

Article

Nearest Neighbors Time Series Forecaster Based on Phase Space Reconstruction for Short-Term Load Forecasting

Jose R. Cedeño González ^{1,†}, Juan J. Flores ^{1,*}, Claudio R. Fuerte-Esquivel ^{1,†} and Boris A. Moreno-Alcaide ^{2,†}

¹ División de Estudios de Posgrado, Facultad de Ingeniería Eléctrica, Universidad Michoacana de San Nicolás de Hidalgo, Morelia 58000, Mexico; rafael.cedeno@umich.mx (J.R.C.G.); claudio.fuerte@umich.mx (C.R.F.-E.)

² National Center for Energy Control, Secretary of Energy, Don Manuelito 32, Olivar de los Padres, Álvaro Obregón, Ciudad de México 01780, Mexico; alcaideboris@gmail.com

* Correspondence: juanf@umich.mx

† These authors contributed equally to this work.

Received: 14 August 2020; Accepted: 1 October 2020; Published: 13 October 2020



Abstract: Load forecasting provides essential information for engineers and operators of an electric system. Using the forecast information, an electric utility company's engineers make informed decisions in critical scenarios. The deregulation of energy industries makes load forecasting even more critical. In this article, the work we present, called Nearest Neighbors Load Forecasting (NNLF), was applied to very short-term load forecasting of electricity consumption at the national level in Mexico. The Energy Control National Center (CENACE—Spanish acronym) manages the National Interconnected System, working in a Real-Time Market system. The forecasting methodology we propose provides the information needed to solve the problem known as Economic Dispatch with Security Constraints for Multiple Intervals (MISCED). NNLF produces forecasts with a 15-min horizon to support decisions in the following four electric dispatch intervals. The hyperparameters used by Nearest Neighbors are tuned using Differential Evolution (DE), and the forecaster model inputs are determined using phase-space reconstruction. The developed models also use exogenous variables; we append a timestamp to each input (i.e., delay vector). The article presents a comparison between NNLF and other Machine Learning techniques: Artificial Neural Networks and Support Vector Regressors. NNLF outperformed those other techniques and the forecasting system they currently use.

Keywords: short-term load forecasting; machine learning; time series forecasting; nearest neighbors algorithm

1. Introduction

The continuing growth of electric power systems requires the development of load forecasting models to ensure the system's reliability in operation, planning, and maintenance decision-making. On the other hand, the load forecasting plays an essential role in electricity markets because participants in this market make their pricing and procurement decisions based on the future load level to be supplied [1].

Load forecasting can be divided, according to the forecasting tasks and horizons, into the following categories [2].

Long Term—uses horizons from one to ten years or even of several decades. This kind of forecast generally uses data with a granularity of weeks or months, and it provides crucial information for planning to expand generation, transmission, and distribution assets.

Medium Term—covers a period of one to twelve months. At this time scale, the electric power system's operation depends on elements that influence demand, such as new loads, seasonal variations, and the maintenance requirements of large consumers. Forecasting at this horizon uses the information granularity of hourly loads and aims to predict the peak loads of days. This information allows engineers to plan for plants' outage times for equipment maintenance, tests, and commissioning events.

Short Term—covers periods from one day to one week. This forecasting is crucial to assess whether planned operations for a day ahead will be within operating limits, or the system reliability could be threatened. This kind of forecasting provides engineers with information to schedule utility processes (e.g., generation, fuel purchases, maintenance, and security analysis). Engineers obtain substantial savings if exact load forecasts support their actions. Short-term load forecasting (STLF) is necessary for the economic scheduling of generation capacity.

Very Short Term—this category ranges from one minute to several dozen minutes. When dealing with load forecasts with short time leads, it is easier to monitor load frequency, providing the Energy Management System (EMS) information to perform economic dispatch. Other uses of this scheme are real-time control and security evaluation.

Several works on short-term load forecasting consider exogenous variables as inputs to their models (e.g., time factors, weather data, and possibly customers' classes). The medium- and long-term forecasts consider historical load and weather data, the customer type (residential, commercial, and industrial), the type of appliances, devices, or machines they use, and other characteristics (e.g., economic and demographic data). Other factors may include timestamps since there are essential differences in load between weekdays and weekends. Mondays and Fridays are adjacent to weekends; therefore, they may have different loads than the rest of the week. Holidays have different behavior and are more difficult to forecast since their occurrence does not follow any pattern. Weather conditions are one of the most important factors that influence the load. Various weather variables are essential for load forecasting (temperature and humidity are the most commonly used).

After closing a day-ahead market, an accurate load forecasting is of paramount importance for supporting operational decisions in intraday markets to secure the necessary generation-load balance, which is affected by unexpected changes in the power consumption [3]. Within this context, this forecasting permits a better calculation of the transmission capacities and the adjustments to power reserves, which has a direct impact on the purchase of electricity to comply with the day-ahead contracts [4].

In practice, a very-short-term load forecasting is of paramount importance in the daily operation of the Mexican Interconnected Power System (MIPS) operated by the National Center for Energy Control (in Spanish Centro Nacional de Control de Energía (CENACE)). Within this context, operators apply forecasting models to the real-time market, composed of three main stages:

- Real-Time Unit Assignment (RTUA). This stage is associated with the unit commitment, where the start-up and shut-down of power plants are determined. The forecasting process starts at every hour in the operation day, with a horizon of eight dispatch intervals, starting at the operation hour;
- Economic Dispatch with Security Constraints for Multiple Intervals (MISCED). At this stage, the power reserve for each power unit, local marginal prices, and ancillary service prices are also determined. This kind of forecast is performed with a horizon of four dispatch intervals, starting at the following interval;
- Economic Dispatch with Interval Security Constraint (EDISC). The purpose of this process is to determine the load for the next 5-min interval. This process must forecast the generation power base for electrical central units and economic variables; the availability of that information allows engineers and operators to make more informed decisions.

Hence, to ensure the reliable real-time operation of the electricity market, CENACE needs load forecasts for four dispatch intervals to use them in the MISCED process. Operators produce forecasts for each electric control area composing the MIPS and the total MIPS load, as illustrated in Figure 1.

In this process, dispatch intervals are fifteen minutes long, where operators execute the MISCED to plan the system's operation at the beginning of the interval. Hence, the forecaster has 10 min to forecast the system load level for the next dispatch interval.

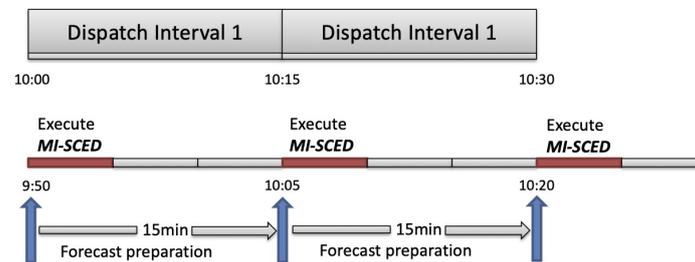


Figure 1. Short-Term Forecasting for Economic Dispatch.

Based on the information above, this paper proposes and implements a model for very-short term forecasting of electric load based on the concept of nearest neighbors, called nearest neighbors load forecasting (NNLF). The hyper-parameters used by Nearest Neighbors are tuned using Differential Evolution (DE), and the forecaster model inputs are determined by using phase-space reconstruction. The system trains the forecasting model when the forecasts are no longer within the required accuracy. The proposed model also uses exogenous variables; we append the week's day and reading time to each input (i.e., delay vector). We demonstrate the validity of the proposed model by forecasting a real load time series of the MIPS, comparing the obtained results with the ones associated with the current forecaster used by CENACE. We also compare NNLF's performance against other Machine Learning techniques: Artificial Neural Networks and Support Vector Regressors.

The remainder of the article is organized as follows: Section 2 presents state of the art in short-term load forecasting for electrical networks. Section 3 presents the forecasting methodology proposed in this article: Nearest Neighbor Load Forecasting. Section 4 presents the experiments we performed to test NNLF and the obtained results. Finally, Section 5 presents our conclusions and outlines future work.

2. Related Work

Artificial intelligence (AI) techniques have been gaining traction in recent years for load forecasting, given the results obtained by these techniques in different areas, such as classification, regression and clustering [5].

In STLF, there exists a wide array of AI-based models that have been implemented solely for this task. In this work, we identified two main approaches: univariate forecasting and multivariate forecasting. In univariate forecasting, the models only use demand reading at a particular time while ignoring other variables. In multivariate forecasting, the models' input includes additional variables besides the demand readings, mostly meteorological variables considered to have an influence on energy consumption [6].

Yang et al. [7] proposed an example of univariate forecasting. In their work, they proposed a gate-recurrent neural network for STLF with data sampled at 5, 20, 30, and 40 min and produced forecasts for one step ahead. Their paper's best results were for the 5-min sampling, where their method obtained a Mean Absolute Percentage Error (MAPE) of 0.49%.

In an attempt to characterize the demand by activity, Jain et al. [8] proposed a solution to forecast 48 h of electrical demand. Using clustering, they identified 13 different patterns; then, they produced 13 different Support Vector Machine (SVM) models that used the last 48 samples of the load, the average temperature, the forecasted temperature, and the week's day as input. With their method, they achieved an average MAPE error of 3%.

Dudek et al. [9] propose pattern-based local linear regression models for STLF. The theoretical basis of their work is that the load time series contains daily patterns. These patterns are encoded to

carry the information of the shape of the curves. They used a Principal Components Regressor (PCR) to produce the forecasts. The data used in their paper were sampled hourly. Their method obtained a MAPE of 1.15% when producing seven days of forecasts.

Khwaja et al. [10] propose a univariate approach. They propose a boosted neural network model. The boosting process trains a model, then, their output targets are replaced with the prior model targets for subsequent models. They repeated this process 25 times. They produced daily forecasts in hourly sampling for a year. The average MAPE obtained by their method was 1.43%.

Fan et al. proposes a univariate method based on Phase Space Reconstruction (PSR) and a bi-square kernel (BSK) regression model [11]. Data, expressed as delay vectors, attempt to reconstruct the load time series's attractor in phase space. These vectors are used by the BSK regressor to approximate the output. They produced results for one day and two weeks. Their model obtained an MAPE of 2.13% for one-day forecasts and 2.15% for two week forecasts.

Fan et al. [12] proposed a method that identified two types of day (regular and anomalous), then, by using a Self-Organizing Map (SOM) gating network, identified the type of day to forecast. Then, depending on the type of day, 24 SVMs (one for each hour) would forecast the day. The SOM network takes as an input the maximum load of the previous day, the average maximal daily load of the last week, the forecasted maximum temperature of the next day, the average maximum temperature of the previous two days, the forecasted maximum humidity of the next day, the forecasted average wind speed of the next day, the temperature sensitivity coefficient, a weekend indicator, and a holiday indicator. On the other hand, the SVMs of regular days use as input the hourly load and the hourly temperature of the last 168 observations of each variable in intervals of 24 h. For the anomalous day models, the SVMs use the hourly load in lags of 24 h of the last five days, the hourly load of the previous Saturday, the hourly load of the previous Sunday, and the hourly temperature of the last 168 observations in intervals of 24 h. The resulting model obtained an average MAPE of 2.3%.

Another example of a multivariate model is proposed by Liu et al. [13]. They propose a deep learning model with a stacked denoising auto-encoder. An auto-encoder is a type of non-recurrent neural network that attempts to reconstruct the inputs, instead of approximating a target value. The stacked denoising auto-encoder adds noise to the inputs to improve its performance. Their datasets are composed of the electrical load, relative humidity, and temperature. In their experiments, they produced a week of forecasts averaging 2.88% MAPE.

Ryu et al. [14] proposed a deep neural network with stacked restricted Boltzmann machine (RBM) layers. The RBM is a type of artificial network that can learn a probability distribution based on the inputs. They used a multivariate approach, where the inputs of the model were the last $24n$ readings of the load (sampled hourly), a day-of-the-week indicator of the last $2n$ days, five weather variables (temperature, humidity, wind speed, solar radiation, cloud cover), and four date labels (season, month, day of the week, and weekday indicator). Their model obtained 2.19% MAPE when forecasting one hour for 19 days.

Yu Ding [15] explores several models applied to wind speed and power forecast. Although the forecasting tasks stated in his book are also short-term, we obtain different results regarding what models produce the most accurate forecasts. A forecasting model's performance depends on the dynamics of the underlying system that produced the time series. In our study, we found that ANN produces better forecasting results than the persistent and ARIMA models, which is not the case in his study.

3. Materials and Methods

This section discusses the theory underlying NNLF—the proposed method and statistical analysis applied to the data. It starts by defining the concepts required to formulate the Nearest Neighbors Algorithm for Load Forecasting (NNLF). We then present a brief statistical analysis of the data using the Auto-Correlation Function and the Dickey–Fuller test. Finally, we discuss an exploration of the reconstructed attractor of the time series.

3.1. Phase Space Reconstruction

To model deterministic dynamical systems, the concept of phase space is often used; this is a space where all the possible system states are present [16].

For naturally occurring chaotic dynamical systems (that is, any time-dependant system that is sensitive to initial conditions that has not been specifically formulated, such as climate/weather, the stock market, and fluid flow), the phase space order and mathematical description of the system are often unknown. Given this uncertainty, Takens [17] formulated an embedding theorem to provide the conditions to reproduce any unknown dynamical system (Theorem 1).

Theorem 1 (Takens Embedding Theorem). *Let M be a compact manifold of dimension m . For pairs (ϕ, y) , $\phi : M \rightarrow M$ a smooth diffeomorphism (this is an invertible function that maps one differentiable manifold to another such that both the function and its inverse are smooth) and $y : M \rightarrow \mathbb{R}$ a smooth function, this is a generic property where the map $\Phi_{(\phi, y)} : M \rightarrow \mathbb{R}^{2m+1}$ defined by*

$$\Phi_{(\phi, y)}(x) = [y(x), y(\phi(x)), \dots, y(\phi^{2m}(x))] \quad (1)$$

is an embedding.

The proofs of Takens theorem can be seen in [17,18].

Takens's theorem allows us to observe one variable of a dynamical system and infer the system's behavior by reconstructing its attractor. To perform this inference, we must choose an appropriate τ value to make this reconstruction as close as possible. Many techniques exist to infer this τ such as the mutual information method or obtaining the mean period (\bar{T}) of the time series.

Methods based on phase space reconstruction have been developed to infer the future system states and, as a consequence, develop new predictive models. One (or more) variable from the system is observed as a function of time (i.e., a time series) and then used to construct mathematical components based on the observed states [16].

However, it is difficult to determine the optimal value of τ that unfolds the reconstructed attractor in practice. If the dynamical system's order is unknown, determining the optimal dimension (m) presents a challenge. Both issues are considered as an open problem. In this paper, as a part of our proposal, we offer a solution to this problem by using Differential Evolution in combination with the Nearest Neighbors Algorithm.

3.2. Nearest Neighbors Algorithm

Nearest Neighbors (NN) is one of the oldest and simplest machine-learning algorithms. NN uses the idea that similar sequences (measured by the distance between the evaluation sequence and the rest of the set) or instances will generate similar results. Ref. [19] presents a forecaster for a chaotic time series based on NN. Ref. [20] describes Nearest Neighbors as a simple non-linear time series forecaster used to forecast diverse problems that present a chaotic structure. While the use of nearest neighbors in classification problems is ample, the interest over it declined in recent years in time series forecasting. This change in focus can be attributed to the emergence of Artificial Neural Networks (ANNs) as a viable forecasting method. However, we consider that Nearest Neighbors combined with Differential Evolution (DE) can be advantageous, since the number of parameters to optimize is smaller than that of ANN, and the computational cost required is lower.

To obtain Δn forecasts, that is, $[s_{N+1}, s_{N+2}, \dots, s_{N+\Delta n}]$, the algorithm will employ any observation available in S .

By using a time delay τ and an embedding dimension m , it is possible to build delay vectors of the form $S_t = [s_{t-(m-1)\tau}, s_{t-(m-2)\tau}, \dots, s_{t-\tau}, s_t]$, where $m, \tau \in \mathbb{Z}^+$. By using these vectors, it is possible to reconstruct an m -dimensional phase space that reflects the dynamics of the system partially observed through the time series ([20,21]).

We use a sliding window of size m to extract all the delay vectors from the time series. To obtain neighbors closer to S_N (the most recent observation), an ϵ parameter is used and the distance to each delay vector S_t is calculated using a distance function $d : X \times X \rightarrow [0, \infty)$. The nearest neighbors are those S_t whose distance to S_N is, at most, ϵ (2).

$$v_\epsilon(S_N) = \{t \mid \|S_N - S_t\| \leq \epsilon \forall t \in N - 1\} \quad (2)$$

Each of these vectors S_t form the neighborhood $v_\epsilon(S_N)$ with radius ϵ surrounding the vector S_N . Kantz et al. [20] suggests that when $m > 20$, the euclidean distance function provides a better measurement of the neighbor for a given delay vector. For lower dimensions, Kantz recommends using the absolute-value norm.

As mentioned before, to calculate the forecasts, the individual values $s_{t+\Delta n}$ are obtained from each vector S_t that satisfies Equation (2). Equation (3) expresses how to compute the forecast.

$$\hat{s}_{N+\Delta n} = \frac{1}{|v_\epsilon(S_N)|} \sum_{t \in v_\epsilon(S_N)} s_{t+\Delta n} \quad (3)$$

In the case $\Delta n = 1$, Equation (3) produces a forecast for the next instant of time. In this paper, forecasts are required for forecasting horizon $n = 2$.

3.3. Differential Evolution

The NN forecasting method requires three parameters: m , τ , and ϵ (defined in the previous section).

To determine the optimal values of these parameters, we propose using Differential Evolution. Differential Evolution (DE) is a meta-heuristic capable of optimizing functions of arbitrary complexity [22]. DE is a population-based optimizer. Population-based optimizers evaluate a set of potential solutions against each other using a fitness function at each iteration of the evolutionary process. After several iterations, DE selects one of the individuals as the best guess to the problem solution. The DE implementation used for this work is known as DE/rand/1/bin.

Since two of the NN algorithm parameters are integers (m and τ) and ϵ is a real number, we add constraints of type and limits to DE. The DE population operations consider these variables continuous, but when used to obtain an individual's fitness, m , and τ , it considers them as integers [23]. The individuals represent the parameters used by the NN algorithm taking the form $X^{(g,i)} = [m, \tau, \epsilon]$.

To evaluate an individuals' fitness, it is necessary to compare the forecasts it produces against the test set, a subset of the time series that the training algorithm never gets to see. Equation (4) shows the fitness function used to evaluate each DE individual

$$\text{fitness}_{X^{(g,i)}} = \text{error}(V, F_{X^{(g,i)}}) \quad (4)$$

where $X^{(g,i)}$ is the i -th individual of generation g . The $\text{error}(\cdot)$ function can be any function that measures the accuracy of the NN process, using the genetic information of individual $X^{(g,i)}$ as parameters against the validation set. We use the Mean Absolute Error (MAE) as the error function in the solution of this problem.

Formally, the problem to solve is given by Equation (5).

$$\begin{aligned} \min \text{fitness}(X^{(g,i)}) & \\ \text{s.t.} & \\ L_j \leq x_j^{(g,i)} \leq U_j & \\ 1 \leq j \leq D & \end{aligned} \quad (5)$$

where D is the dimension of the search space, L is the lower limit, and U is the upper limit of the j values that compose the individuals.

To satisfy the constraints of the NN parameters, we need to consider the mutation function of DE. In DE, the individuals are crossed over with a randomly generated mutant individual. DE uses (6) in the mutation process

$$V^{(g,i)} = X^{(g,r_0)} + (X^{(g,r_1)} - X^{(g,r_2)})F \quad (6)$$

$$\forall i \in [1, N_{pop}]$$

where r_0 , r_1 and r_2 are the indexes of randomly selected individuals, F is a positive real number that controls the speed in which the population evolves, and N_{pop} is the population size.

Since only the contributions of a mutant vector can violate the limits of the parameters, a reset method called bounce-back [24] is needed. This method replaces the individual that has exceeded one or more of its limits with a valid individual that satisfies every constraint. This strategy does not alter the DE process, since it selects a value that is between the base value of the parameter and the limit that is being violated.

When a mutant individual violates the parameter limits, Equations (7) and (8) are used to replace this parameter to satisfy both lower and upper limits

$$v_j^{(g,i)} = x_j^{(g,b)} + U(0,1)(L_j + x_j^{(g,b)}) \quad (7)$$

$$v_j^{(g,i)} = x_j^{(g,b)} + U(0,1)(U_j + x_j^{(g,b)}) \quad (8)$$

where $x^{(g,b)}$ is the base vector of the population (that is, $x^{(g,b)}$ is a vector that points to the origin of the search space).

By using DE, it is possible to simultaneously optimize the parameters used by NN for the specific fitness function required for the task [21].

It is impossible to assume that a set of parameters obtained for a time series is valid for a similar time series, since they can describe different dynamical systems. Depending on the time series (this is how recurrent the sequences present in it are), it is safe to assume that the parameters obtained with this method will be valid for a long number of forecasts. However, if the time series changes its regime considerably or new patterns start to emerge, it would be necessary to repeat the optimization process.

The time series also has to be stationary (this is, it should not have any trend) to use NNLF successfully. If the time series contains any trend, the historical data will not be helpful since it would reflect a different regime, not present in recent times.

3.4. Data Analysis

This section provides a brief analysis of the time series, which focuses on the seasonality and stationarity of the time series that composes the MIPS. This analysis aims to determine if the time series in their current state would require further preprocessing. As mentioned before, NNLF (similar to many other forecasting methods) requires that the time series is stationary. The time series should also contain repeated sequences since NNLF will look back on the time series's history, looking for similarities with the most recent observations.

The MIPS contains seven regional transmission control areas, which are electrically interconnected to improve the system's operation reliability: Central (CEL), Northeast (NES), Northwest (NOR), North (NTE), Occidental (OCC), Oriental (ORI), and Peninsular (PEN).

The data comprises 13 months of data measures in a one-minute resolution. The data were sub-sampled to a 15 min resolution, since the forecasting task requires this resolution.

Seasonality and stationarity measures characterize a time series. The Auto Correlation Function (ACF) and its corresponding plot measure the seasonality [25], and the Augmented Dickey–Fuller

(ADF) test [26] determines whether or not the time series is stationary. We also analyzed the phase space reconstruction of the time series and proposed modification to construct delay vectors in Section 3.4.3.

3.4.1. Auto-Correlation Analysis of the Datasets

We obtained the ACF plots of all of the regions that compose the MIPS. Figure 2 shows the ACF plot of four randomly selected regions. The figures display how correlated is a time series against a time-lagged (displaced) copy of itself. The autocorrelation coefficient at lag h (r_h) is calculated using Equation (9)

$$r_h = \frac{\frac{1}{N} \sum_{t=1}^{N-h} (s_t - \bar{s})(s_{t+h} - \bar{s})}{\frac{1}{N} \sum_{t=1}^N (s_t - \bar{s})^2} \quad (9)$$

where N is the length of the time series, t is a time instant, s_t is the value of the time series at time instant t , and \bar{s} is the mean of the time series.

The plot also contains a confidence interval (the blue “shadow” around 0.0 auto-correlation coefficient); when r_h exceeds the confidence interval, it means that there is certainty that the auto-correlation coefficient value is not a product of the randomness present in the data. It is common practice to set the threshold to 5%. Equation (10) computes this threshold.

$$\pm z_{1-\alpha/2} \sqrt{\frac{1}{N} \left(1 + 2 \sum_{t=1}^h s_t^2\right)} \quad (10)$$

A strong daily seasonal component is present in all of the plots, since the auto-correlation coefficient increases around the 96th lag (the data sampling period is 15 min – $4 \times 24 = 96$).

Alternating ACF plots are very common in stationary time series (see Figure 2a). If the magnitude of the autocorrelation coefficients decays exponentially with the lag, in other words, if the energy of the ACF is finite, and its average is also finite, the time series is stationary [25]. When the autocorrelation coefficient at a particular lag is zero, this indicates that that lag is unimportant in determining the present value of the time series. This comparison with zero is made by comparing the sample autocorrelation coefficient with its standard error. We do a hypothesis test, where the null hypothesis is “the autocorrelation coefficient at lag k is zero”. When the auto-correlation values exceed the 5% confidence interval threshold, the null hypothesis can be rejected. Positive (negative) autocorrelation coefficients indicate that measurements at that lag affect positively (negatively) the time series’ current value.

Figure 2b,c shows a high auto-correlation coefficient for all of the lags observed, with particular bumps around the 96th lag. These time series are non-stationary since their auto-correlation coefficient decays slowly, indicating that the time series’ value increases as the lag increases. An additional test must confirm or discard the non-stationarity of the time series. Section 3.4.2 shows that there is no sufficient evidence to consider the time series as non-stationary.

Examining the plot of the whole time series (Figure 3), it is possible to appreciate an evident annual change in demand. However, because of the amount of data available, it was not possible to make the annual ACF plots.

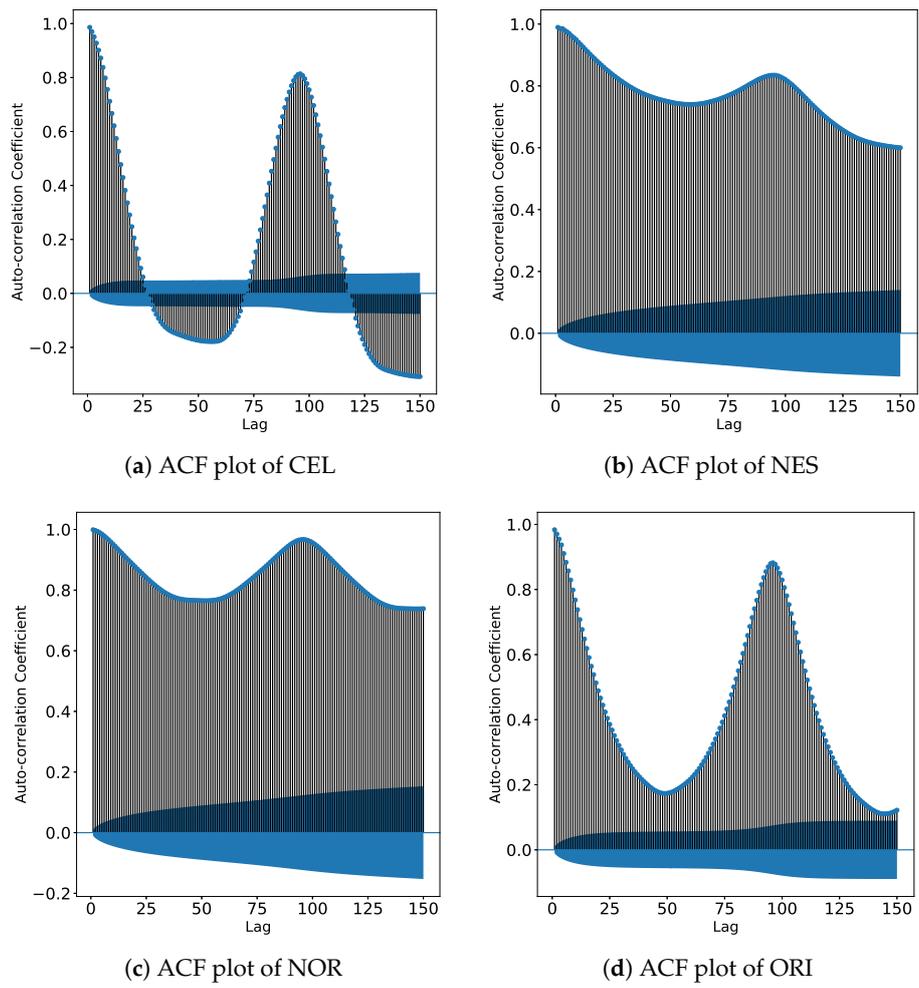


Figure 2. ACF Plots of Four Regions of the MIPS.

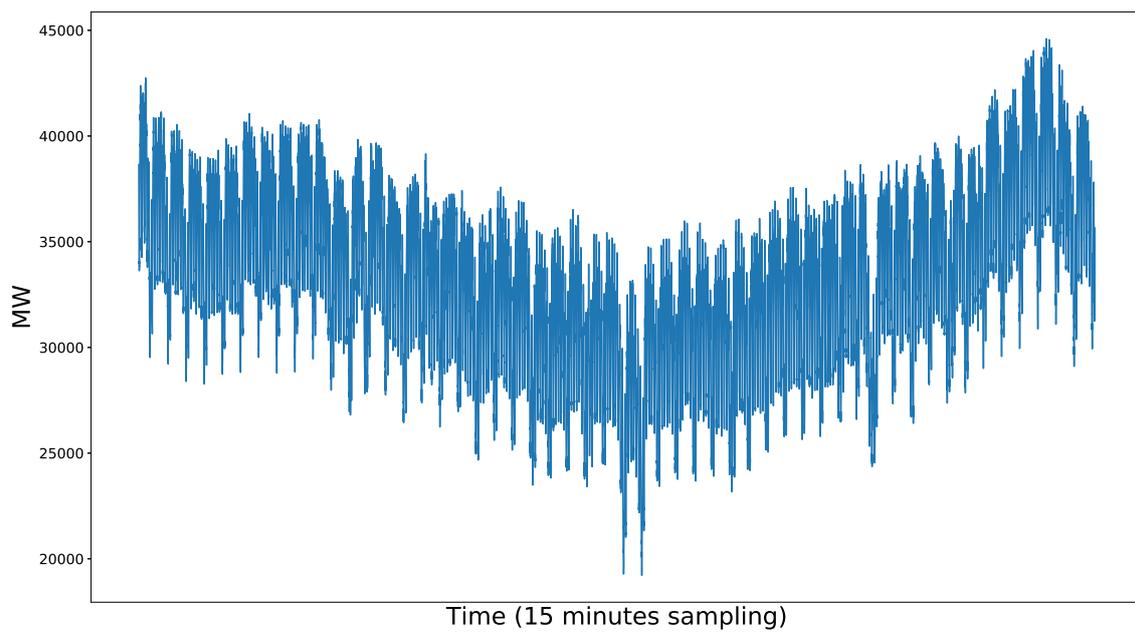


Figure 3. MIPS Time Series.

3.4.2. Augmented Dickey–Fuller Test of the Datasets

We used the Augmented Dickey–Fuller (ADF) test to verify whether the time series was stationary or not. In statistics, the Dickey–Fuller test tests the null hypothesis that a unit root is present in an auto-regressive (AR) model. A simple AR(1) model is shown in Equation (11)

$$s_t = \rho s_{t-1} + e_t \quad (11)$$

where ρ is the AR coefficient, and e_t is the error term. A unit root is an indication that a time series has a stochastic trend; a unit root is present if $\rho = 1$. The model would not be stationary in this case. Equation (11) can be rewritten as Equation (12)

$$\Delta s_t = (\rho - 1)s_{t-1} + e_t = \delta s_{t-1} + e_t \quad (12)$$

where Δ is the difference operator. We cannot use the standard t-distribution to provide critical values since the test is done over the residual term rather than raw data. Therefore, the values of this statistic t are obtained from a specific distribution known as Dickey–Fuller.

The ADF test adds lagged differences and is shown in Equation (13)

$$\Delta s_t = \delta s_{t-1} + \sum_{k=1}^n u_k \delta s_{t-k} + e_t \quad (13)$$

where n is the number of lag coefficients, and u_k is the lagged coefficient.

Table 1 shows the ADF statistic and p -value of each time series.

Table 1. ADF Test Results of the MIPS Regions.

Time Series	ADF Statistic	p -Value
CEL	−24.430219	0.000000
NES	−7.562009	0.000000
NOR	−5.516391	0.000002
NTE	−6.467922	0.000000
OCC	−14.851454	0.000000
ORI	−14.828665	0.000000
PEN	−11.073954	0.000000
MIPS	−11.762601	0.000000

In the previous section, the ACF indicates the possible non-stationarity of the time series. In order to determine whether or not the time series is indeed non-stationary, we will perform the Augmented Dickey–Fuller (ADF) test [27], a standard statistical test used to determine whether a given time series is stationary or not. ADF is a statistical significance test where there is a hypothesis testing involved, with null and alternate hypotheses, and as a result, we compute a test statistic and p -values. As in all statistical tests, a statistic is computed and compared against a critical value. The probability distribution that the statistic follows determines the test's critical value. For a given significance value (typically 1, 5, or 10%), we determine two regions: a rejection and a non-rejection (also called critical) region. The probability of the critical region is α . The decision rule is to reject the null hypothesis H_0 if the observed value is in the critical region, and to fail to reject the hypothesis otherwise.

The critical values for the ADF statistic are: 1%: −3.430, 5%: −2.862, and 10%: −2.567. These results can be deceiving, since we know that the demand for electrical energy is always increasing. The explanation for these results is that the dataset is relatively short (it is composed of one year of observations), and the global trend of the time series is not perceptible. Nonetheless, for the experiments, the time series will be treated as stationary. Figure 4 shows the same reconstructed

attractor, but a different color highlights each day of the week. Using this distinction, the differences between days became more apparent as the same days tend to share their orbits.

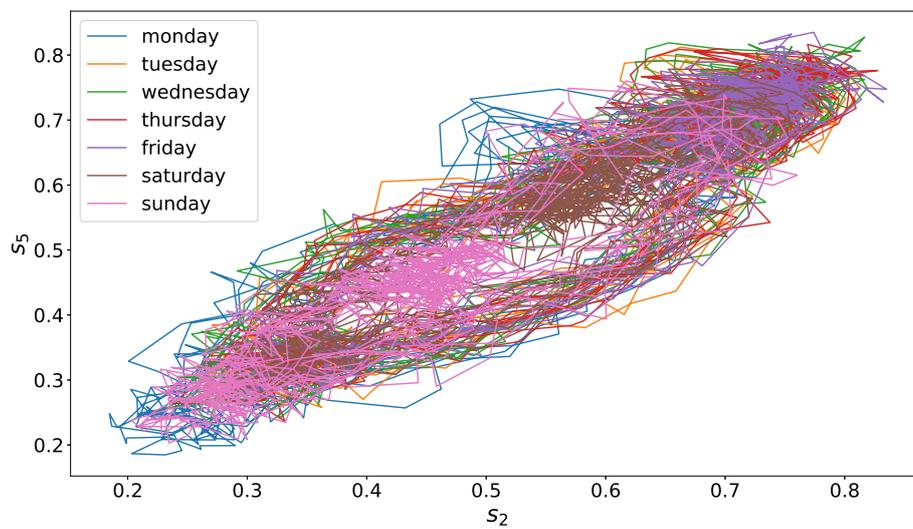


Figure 4. Reconstructed Attractor (highlighted days).

3.4.3. Phase Space Reconstruction Analysis

As mentioned in Section 3.1, this proposal is based on phase space reconstruction. Figure 5 shows the reconstructed attractor of the time series. We found that by separating the data by day of the week, the unique features that distinguish each day stand out.

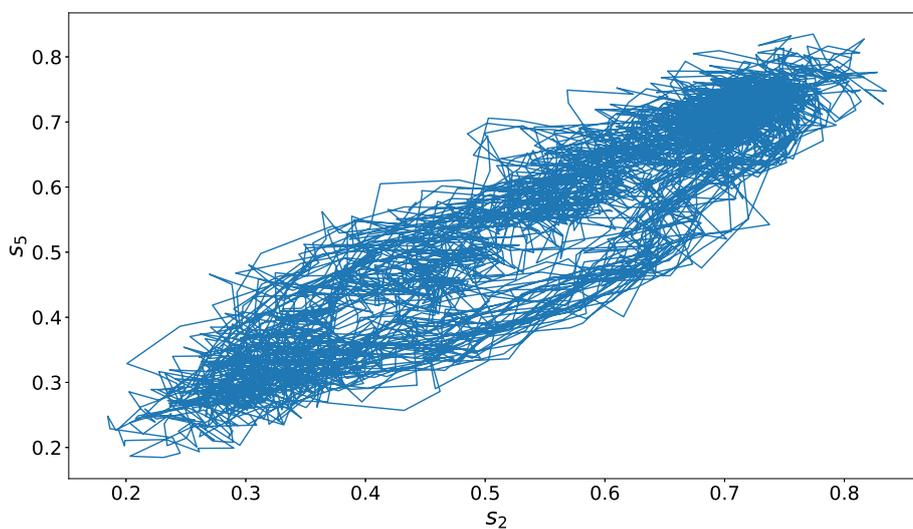


Figure 5. Reconstructed Attractor.

In this article, we took the day of the week and the sampling time as additional dimensions of the delay vectors. That is, we append the day of the week and the time to the samples obtained from the time series; those exogenous variables are taken into consideration by the forecasting models. With this change, we expect to allow the forecasters to consider the influence of external factors [12].

Each delay vector S_t includes day and time as control dimensions. Another change was the normalization of the values of the time series. The values were normalized, so none of the dimensions dominate the rest of the dimensions. The resulting vector is shown in Equation (14)

$$S_t = [\text{DoW}(d_t), \text{MoD}(d_t), \tilde{s}_{t-(m-1)\tau}, \dots, \tilde{s}_{t-\tau}, \tilde{s}_t] \quad (14)$$

where \bar{s}_t is the normalized value of the time series at time t , d_t represents the date and hour of the last element of the delay vector, $\text{DoW}(\cdot)$ returns the normalized day of the week (as a real number in the range $[0, 1]$; the days are represented as a number from 0 to 6 and then divided by 6) starting on Monday that corresponds to the date d_t , and $\text{MoD}(\cdot)$ returns the normalized minute of the day (represented as a real number in the range $[0, 1]$ (instead of representing the hour as HH:MM, it is converted to minutes from 0 to 1339 and then divided by 1339)) that corresponds to d_t 's sampling time. The combination of this usage of delay vectors with exogenous variables and the optimization of NN parameters with DE is what we called Nearest Neighbors for Load Forecasting (NNLF).

To provide an illustrative example of the phase space reconstruction and one such delay vector, let us assume that $m = 13$, and $\tau = 1$; a delay vector taken on a Tuesday at 00:00 would be: (0.333, 0.0, 0.584, 0.368, 0.571, 0.568, 0.326, 0.678, 0.735, 0.406, 0.725, 0.718, 0.475, 0.611, 0.672). The first two numbers are $\text{DoW}(\text{Tuesday}) = 2/6 = 0.333$, $\text{MoD}(00:00) = 0.0$; the last 13 numbers are measurements from the time series.

With the ideas discussed in this section, it is possible to construct a forecaster based on Nearest Neighbors, optimizing its parameters with Differential Evolution (the authors' solution proposal). The data available show a daily correlation and can be considered stationary. Nonetheless, more data are required to ensure that the stationarity test holds. The problems that a trend poses when modeling a time series can be solved by introducing some transformation, such as a first-order difference. When a time series is transformed, later on, the forecasts are inversely transformed into the original units. The reconstructed attractor shows that partitioning the data into days of the week allows for identifying different orbits typical for the same days of the week.

4. Experiments and Results

This section contains the results obtained by NNLF and a set of two different forecasters. It defines the methodology followed by the experiments, shows and discusses the forecast plots produced by each of the methods included in the comparison, and, finally, it includes an analysis of the errors obtained by each forecaster.

4.1. Forecasters Included in the Comparison

Considering the experimental techniques observed in the literature review, we selected two different machine learning techniques: Multi-Layer Perceptron (MLP) and Support Vector Regressor (SVR). To complete the comparison, three of the most used techniques were included: Seasonal ARIMA (sARIMA), Kalman Filter, and the persistence method.

4.1.1. Multi-Layer Perceptron

Multi-Layer Perceptrons are computing systems vaguely inspired by the biological neural networks that constitute animal brains [28]. An MLP is composed of a set of basic computing units known as neurons, organized by layers. The data flow from a layer known as the input layer (which has the same number of neurons as the input vector) through one or more hidden layers of variable size to reach a final layer known as the output layer, whose size depends on its number outputs required [29].

4.1.2. Support Vector Regressor

A Support Vector Regressor is a forecasting model that attempts to identify the minimum margins of separation between the data vectors to fit most of them inside the margins while limiting margin violations. γ is the hyperparameter that controls the width of that margins [30].

4.1.3. Seasonal ARIMA

Auto-regressive models assume that successive observations in a time series show serial dependence between the signal and noise. ARIMA is a statistical forecasting technique, often used

in time series that contain specific properties, such as seasonality and stationarity [31]. In this work, we used the seasonal variant, which also considers the frequency in which specific patterns repeat in a time series.

4.1.4. Kalman Filter

The Kalman filter (KF) is an algorithm that can filter out the noise present in the observations given a set of noisy observations. KF assumes that the noise and the observed process come from a gaussian process [32].

4.1.5. Persistence Method

The persistence method (PM) assumes that the observed process will continue to have the same behavior until it does not. It is also known as the naive method since it repeats the last known observation.

4.2. Experiment Description

The forecasting task is to forecast the MIPS net demand using the time series of the control regions that compose it. We need to forecast 30-min ahead for a whole day (i.e., 92 forecasts). The validation set covers the time interval from 00:00 to 23:45 h of 24 June 2018, sampled every 15 min.

Below is a description of the parameters used with each forecaster.

NNLF: DE was executed for 30 generations, each generation with a population size of 30.

This optimization process was repeated for 30 independently seeded executions. The individuals could take values from $m \in [1, 100]$, $\tau \in [1, 100]$, $\epsilon \in [0, \max(TS) - \min(TS)]$ with a recombination probability of 0.75. To evaluate the individuals, DE uses MAE as the error function.

MLP: Even though there is a theorem stating that a single layer is enough to approximate any function at any level of accuracy [33], training deep ANNs takes less computational effort than training shallow ones. This fact is widely discussed in the Deep Learning book, by Goodfellow et al. [34]. The topology of the designed MLP consists of m input neurons (where m is the same vector length used by NNLF for each of the regions), two hidden layers of 100 neurons each and a single output neuron. This topology was selected empirically by probing neural networks with one to four hidden layers with different layer sizes. The activation function used was the tanh function, selected empirically, by testing different activation functions.

SVR: The kernel used was RBF. We used grid search to optimize the SVR parameters (C and γ) for each time series; the search intervals that establish the grid are $[1, 1000]$ and $[0.01, 10]$, respectively.

sARIMA: The sARIMA parameters were obtained using the `auto.arima` function of the forecast library of R [35].

KF: The initial transition matrices were set empirically by trial and error for each of the time series. The initial transition covariance for each Kalman filter instance was set to $\begin{pmatrix} \sigma_X & 0 \\ 0 & \sigma_X \end{pmatrix}$, where σ_X is the variance of the time series. The filter was trained using the Expectation–Maximization algorithm.

PM: In this method, the forecasts are the last known observations; that is, $\hat{y}_{t+2} = y_t$, since, for this problem, forecasting two time steps in advance is desired. No other parameters are involved.

All of the forecasters used the same datasets, obtained with DE, with the exceptions of sARIMA, KF, and PM, since those methods are not vector-based.

The comparative analysis relative to the forecasting task uses both the Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). MAE is described in Equation (15), while MAPE is described in Equation (16).

$$\text{MAE}(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^n |y_i - \hat{y}_i| \quad (15)$$

$$\text{MAPE}(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \quad (16)$$

In both equations, y is the set of actual values, \hat{y} is the set of forecasts, and n is the number of samples.

4.3. Results

Table 2 shows the data et parameters used by all of the forecasters, except the statistical methods (MLP and SVR did not use the ϵ parameter).

Table 2. Forecasters Parameters.

Time Series	$[m, \tau, \epsilon]$
CEL	[6, 4, 0.34100608]
NES	[8, 1, 0.14140621]
NOR	[2, 46, 0.05696183]
NTE	[2, 4, 0.05038196]
OCC	[13, 31, 0.34437311]
ORI	[2, 6, 0.06026154]
PEN	[1, 22, 0.02691927]

Table 3 shows the forecasting errors for the MIPS net demand of the forecasters (the MIPS forecast is the sum of the forecasts of each region for a given time instant.) The first number in each cell is the MAE score, while the parenthesis number is their MAPE score. The table also included a column for the SICAPI, which is the forecaster used by CENACE. For this particular forecaster, the MAPE error per region was not available. CENACE uses a forecasting system designed to forecast the MIPS only. This forecaster is known to employ an analog multiple regression model. Additional information about this forecaster is undisclosed because of government regulatory conditions. Compared to the forecasting error produced by SICAPI, NNLF reduces the forecasting error by 4.6%, meaning that, for the day, SICAPI overshoot the total energy consumption by 340.4MW. NNLF obtains the best forecasting error for the MIPS time series, out-performing all of the forecasters by an ample margin. Although NNLF does not obtain most of the individual best scores by region, it produces a better forecast for the total load demand than those obtained by the other methods.

Table 3. MAE and MAPE scores for MIPS Time Series.

Series	NNLF	MLP	SVR	sARIMA	KF	PM	SICAPI
CEL	126.5 (2.138)	127.4 (2.139)	119.8 (2.008)	318.4 (5.415)	156.0 (2.605)	181.0 (3.029)	-
NES	113.4 (1.698)	116.2 (1.738)	122.3 (1.822)	349.5 (5.284)	134.0 (1.999)	157.3 (2.341)	-
NOR	23.4 (0.737)	20.1 (0.618)	26.9 (0.825)	51.8 (1.575)	57.4 (1.775)	78.5 (2.430)	-
NTE	51.2 (1.411)	21.8 (0.601)	20.4 (0.562)	462.8 (12.725)	29.0 (0.793)	39.9 (1.091)	-
OCC	97.3 (1.468)	92.4 (1.39)	79.8 (1.207)	630.5 (9.547)	114.9 (1.721)	122.3 (1.824)	-
ORI	97.5 (1.732)	99.6 (1.771)	102.3 (1.814)	343.4 (6.330)	123.3 (2.164)	147.3 (2.577)	-
PEN	13.3 (0.867)	11.7 (0.758)	12.6 (0.829)	137.5 (9.289)	23.3 (1.537)	33.0 (2.173)	-
MIPS	213.3 (0.644)	232.4 (0.734)	229.0 (0.687)	2124.7 (6.471)	394.3 (1.174)	499.8 (1.488)	223.6 (-)

It is important to note that each control region's contribution to the MIPS is not equal; i.e., some regions consume more energy than the others. Table 3 shows that NNLF wins in two of the four regions where the three best forecasters obtained their worst scores. Those regions are the most difficult to forecast and the ones that consume more energy. Therefore, those regions contribute more to the MIPS than the rest of the regions. This situation enabled NNLF to obtain the best score without being the best for all regions.

We have focused on improving the load forecasting associated with the Mexican National Electric System's total electric demand. The net load is used by CENACE to perform the generation/load balance to maintain the system's frequency within normal operating limits in the real-time market. CENACE oversees the Operational Control of the Mexican National Electric System and the Wholesale

Electricity Market (MEM by its acronym in Spanish). Within the MEM, the short-term market stands out: the day-ahead market and real-time market. The real-time market requires, among other information, the demand forecast of the National Electric System 15 min in advance, before the start of the next Dispatch Interval, in addition to the 30, 45, and 60-min forecasts, to execute the security-constrained economic dispatch for multiple intervals. However, the first dispatch interval is the only one that establishes physical delivery commitments and local marginal prices [36]. We will work on alternatives to improve the load forecasting associated with each control area in the near future.

As mentioned in Section 2, the forecasting tasks vary vastly from paper to paper. The most similar forecasting task we found is described in [7], where they use 5-min to 40-min samplings. Although they do not use 15-min sampling, their best results for 5-min and 20-min samplings are 0.49% and 1.31%, respectively. NNLF obtained 0.644% MAPE, which is closer in value to the score obtained by their method at 5-min sampling, but three times farther in sampling.

The following figures show the forecasts made by each forecaster compared with the validation set.

Figure 6 shows the MIPS forecasts made by NNLF. NNLF shows a good approximation of the time series from 00:15 to 03:15 h, but misses a significant drop that lasted 30 min (all errors in that interval fall within the required margin of 500 MW). That figure clearly shows that from 09:30 to 18:15 h, the forecast follows the increasing trend, but is mostly below the actual value. This behavior implies that NNLF could capture the global trend in the time series, but smooths the high-frequency variations, which may not be desirable. From 19:45 to 21:15 h, the forecast lags behind the actual curve, especially at 20:30 and 20:45 h. This tells us that most days observed by NNLF did not show this kind of behavior; the curve that NNLF produces is less abrupt than the one produced by SICAPI.

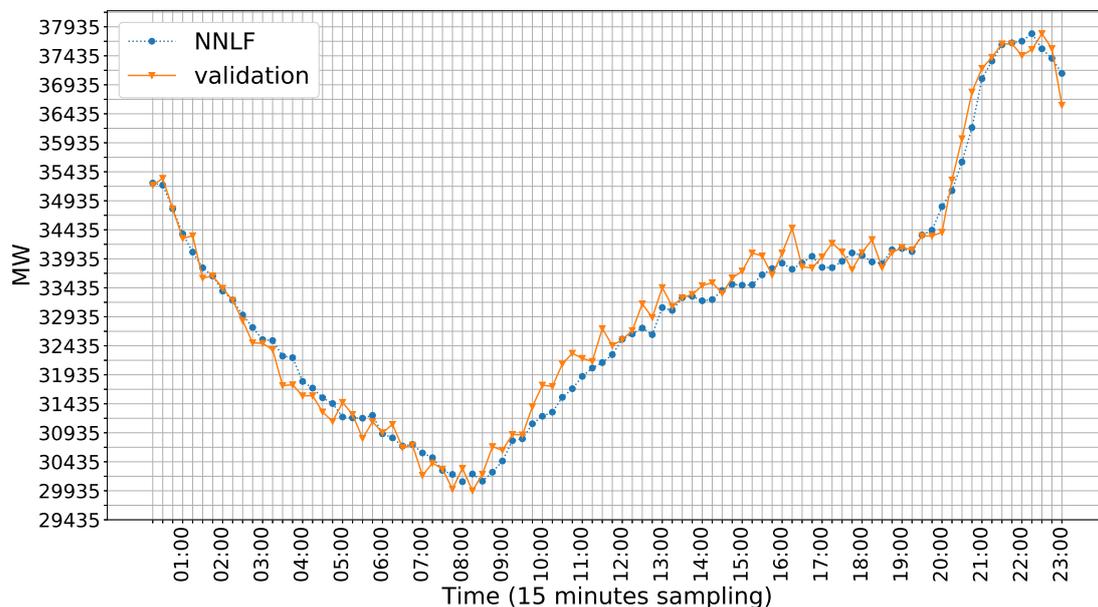


Figure 6. NNLF Forecasts vs. Validation.

Figure 7 shows the forecasts made by MLP. This forecaster obtained a reasonable approximation of the time series for the range between 00:15 and 04:45 h. Nonetheless, it misses the trend of the high-frequency changes from 05:00 to 07:00 h. From 09:30 to 19:00 h, MLP follows the increasing trend more tightly than NNLF, but, as with NNLF, it misses the high-frequency changes present in the validation set. However, what penalizes the score obtained by MLP the most are the forecasts made from 19:30 to 21:45 h. The trend captured by MLP does not correspond to the actual values of the time series. This fact gives weight to the assumption that the curve of this day is atypical and under-represented in the time series' history.

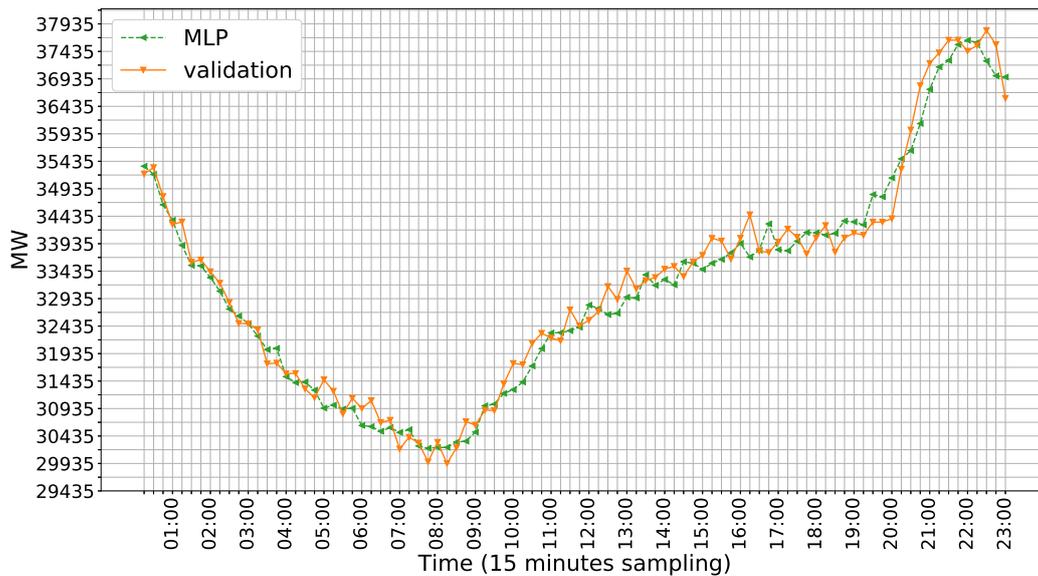


Figure 7. MLP Forecasts vs. Validation.

Figure 8 shows the forecasts made by SVR. This method shows a very similar forecast to MLP but with better approximation, particularly at the start of the day. The forecasts made at the ramp's start are also more accurate than the ones made by MLP, but not as good as the ones done by NNLF. The worst forecasts made by these methods are from the end of the day.

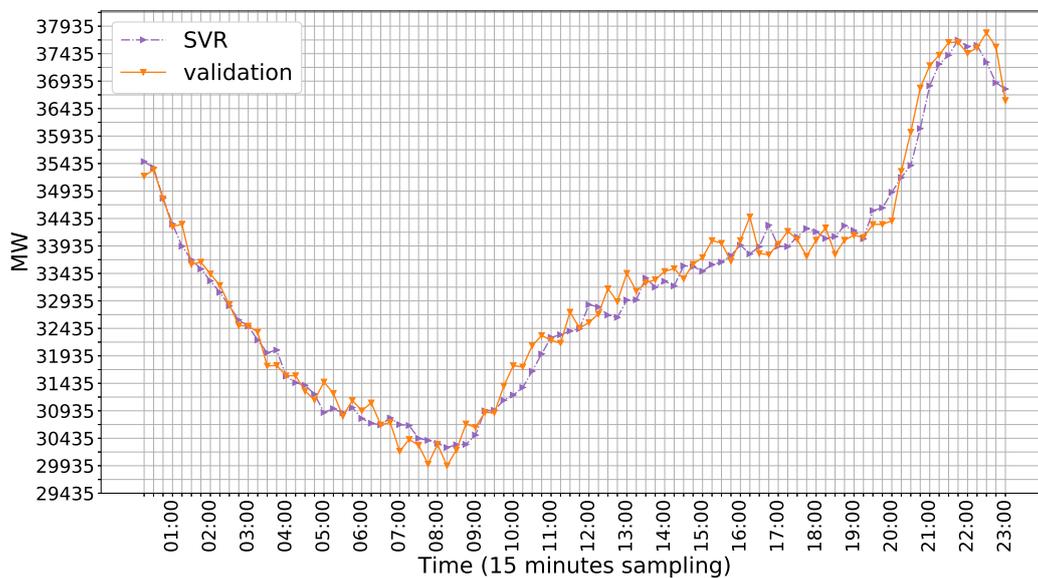


Figure 8. SVR Forecasts vs. Validation.

Figure 9 shows the forecasts made by sARIMA. It is possible to see that sARIMA ignores the data and produced a version of a typical day. This behavior happens because of the low order of the resulting sARIMA processes generated by `auto.arima` and because of the limitations of its formulation for high-resolution time series.

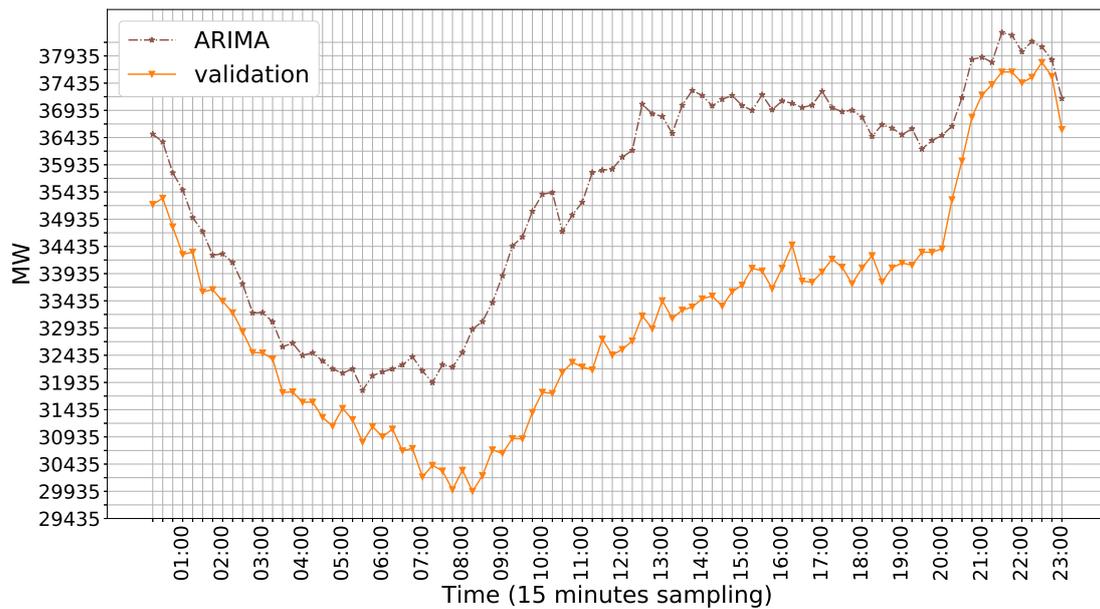


Figure 9. sARIMA Forecasts vs. Validation.

Figure 10 shows the forecasts made by KF. This method was not able to capture the behavior of the time series correctly, which can be attributed to the initial observations being similar to “upward” rather than “downward” observations. This is a problem we faced with NNLF when using delay vectors without exogenous variables. The exogenous variables help much to identify the trend of the time series correctly. Once the trend is clear, KF is unable to recover.

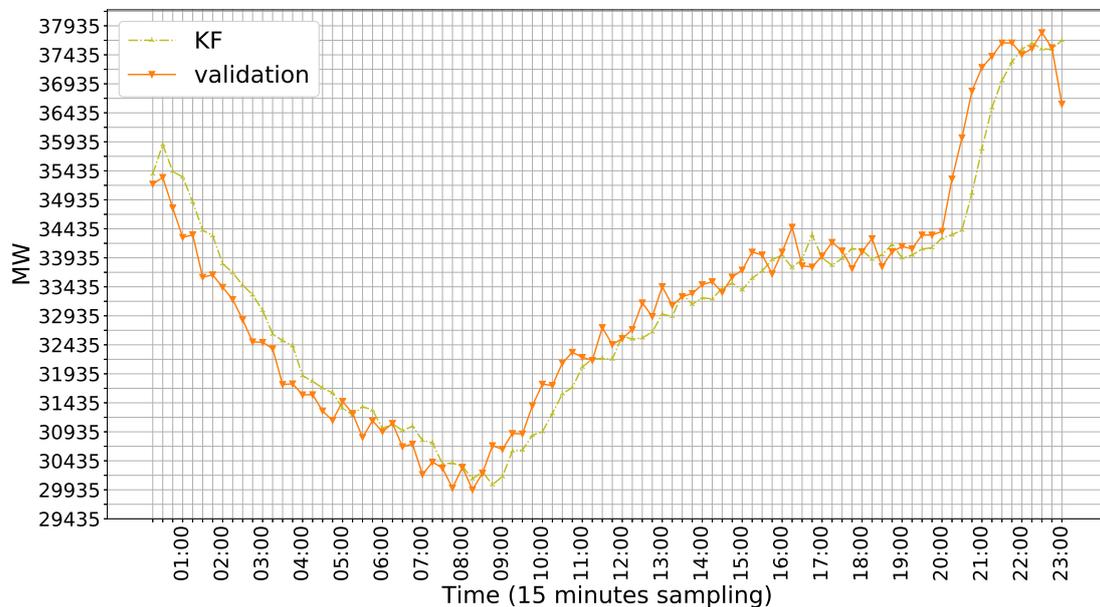


Figure 10. KF Forecasts vs. Validation.

Figure 11 shows the forecasts made by PM. This technique is not suited for this type of problem, since the time series presents significant high-frequency changes. Its performance is limited because when the ramp of demand happens, this method will be two time steps behind it, rendering its forecasts useless.

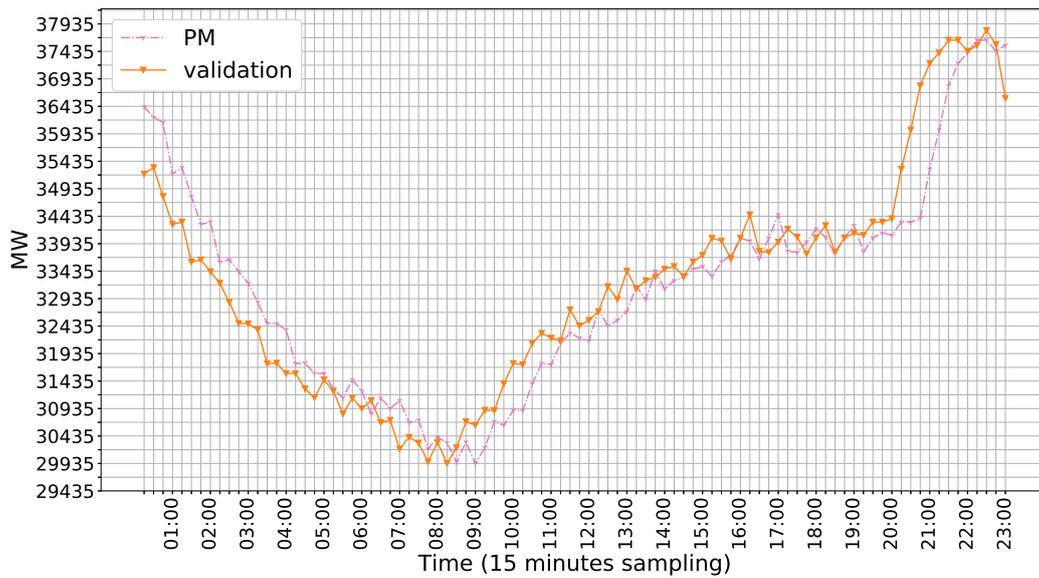


Figure 11. PM Forecasts vs. Validation.

The forecast plots consistently show that none of the forecasting methods can follow the increase in demand towards the end of the day. It is possible to consider that the ramp of demand was atypical, or that the data do not contain enough samples of a day like this one. To corroborate these hypotheses, a simple analysis was performed: the histogram of the errors should approximate a normal distribution if enough data samples are present in the dataset.

Figure 12a shows the histogram of the forecasting errors made by NNLF and a normal fit from the data. The histogram barely resembles a normal distribution curve. Nonetheless, there is a decline in higher magnitude errors in the tails.

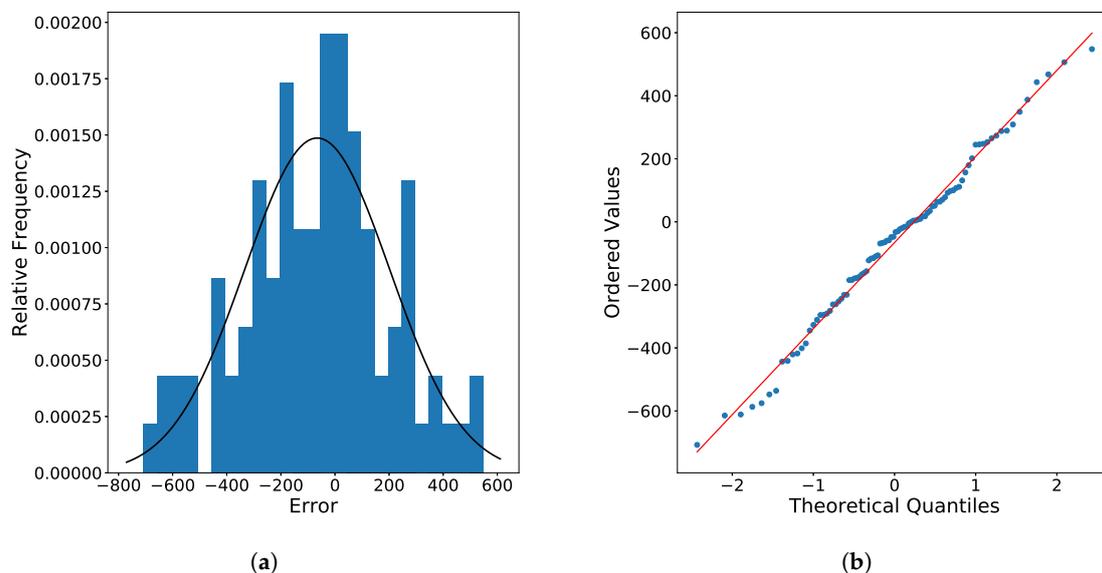


Figure 12. Analysis of NNLF Errors. (a) Histogram of NNLF Forecast Errors ($\mu = -65.64, \sigma = 268.33$). (b) Probability Plot of NNLF Errors.

Figure 12b shows that the errors do follow a normal distribution, with some outliers at the quantiles' extremes.

Figure 13a shows the histogram of the errors obtained by MLP. In this case, the histogram does not follow the normal curve, either. As with NNLF, the errors are not centered at 0.

Figure 13b shows a significant outlier at the right extreme of the probability plot and some dispersion at the left end of the normal line. This case can also be considered as normally distributed.

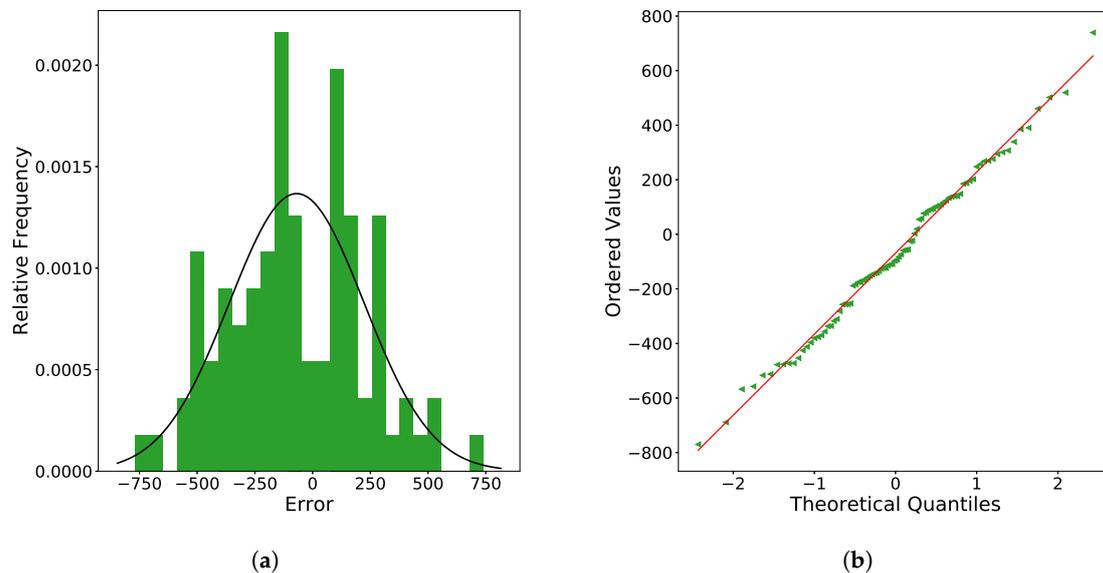


Figure 13. Analysis of MLP Errors. (a) Histogram of MLP Forecast Errors ($\mu = -68.92$, $\sigma = 291.76$). (b) Probability Plot of MLP Errors.

Figure 14a shows the histogram of the errors made by SVR. Again, the errors do not follow the normal curve and are not centered at 0. However, the range of SVR errors is similar to those of NNLF.

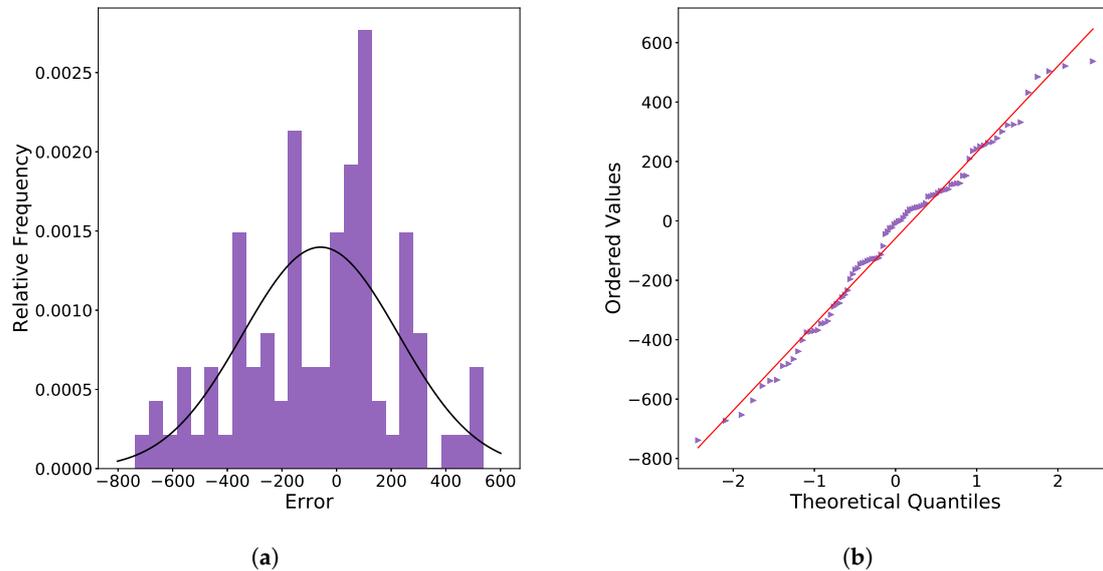


Figure 14. Analysis of SVR Errors. (a) Histogram of SVR Forecast Errors ($\mu = -262.74$, $\sigma = 416.14$). (b) Probability Plot of SVR Errors.

The probability plot of the SVR errors (Figure 14b) shows many more outliers around the middle and at both ends. The errors obtained by this method cannot be considered normally distributed. Since the SVR parameters were optimized, these results could mean that SVR requires more features to model the time series properly. The authors are well aware that some studies consider other exogenous variables and that they tend to improve the quality of the forecasts obtained. However, it is not feasible, at least at the moment of writing this paper, to include more variables, because of the size, geographical distance, and diversity of activities between regions.

Figure 15a shows the histogram of the errors made by ARIMA. There is absolutely no hint that the errors are normally distributed. Its probability plot (Figure 15b) confirms this.

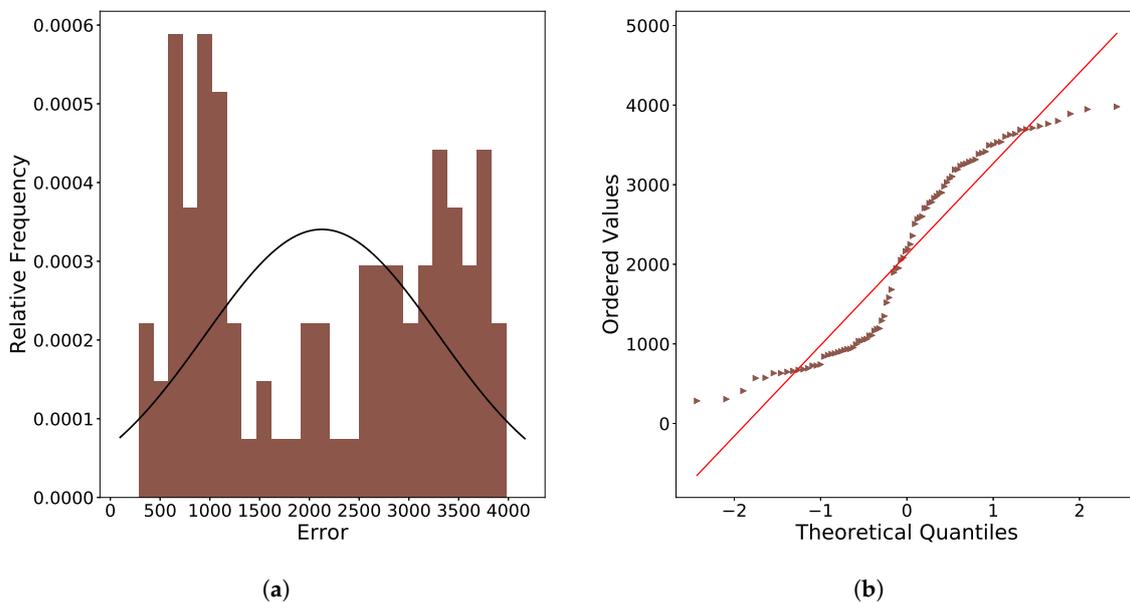


Figure 15. Analysis of ARIMA Errors. (a) Histogram of ARIMA Forecast Errors ($\mu = 2124.75$, $\sigma = 1171.63$). (b) Probability Plot of ARIMA Errors.

Figure 16a shows the histogram of the errors obtained by KF. The errors are mostly centered around 0, but their range is very wide and not suitable for the task.

The probability plot of the errors obtained by KF (Figure 16b) shows large outliers at the left end, but overall they describe a fair approximation of the normal distribution.

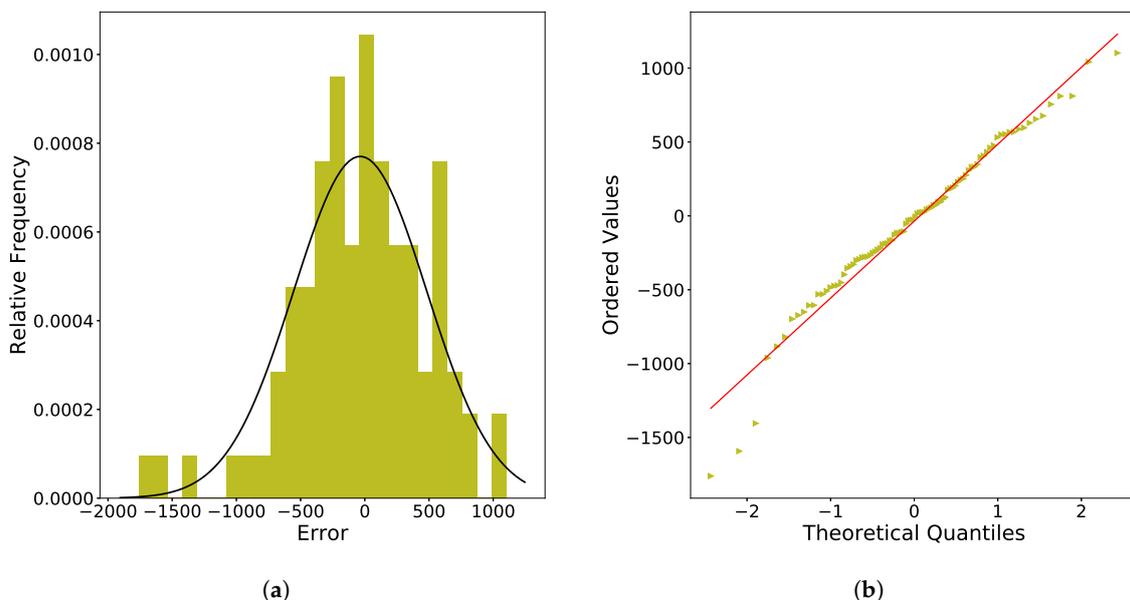


Figure 16. Analysis of KF Errors. (a) Histogram of KF Forecast Errors ($\mu = -35.85$, $\sigma = 517.87$). (b) Probability Plot of KF Errors.

Figure 17a shows the histogram of the errors obtained by PM. Similarly to KF, they resemble a normal distribution with the errors centered around 0, but with a wide range of errors.

Figure 17b also show that the errors follow a normal distribution but with very large outliers.

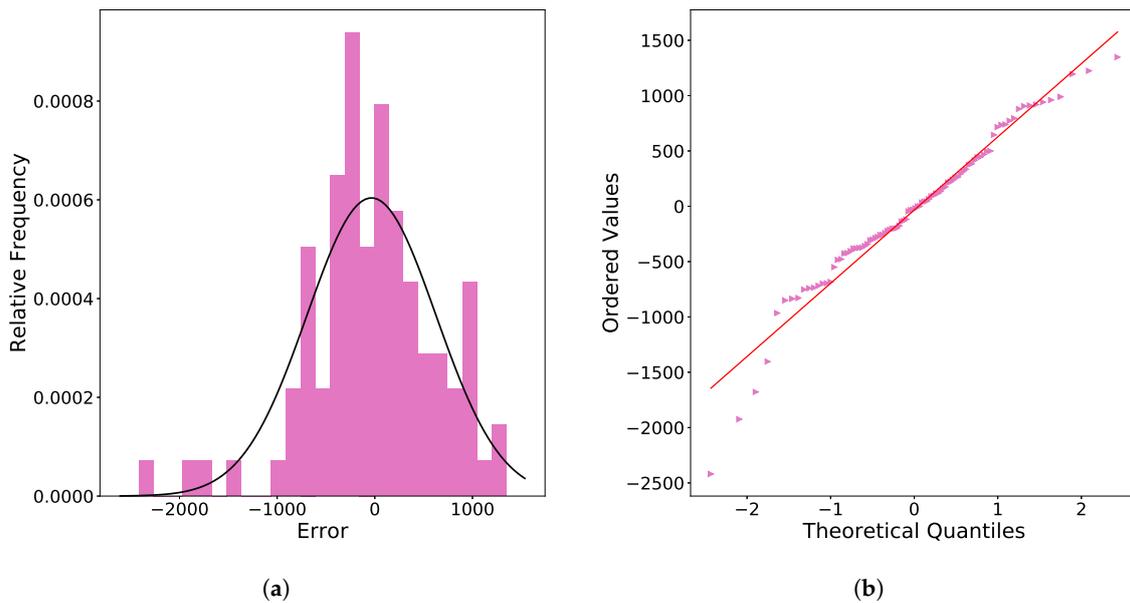


Figure 17. Analysis of PM Errors. (a) Histogram of PM Forecast Errors ($\mu = -34.28$, $\sigma = 660.63$). (b) Probability Plot of PM Errors.

From the error histogram plots, we can conclude that there is bias in the mean of almost every forecaster, which can be a sign that the forecasted day has atypical features or there is still a structure to be modeled to improve the forecasts. The probability plots show that NNLF is the forecaster whose errors best approximate a normal distribution, considering that it is the method where the errors both follow the normal fit and also contain the lowest outliers.

Although the distribution of the errors obtained by NNLF does not closely follow a normal distribution, perhaps the evaluated day could contain atypical features, considering every forecaster's performance. Nonetheless, NNLF produced a better forecast than the rest of the compared forecasters, including the one used by CENACE.

5. Conclusions and Future Work

This section presents the general conclusions obtained by applying NNLF to forecast the electricity demand at the national level in Mexico. We also provide a set of directions in which we plan to work in the near future.

5.1. Conclusions

An alternative and innovative solution to the very-short-term load forecasting problem for electrical power systems has been proposed and implemented in this paper. NLF is innovative since the literature that reflects related work does not contain any works on applying NN, combined with DE and encoded exogenous variables, in their methods. NNLF is a uni-variable model with exogenous variables. The article not only presents a methodology for load forecasting, it also presents a statistical analysis of the input data and the distribution of the forecasting errors produced by NNLF.

A time series forecasting open problem is to determine how much and what parts of the history of the observed variables are useful and essential to be considered by the forecasting models. In this regard, we present a brief introduction to phase space reconstruction. We present the reconstructed attractor for the electric load at the national level. The attractor exhibits a shift in load magnitudes on different days of the week. This fact supports the idea of including time information in the coding of the delay vectors, which constitute the input to the forecasting model.

The tests performed in the proposed methodology used real data from the National Interconnected System in Mexico. Data, testing, and several in-person interviews were provided by and conducted at

the National Center for Energy Control. Since the Mexican interconnected power system comprises seven control regions, we conducted tests by region and at the national level. NNLF and MLP performances were similar in the different regions, outperforming SVR in all cases. NNLF was superior at the national level. The results of the proposed model were also compared against the forecasting system currently used at CENACE; NNLF was far superior compared to what they currently use. We did not include those tests in this article for confidentiality reasons. It is hard to compare NNLF against state-of-the-art, since different papers present work on different datasets (and some with different error measures). However, The MAPE score obtained by NNLF is competitive when compared to other similarly structured forecasting tasks.

5.2. Future Work

Bearing in mind the improvement in the forecast accuracy, we plan to take the following actions, additions, or modifications in models and data.

In the future, we will incorporate external variables to the model, e.g., temperature and humidity. We expect that the additional information provided by these exogenous variables will help improve the forecast accuracy.

We are working on alternatives to improve the load forecasting associated with each control area; we think that a single model for each dataset may not provide the best accuracy. Instead, we will test several models with each region and adopt the most accurate for each one.

In terms of the subset of models to test for each region, we would like to compare the load forecasting produced by NNLF with Deep Learning approaches. We are currently implementing and experimenting with Convolutional and Recurrent Neural Networks (e.g., LSTM). Other approaches and techniques that we are working on and will test in the near future to improve the forecasting accuracy of load forecasting at the national level are time series to image transformations and signal decomposition using wavelets.

Another interesting concept we are testing is data augmentation for time series. This concept has been extensively used for image processing and classification, but barely used in time series, if at all. Data augmentation provides a means of regularization, avoiding the models to overfit, and providing the model with more extensive datasets to train.

Author Contributions: Conceptualization, J.J.F., C.R.F.-E., J.R.C.G. and B.A.M.-A.; methodology, J.J.F., C.R.F.-E. and J.R.C.G.; software, J.R.C.G.; validation, J.R.C.G. and B.A.M.-A.; formal analysis, J.J.F., C.R.F.-E., J.R.C.G. and B.A.M.-A.; investigation, J.J.F., C.R.F.-E. and J.R.C.G.; resources, J.J.F. and C.R.F.-E.; data curation, C.R.F.-E., B.A.M.-A. and J.R.C.G.; writing—original draft preparation, J.R.C.G.; writing—review and editing, J.J.F., B.A.M.-A. and C.R.F.-E.; visualization, J.R.C.G.; supervision, J.J.F., C.R.F.-E.; project administration, J.J.F., C.R.F.-E.; funding acquisition, C.R.F.-E. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Fondo de Sustentabilidad Energética SENER-CONACyT, México, under grants 246949 and 249795.

Acknowledgments: We want to thank the CENACE for the support provided for the making of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ACF	Auto Correlation Function
ADF	Augmented Dickey-Fuller
AI	Artificial Intelligence
ANN	Artificial Neural Networks
AR	Auto-regressive
ARIMA	Auto-regressive Integrated Moving Average
BSK	Bi-square Kernel
CEL	Central Control Region
CENACE	Centro Nacional de Control de Energía (National Center for Energy Control)

DE	Differential Evolution
EDISC	Economic Dispatch with Interval Security Constraint
EMS	Energy Management System
KM	Kalman Filter
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MIPS	Mexican Interconnected Power System
MISCED	Multiple Interval with Security Constraints Economic Dispatch
MLP	Multi-layer Perceptron
NES	Northeast Control Region
NN	Nearest Neighbors
NNLF	Nearest Neighbors for Load Forecasting
NOR	Northwest Control Region
NSS	N-step Simultaneously
NTE	North Control Region
OCC	Occidental Control Region
ORI	Oriental Control Region
OSA	One Step Ahead
PCR	Principal Components Regressor
PEN	Peninsular Control Region
PM	Persistence Method
PSR	Phase Space Reconstruction
RBM	Restricted Boltzmann Machine
RTUA	Real-Time Unit Assignment
SOM	Self-organizing Map
STLF	Short-term Load forecasting
SVM	Support Vector Machine
SVR	Support Vector Regressor

References

1. Feinberg, E.A.; Genethliou, D. Load forecasting. In *Applied Mathematics for Restructured Electric Power Systems*; Springer: Berlin, Germany, 2005; pp. 269–285.
2. Hammad, M.A.; Jereb, B.; Rosi, B.; Dragan, D. Methods and Models for Electric Load Forecasting: A Comprehensive Review. *Logist. Sustain. Transp.* **2020**, *11*, 51–76. [[CrossRef](#)]
3. Jones, L.E. *Renewable Energy Integration: Practical Management of Variability, Uncertainty, and Flexibility in Power Grids*; Academic Press: Cambridge, MA, USA, 2017.
4. Maciejowska, K.; Nitka, W.; Weron, T. Day-ahead vs. Intraday—Forecasting the price spread to maximize economic benefits. *Energies* **2019**, *12*, 631. [[CrossRef](#)]
5. Alpaydin, E. *Introduction to Machine Learning*; MIT Press: Cambridge, MA, USA, 2020.
6. Khatoon, S.; Ibraheem; Singh, A.K.; Priti. Effects of various factors on electric load forecasting: An overview. In Proceedings of the 2014 6th IEEE Power India International Conference (PIICON), Delhi, India, 5–7 December 2014; pp. 1–5.
7. Yang, L.; Yang, H. Analysis of Different Neural Networks and a New Architecture for Short-Term Load Forecasting. *Energies* **2019**, *12*, 1433. [[CrossRef](#)]
8. Jain, A.; Satish, B. Clustering based short term load forecasting using support vector machines. In Proceedings of the 2009 IEEE Bucharest PowerTech, Bucharest, Romania, 28 June–2 July 2009; pp. 1–8.
9. Dudek, G. Pattern-based local linear regression models for short-term load forecasting. *Electr. Power Syst. Res.* **2016**, *130*, 139–147. [[CrossRef](#)]
10. Khwaja, A.; Zhang, X.; Anpalagan, A.; Venkatesh, B. Boosted neural networks for improved short-term electric load forecasting. *Electr. Power Syst. Res.* **2017**, *143*, 431–437. [[CrossRef](#)]
11. Fan, G.F.; Peng, L.L.; Hong, W.C. Short term load forecasting based on phase space reconstruction algorithm and bi-square kernel regression model. *Appl. Energy* **2018**, *224*, 13–33. [[CrossRef](#)]

12. Fan, S.; Chen, L. Short-term load forecasting based on an adaptive hybrid method. *IEEE Trans. Power Syst.* **2006**, *21*, 392–401. [[CrossRef](#)]
13. Liu, P.; Zheng, P.; Chen, Z. Deep Learning with Stacked Denoising Auto-Encoder for Short-Term Electric Load Forecasting. *Energies* **2019**, *12*, 2445. [[CrossRef](#)]
14. Ryu, S.; Noh, J.; Kim, H. Deep neural network based demand side short term load forecasting. *Energies* **2017**, *10*, 3. [[CrossRef](#)]
15. Ding, Y. *Data Science for Wind Energy*, 1st ed.; Chapman and Hall/CRC: London, UK, 2019. [[CrossRef](#)]
16. Abarbanel, H. *Analysis of Observed Chaotic Data*; Springer Science & Business Media: Berlin, Germany, 2012.
17. Takens, F. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980*; Springer: Berlin, Germany, 1981; pp. 366–381.
18. Noakes, L. The Takens embedding theorem. *Int. J. Bifurc. Chaos* **1991**, *1*, 867–872. [[CrossRef](#)]
19. Farmer, J.D.; Sidorowich, J.J. Predicting chaotic time series. *Phys. Rev. Lett.* **1987**, *59*, 845. [[CrossRef](#)] [[PubMed](#)]
20. Kantz, H.; Schreiber, T. *Nonlinear Time Series Analysis*; Cambridge University Press: Cambridge, UK, 2004.
21. Flores, J.J.; González, J.R.C.; Farias, R.L.; Calderon, F. Evolving nearest neighbor time series forecasters. *Soft Comput.* **2017**, *23*, 1039–1048. [[CrossRef](#)]
22. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [[CrossRef](#)]
23. Lampinen, J.; Zelinka, I. Mixed integer-discrete-continuous optimization by differential evolution. In Proceedings of the 5th International Conference on Soft Computing, Brno, Czech Republic, 9–12 June 1999; pp. 71–76.
24. Price, K.; Storn, R.M.; Lampinen, J.A. *Differential Evolution: A Practical Approach to Global Optimization*; Springer: Berlin, Germany, 2006.
25. Montgomery, D.C.; Jennings, C.L.; Kulahci, M. *Introduction to Time Series Analysis and Forecasting*; John Wiley & Sons: New York, NY, USA, 2015.
26. Banerjee, A.; Dolado, J.J.; Galbraith, J.W.; Hendry, D. *Co-Integration, Error Correction, and the Econometric Analysis of Non-Stationary Data*; OUP Catalogue, Oxford University Press: Oxford, UK, 1993.
27. Fuller, W. *Introduction to Statistical Time Series*; John Wiley & Sons: New York, NY, USA, 1976.
28. Van Gerven, M.; Bohte, S. Artificial neural networks as models of neural information processing. *Front. Comput. Neurosci.* **2017**, *11*, 114. [[CrossRef](#)] [[PubMed](#)]
29. Beale, H.D.; Demuth, H.B.; Hagan, M. *Neural Network Design*; PWS: Boston, MA, USA, 1996.
30. Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*; O'Reilly Media: Sebastopol, CA, USA, 2019.
31. Box, G.E.; Jenkins, G.M. *Time Series Analysis: Forecasting and Control*; Holden-Day: San Francisco, CA, USA, 1970.
32. Welch, G.; Bishop, G. *An Introduction to the Kalman Filter*; University of North Carolina at Chapel Hill: Chapel Hill, NC, USA, 1995.
33. Csáji, B.C. Approximation with artificial neural networks. *Fac. Sci. Eotvos Lornd Univ. Hung.* **2001**, *24*, 7.
34. Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; Volume 1.
35. R Development Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2008; ISBN 3-900051-07-0.
36. Energia, S.D. Short-Term Energy Market Manual. 2016. Available online: [https://www.cenace.gob.mx/Docs/MercadoCortoPlazo/Manual%20de%20Mercado%20de%20Energ%C3%ADa%20de%20Corto%20Plazo%20\(DOF%20SENER%2017-Jun-16\).pdf](https://www.cenace.gob.mx/Docs/MercadoCortoPlazo/Manual%20de%20Mercado%20de%20Energ%C3%ADa%20de%20Corto%20Plazo%20(DOF%20SENER%2017-Jun-16).pdf) (accessed on 5 October 2020). (In Spanish).

