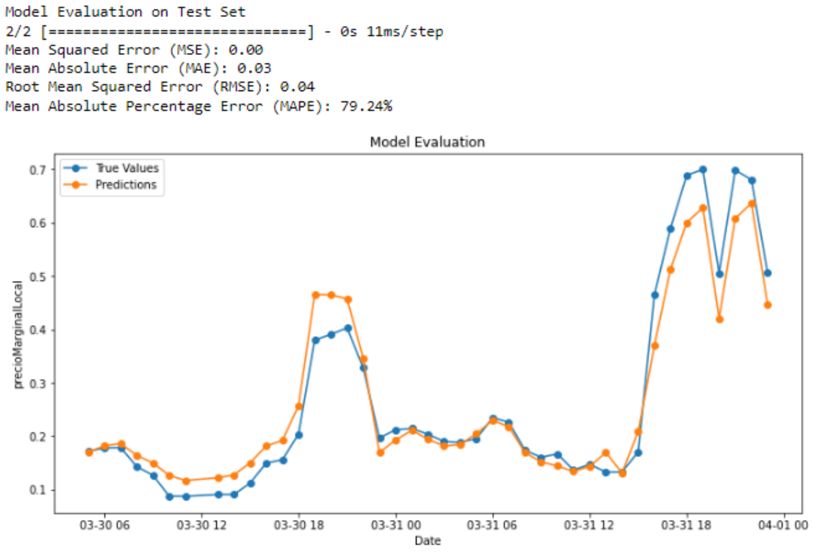
The hybrid model where "time series split = 3" is applied, so the results shown below are satisfactory, not only at node 05PAR-115, because similar results were also obtained at all nodes of the implementation. See Figures 15 and 16:



```
Model Evaluation - Fold 1
2/2 [==============================] - 1s 9ms/step
Mean Squared Error (MSE): 0.00
Mean Absolute Error (MAE): 0.03
Root Mean Squared Error (RMSE): 0.04
Mean Absolute Percentage Error (MAPE): 52.75%
```

Fig. 15 Model Validation on the First Split

In the graph shown in Figure 16, the network's generalization to the actual values is nearly perfect, with its metrics supporting that the forecast is performing well.



```
Model Evaluation on Test Set
2/2 [==============================] - 0s 11ms/step
Mean Squared Error (MSE): 0.00
Mean Absolute Error (MAE): 0.03
Root Mean Squared Error (RMSE): 0.04
Mean Absolute Percentage Error (MAPE): 79.24%
```

Fig. 16 Model Validation on the Test Set.

The obtained MAPEs between the training folds and the test set suggest that the MAPEs are very high in terms of predicting future values or unseen data. This situation is natural, as there are multiple possibilities for prices, and only hundreds of them can be learned from the past to infer how the future ones will be. Moreover, learning from past data does not guarantee accuracy in predicting the future due to inherent differences between both datasets. Therefore, it is not surprising that the model's performance in predicting unseen values is limited, especially in complex and unstable systems like the energy market. However, it is useful to examine the forecasts and evaluate how the model generalizes to gain a comprehensive perspective along with other evaluation metrics.

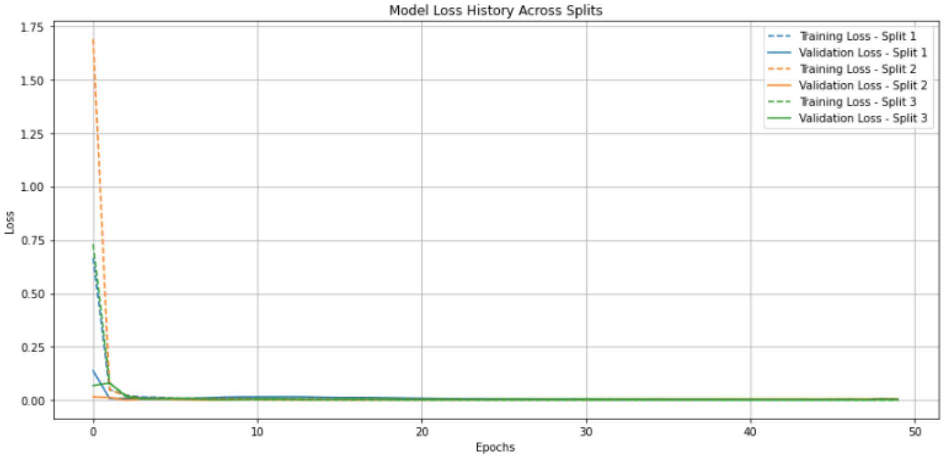
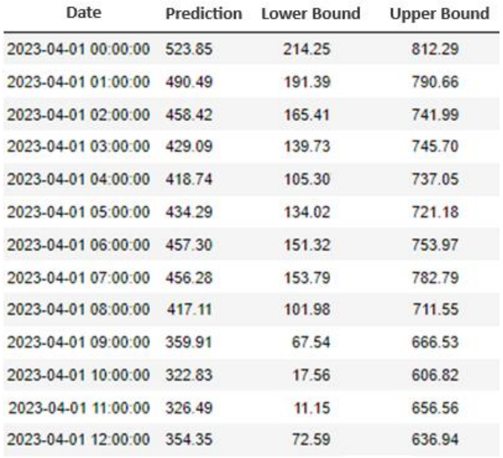


Fig. 17 Model Validation of LSTM Model.

In Figure 17, it is observed that each training fold reached a mean squared error (MSE) close to zero even before reaching 10, indicating that the model has effectively learned from the training data and has appropriately generalized to make accurate forecasts with real data. The forecasts for the next 12 hours are shown in Table 1.

Table 1 Forecast List from LSTM-H Model.



Date	Prediction	Lower Bound	Upper Bound
2023-04-01 00:00:00	523.85	214.25	812.29
2023-04-01 01:00:00	490.49	191.39	790.66
2023-04-01 02:00:00	458.42	165.41	741.99
2023-04-01 03:00:00	429.09	139.73	745.70
2023-04-01 04:00:00	418.74	105.30	737.05
2023-04-01 05:00:00	434.29	134.02	721.18
2023-04-01 06:00:00	457.30	151.32	753.97
2023-04-01 07:00:00	456.28	153.79	782.79
2023-04-01 08:00:00	417.11	101.98	711.55
2023-04-01 09:00:00	359.91	67.54	666.53
2023-04-01 10:00:00	322.83	17.56	606.82
2023-04-01 11:00:00	326.49	11.15	656.56
2023-04-01 12:00:00	354.35	72.59	636.94

The predictions include three relevant columns for confidence intervals, where the forecasts generated by the hybrid LSTM model for the Local Marginal Price (LMP) are displayed, with each row representing a prediction for a specific hour, and the columns providing the corresponding date, the LMP prediction, as well as the lower and upper bounds of the prediction. See Figure 18:

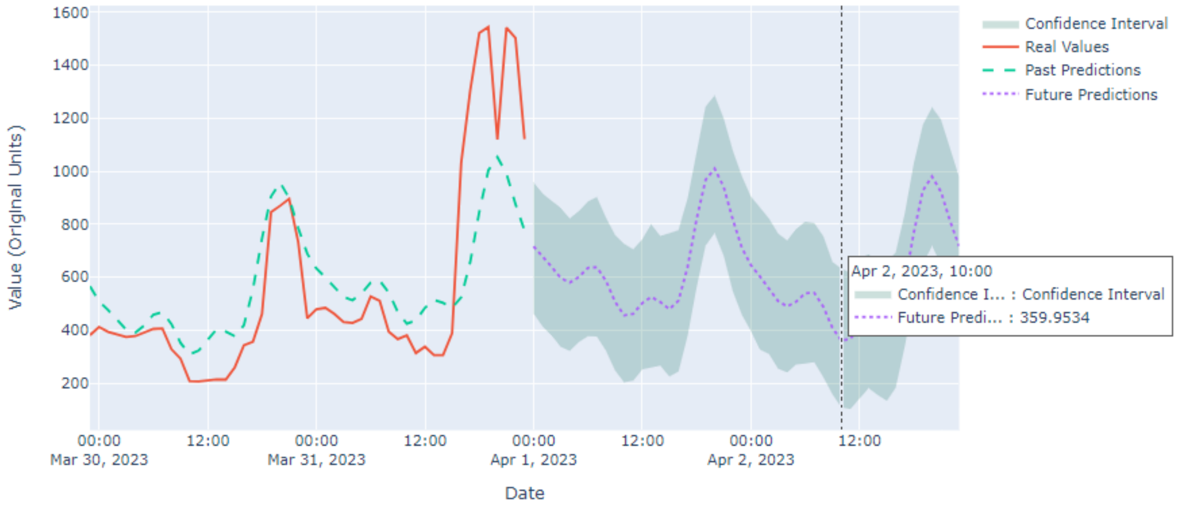


Fig. 18 Graph of LPM predictions for node 05PAR-115 in Hidalgo del Parral, Chihuahua, (SIN).

- **Date:** Indicates the date and time for which the prediction was made.
- **Prediction:** Represents the estimated value of the LPM for the corresponding hour.
- **Lower Limit:** Indicates the lower limit of the confidence interval associated with the LPM prediction. This value provides an estimate of the model's variability, and it is expected that the actual value of the LPM will be above this limit with a 95% confidence level.
- **Upper Limit:** This represents the upper limit of the confidence interval associated with the LPM prediction. Similar to the lower limit, this value sets the range within which the actual value of the LPM is expected to be with a 95% confidence level.

4.8 Hyperparameter Sensitivity Analysis.

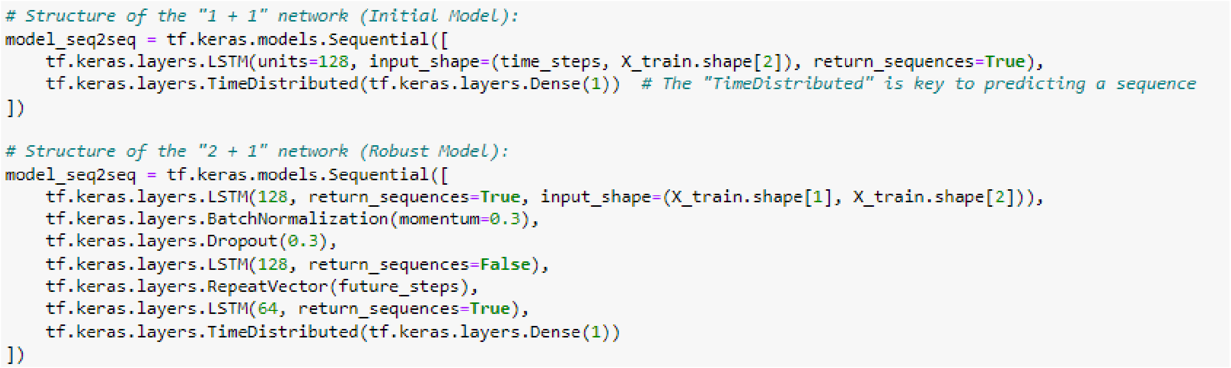
The purpose is to evaluate variations in network complexity by adding layers and adjusting hyperparameters, and how they affect the model's ability to learn more intricate patterns and make more accurate forecasts. This strategy aims to determine if a more robust and complex neural network can significantly improve the generalization capability and prediction accuracy compared to the initial model.

The neural network identified as "1 + 1" is an architecture consisting of an LSTM layer with 128 units that returns sequences instead of a single value. The Time Distributed layer is used to apply the dense layer at each time step.

On the other hand, the neural network identified in this analysis as "2 + 1" has a more complex architecture: It includes two LSTM layers (with 128 and 64 units, respectively), Batch Normalization and Dropout layers, and utilizes the Repeat Vector technique to repeat the output before feeding it into another LSTM layer in the first two layers. Additionally, the Time Distributed layer is applied to the output as well.

The inclusion of Batch Normalization and Dropout layers in the "2 + 1" architecture serves a specific purpose in improving the model's robustness. The Batch Normalization layer helps stabilize training by normalizing intermediate activations during the learning process, which can help avoid issues like gradient vanishing.

On the other hand, the Dropout layer introduces regularization by randomly turning off neurons during training, thus reducing the likelihood of overfitting and improving the model's generalization. Additionally, the Repeat Vector technique is employed to repeat the output of the LSTM layers before feeding it into another LSTM layer. This repetition provides additional contextual information to the model, allowing for a richer representation of the data sequences. Finally, the Time Distributed layer applied to the output ensures that predictions are made for each time step, providing coherent sequential output. See the structure clearly in Figure 19:



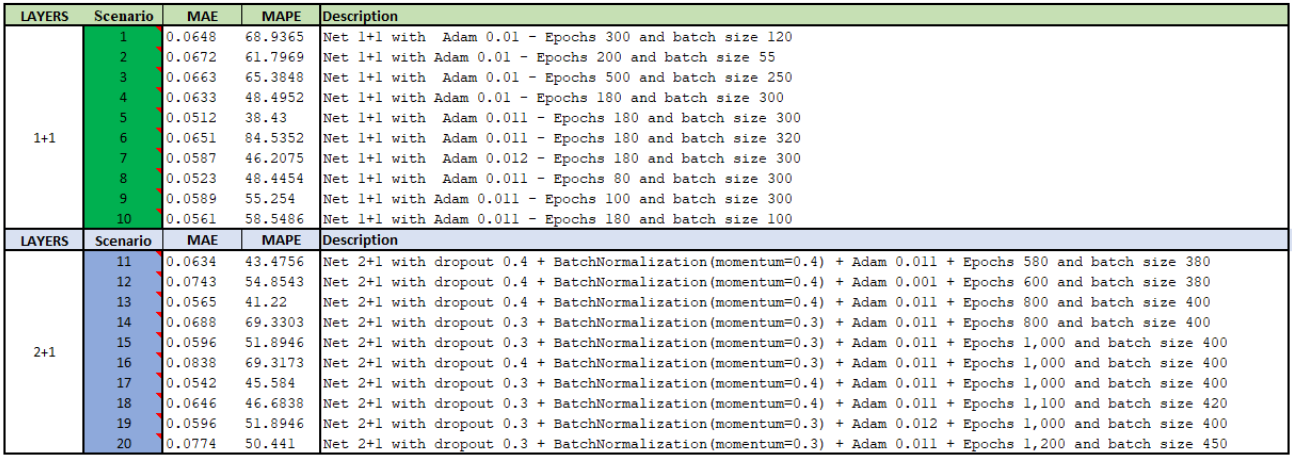
```
# Structure of the "1 + 1" network (Initial Model):
model_seq2seq = tf.keras.models.Sequential([
    tf.keras.layers.LSTM(units=128, input_shape=(time_steps, X_train.shape[2]), return_sequences=True),
    tf.keras.layers.TimeDistributed(tf.keras.layers.Dense(1))  # The "TimeDistributed" is key to predicting a sequence
])

# Structure of the "2 + 1" network (Robust Model):
model_seq2seq = tf.keras.models.Sequential([
    tf.keras.layers.LSTM(128, return_sequences=True, input_shape=(X_train.shape[1], X_train.shape[2])),
    tf.keras.layers.BatchNormalization(momentum=0.3),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.LSTM(128, return_sequences=False),
    tf.keras.layers.RepeatVector(future_steps),
    tf.keras.layers.LSTM(64, return_sequences=True),
    tf.keras.layers.TimeDistributed(tf.keras.layers.Dense(1))
])
```

Fig. 19 Architectures of the 2 LSTM Networks Undergoing Sensitivity Analysis.

The following Table 2 provides a clear overview of the adjustments made for the sensitivity analysis. There are 10 scenarios with a basic network and 10 scenarios with a double-layer network and a single output neuron. Each scenario represents a specific parameter or condition configuration, totaling 20 different combinations, each representing a unique instance of the parameters or conditions explored in the study. The results were recorded for critique and subsequent analysis.

Table 2 Description of Architectures and Sensitivity Analysis Scenarios



LAYERS	Scenario	MAE	MAPE	Description
1+1	1	0.0648	68.9365	Net 1+1 with Adam 0.01 - Epochs 300 and batch size 120
	2	0.0672	61.7969	Net 1+1 with Adam 0.01 - Epochs 200 and batch size 55
	3	0.0663	65.3848	Net 1+1 with Adam 0.01 - Epochs 500 and batch size 250
	4	0.0633	48.4952	Net 1+1 with Adam 0.01 - Epochs 180 and batch size 300
	5	0.0512	38.43	Net 1+1 with Adam 0.011 - Epochs 180 and batch size 300
	6	0.0651	84.5352	Net 1+1 with Adam 0.011 - Epochs 180 and batch size 320
	7	0.0587	46.2075	Net 1+1 with Adam 0.012 - Epochs 180 and batch size 300
	8	0.0523	48.4454	Net 1+1 with Adam 0.011 - Epochs 80 and batch size 300
	9	0.0589	55.254	Net 1+1 with Adam 0.011 - Epochs 100 and batch size 300
	10	0.0561	58.5486	Net 1+1 with Adam 0.011 - Epochs 180 and batch size 100
LAYERS	Scenario	MAE	MAPE	Description
2+1	11	0.0634	43.4756	Net 2+1 with dropout 0.4 + BatchNormalization(momentum=0.4) + Adam 0.011 + Epochs 580 and batch size 380
	12	0.0743	54.8543	Net 2+1 with dropout 0.4 + BatchNormalization(momentum=0.4) + Adam 0.001 + Epochs 600 and batch size 380
	13	0.0565	41.22	Net 2+1 with dropout 0.4 + BatchNormalization(momentum=0.4) + Adam 0.011 + Epochs 800 and batch size 400
	14	0.0688	69.3303	Net 2+1 with dropout 0.3 + BatchNormalization(momentum=0.3) + Adam 0.011 + Epochs 800 and batch size 400
	15	0.0596	51.8946	Net 2+1 with dropout 0.3 + BatchNormalization(momentum=0.3) + Adam 0.011 + Epochs 1,000 and batch size 400
	16	0.0838	69.3173	Net 2+1 with dropout 0.4 + BatchNormalization(momentum=0.3) + Adam 0.011 + Epochs 1,000 and batch size 400
	17	0.0542	45.584	Net 2+1 with dropout 0.3 + BatchNormalization(momentum=0.4) + Adam 0.011 + Epochs 1,000 and batch size 400
	18	0.0646	46.6838	Net 2+1 with dropout 0.3 + BatchNormalization(momentum=0.4) + Adam 0.011 + Epochs 1,100 and batch size 420
	19	0.0596	51.8946	Net 2+1 with dropout 0.3 + BatchNormalization(momentum=0.3) + Adam 0.012 + Epochs 1,000 and batch size 400
	20	0.0774	50.441	Net 2+1 with dropout 0.3 + BatchNormalization(momentum=0.3) + Adam 0.011 + Epochs 1,200 and batch size 450

The result of the sensitivity analysis is depicted in Figure 24, which gathers the 20 scenarios to verify the differences between them and how they generalize with the real data represented in black in the graph. However, for better appreciation, the scenarios will be shown in two groups: those belonging to the "1 + 1" network in Figure 25 and those to the "2 + 1" network group in Figure 20.

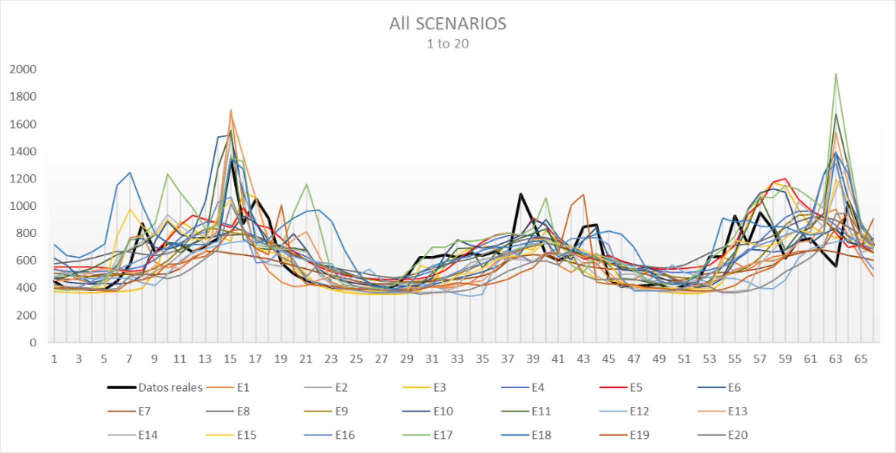


Fig. 20 Graph of all Sensitivity Analysis Scenarios.

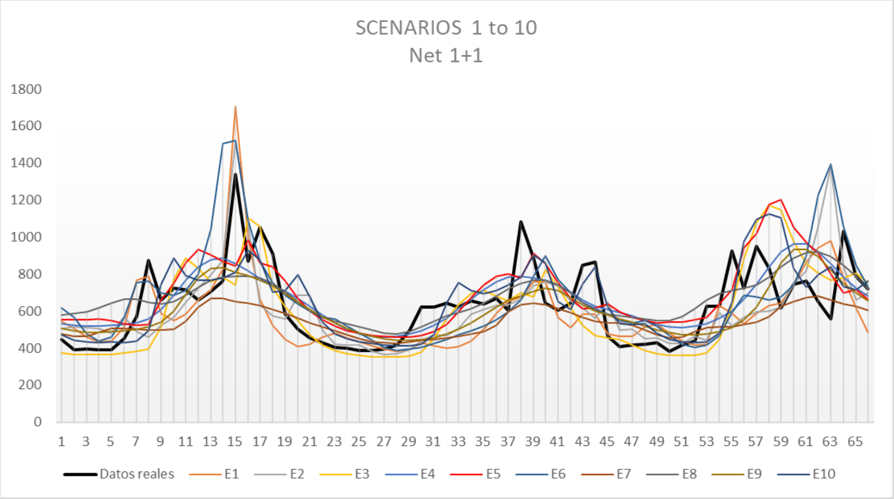


Fig. 21 Graph of Sensitivity Analysis for the 1+1 Network.

Figure 20 and 21 allows us to see more clearly those forecasts corresponding to the "1 + 1" network, where no scenario satisfactorily approaches to generalize the behavior of the data, but it is worth recognizing that the most similar scenarios are the 5th and 6th respectively. The real values are shown in black, and although all scenarios present some degree of similarity with the real ones, it is important to complement this general observation with the detailed metrics provided in the Table 2.

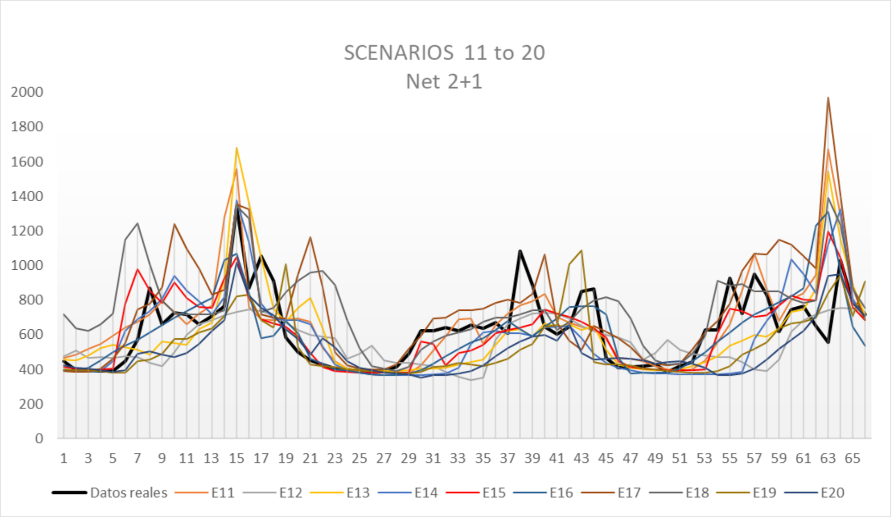


Fig. 22 depicts the Sensitivity Analysis of the 2+1 network.

On the other hand, when examining the 2-layer plus 1 network (2 + 1), represented graphically in Figure 26, it is observed that the scenario that most closely resembles the real values (represented in black) is scenario 15 (in red). However, this scenario shows a notable lag and limited efficiency at first glance. Compared to the "1 + 1" network, it is evident that the "2 + 1" network is even less efficient. The sensitivity analysis, on the other hand, primarily provides the range of hyperparameters in which the network performs well, as extrapolating each hyperparameter to values outside those experienced in the scenarios only deteriorates the network's ability to replicate the hourly behavior pattern of the Local Marginal Price (LMP).

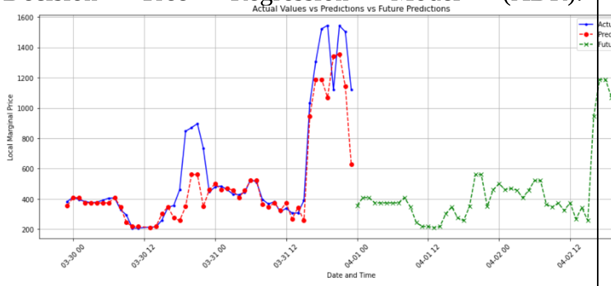
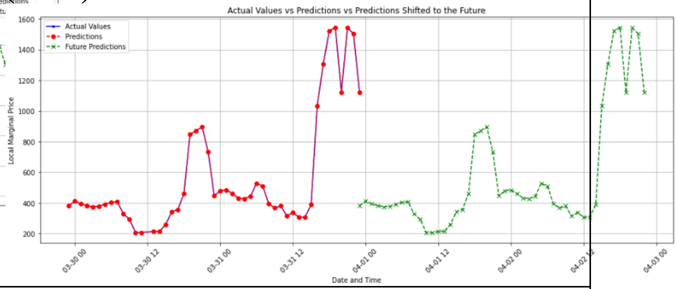
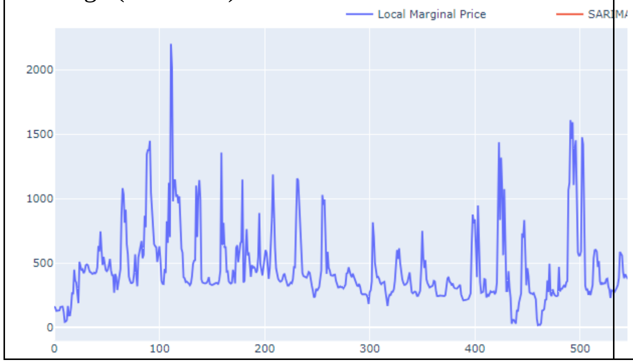
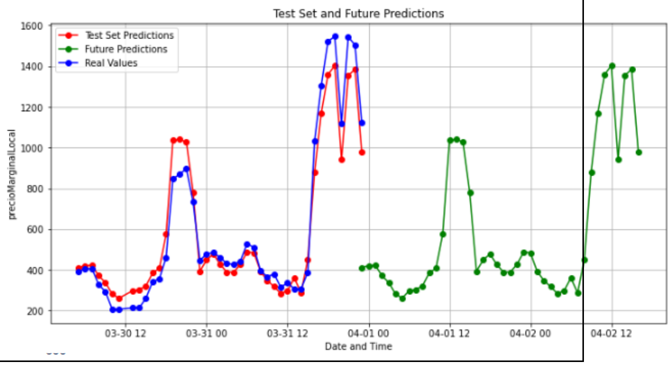
Table 2 summarizes the results of the sensitivity analysis where 20 different scenarios of the LSTM network are compared based on two metrics: Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). Here's a summary based on the data obtained from the 20 test scenarios:

- Value Ranges: MAE: The minimum MAE is 0.0512 (scenario 5) and the maximum is 0.0838 (scenario 16). MAPE: The minimum MAPE is 38.43 (scenario 5) and the maximum is 84.535 (scenario 6).
- Best Performance: Scenario 5 has the lowest MAE (0.0512) and the lowest MAPE (38.43), indicating the best overall performance in both metrics. Its graph shows an acceptable generalization of its forecasts compared to the real data.
- Worst Performance: Scenario 16 has the highest MAE (0.0838), and the second worst is scenario 6 with the highest MAPE (84.5352), indicating the worst performance in terms of percentage error.

4.9 Comparative Table of Machine Learning Models

The debate over the best forecasting model for predicting time series data depends on factors like data nature, information availability, and considered variables. To comprehensively address this, a detailed comparison between different models is necessary. Besides numerical metrics, visual interpretation and a deep understanding of the data context are crucial. Therefore, a comparative evaluation that includes both metric and visual analysis is conducted. The generalization of a network is evident in the graphs of the forecasting models in Table 4, showing each model's approximation to the actual data, particularly the regression models (tree and polynomial). To project future values, coding was necessary because these regression models do not capture temporal patterns, resulting in straight or curved projections like the RPSG model. While these models fit closely to the actual data and provide acceptable metrics for seen data, SARIMA and LSTM-H are better for future predictions. These models capture behavior patterns and provide forecasts over time, outperforming the regression models. However, SARIMA tends upward due to its autoregressive component, leading to high forecast errors and deviation from actual values when predicting more than 24 hours ahead. In contrast, the LSTM-H model excels in generalization, projection, metrics, and behavior pattern, making it a robust and validated choice for better forecasts.

Table 3. Graphs of the forecasts obtained in each Machine Learning model, at node 05PAR-115, Parral, Chihuahua, SIN

Decision Tree Regression Model (ADR). 	Second Degree Polynomial Regression Model (RPSG). 
Seasonal Auto-Regressive Integrated Moving Average (SARIMA) Model. 	Long Short-Term Memory Hybrid (LSTM-H) Model. 

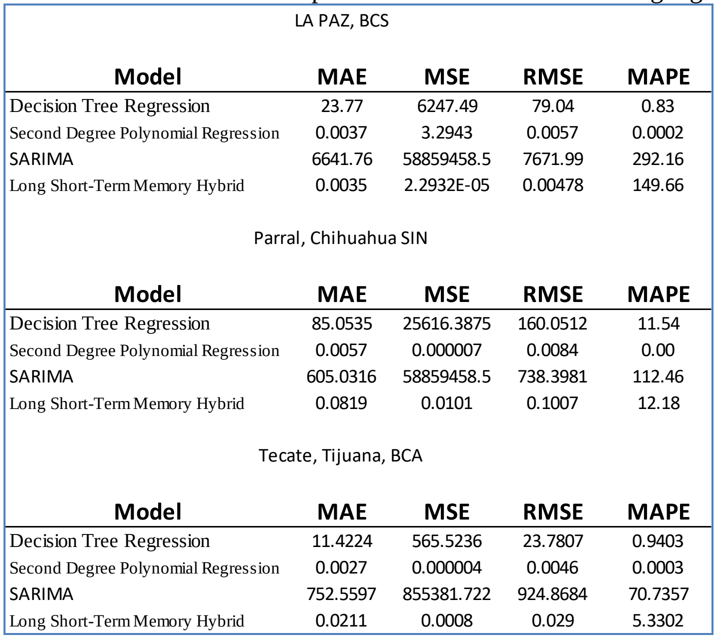
Results of the models applied to node 05PAR-115 in Hidalgo del Parral, Chihuahua, show significant variations in their predictive performance:

1. **Decision Tree Regression**: This model exhibits an MAE of 85.0535, indicating an average mean absolute error of approximately 85 in predictions. The MSE and RMSE are also relatively high, suggesting that predictions may be quite dispersed concerning the actual values. The MAPE of 11.54% indicates that, on average, predictions have a mean absolute percentage error of 11.54%.
2. **Second Degree Polynomial Regression**: This model seems to perform exceptionally well, with extremely low values for MAE, MSE, and RMSE, all very close to zero. This suggests a highly accurate predictive capacity tightly aligned with the training data. The MAPE of 0.00% indicates that predictions virtually have no percentage errors.
3. **SARIMA**: The SARIMA model shows less satisfactory performance compared to the other models, with an MAE of 605.0316 and an RMSE of 738.3981. These values are significantly higher than those of Decision Tree Regression and LSTM Hybrid, suggesting that predictions may be less accurate and more dispersed.
4. **LSTM Hybrid**: This model shows performance similar to Decision Tree Regression, with an MAE of 0.0819 and an RMSE of 0.1007. The MAPE of 12.18% indicates that predictions have a mean absolute percentage error of 12.18%, placing it at an intermediate level regarding predictive accuracy.

The results reveal that the decision tree regression model performs exceptionally on validation and test sets, standing out for its outstanding metrics. However, its predictive capacity is limited to only one step ahead due to its non-sequential nature, resulting in stagnant long-term forecasts. On the other hand, the second-degree polynomial regression shows exceptional fit to historical data, even outperforming the decision tree in certain aspects. However, its inability to capture temporality and generalize volatile values results in predictions that are simply extrapolations of past data. In contrast, the SARIMA model presents inferior performance in forecasts, showing significant errors compared to actual values. As for the LSTM, although it exhibits good overall performance, its high MAPE suggests difficulties in predicting accurately, possibly due to data volatility or the presence of zero values in critical variables such as precipitation and congestion. These considerations are crucial when interpreting the predictive capability of each model in dynamic and realistic scenarios.

Since the implementation is carried out across 28 nodes, to analyze the model's behavior in nodes located in different areas and systems of the country, below are the corresponding graphs for two specific nodes: node 07LPZ-115, located in La Paz, Baja California Sur (BCS), and node 07TCT-69, located in Tecate, Tijuana, belonging to the Baja California (BCA) system. See Table 7

Table 7. Metrics obtained in the implementation of the 4 forecasting algorithms



LA PAZ, BCS

Model	MAE	MSE	RMSE	MAPE
Decision Tree Regression	23.77	6247.49	79.04	0.83
Second Degree Polynomial Regression	0.0037	3.2943	0.0057	0.0002
SARIMA	6641.76	58859458.5	7671.99	292.16
Long Short-Term Memory Hybrid	0.0035	2.2932E-05	0.00478	149.66

Parral, Chihuahua SIN

Model	MAE	MSE	RMSE	MAPE
Decision Tree Regression	85.0535	25616.3875	160.0512	11.54
Second Degree Polynomial Regression	0.0057	0.000007	0.0084	0.00
SARIMA	605.0316	58859458.5	738.3981	112.46
Long Short-Term Memory Hybrid	0.0819	0.0101	0.1007	12.18

Tecate, Tijuana, BCA

Model	MAE	MSE	RMSE	MAPE
Decision Tree Regression	11.4224	565.5236	23.7807	0.9403
Second Degree Polynomial Regression	0.0027	0.000004	0.0046	0.0003
SARIMA	752.5597	855381.722	924.8684	70.7357
Long Short-Term Memory Hybrid	0.0211	0.0008	0.029	5.3302

5 Conclusion

This project presents an innovative perspective by uniquely combining LSTM neural networks and the Prophet model. This ingenious combination allows leveraging the strengths of both approaches, resulting in a novel and well-constructed model. The obtained metrics and the model's generalization capability support its effectiveness and reliability in predicting the Local Marginal Price (LMP), which can be compared with 3 models (ADR, RPSG, and SARIMA), as evidenced in this research.
Additionally, an intuitive user interface has been developed for data exploration, facilitating the analysis and understanding of results by users. However, the most notable achievement is the discovery of an AutoML (Automated Machine Learning) model using decision trees (ADR) to identify the most relevant variables for efficient LMP forecasting, thus providing a system that automates the machine learning process. This combined approach represents a significant advance in energy price forecasting research and promises to have a positive impact on the construction of efficient forecasting models. Some key conclusions are:

- Decision Trees for Variable Selection: The implementation of decision trees allowed for efficient recognition of the most relevant variables for PML forecasting. This machine learning technique facilitated the identification of key factors in a complex and multidimensional dataset.
- Comparative Forecasting of 4 Machine Learning Models in a Python environment that manages to represent the strengths and weaknesses of the models and their predictive capacity in real data and future data.

Overall, the application of machine learning techniques was crucial for addressing the inherent complexity of time series data and achieving substantial improvements in the accuracy of Local Marginal Price (LMP) forecasts, particularly decision trees like the LSTM-H model, determining very interesting forecasts and obtaining their models and code commands for obtaining graphs that significantly aid efficient analysis and forecasting. Sensitivity analysis is also important for obtaining viable ranges from which to adjust the described LSTM networks and the importance of properly adjusting the neural network architecture, with the addition of layers and precise adjustments to hyperparameters to improve the model's ability to learn intricate patterns and make more accurate forecasts.

Each model has its strengths and weaknesses, but the LSTM-H model emerges as a solid and versatile option for predicting local marginal prices in the energy market context, and its reliable construction followed machine learning guidelines for efficient forecasting, which evidently demonstrated its generalization in seen data as its ability to predict in the future, offering a balance between accuracy, generalization capability, and adaptation to new data.

References

Livas-García, A., May Tzuc, O., Cruz May, E., Tariq, R., & Torres, M. J. (2022). Forecasting of locational marginal price components with artificial intelligence and sensitivity analysis: A study under tropical weather and renewable power for the Mexican Southeast. *Electric Power Systems Research*, 206, 107793. https://doi.org/10.1016/j.epsr.2022.107793.

Ramos, A., Cortés, G., Latorre, J. M., & Cerisola, S. (2006). Análisis de la relación precio marginal y demanda de electricidad mediante conglomerados. *X Congreso de Ingeniería de Organización*. Valencia, España.

Singhal, D., & Swarup, K. (2011). Electricity price forecasting using artificial neural networks. *Elsevier Energy*, 36(8), 550-555. https://doi.org/10.1016/j.energy.2011.02.004.