Fast and Efficient Very Short-Term Load Forecasting using Analogue and Moving Average Tools

Uriel I. Lezama-Lope, *Student Member, IEEE*, Alberto Benavides-Vázquez Guillermo Santamaría-Bonfil and Roger Z. Ríos-Mercado

***Abstract*—The electricity market's continuous and secure operation depends on accurately predicting real-time demand. This study presents an innovative Analogue Moving Average (AnMA) method that uses classical statistical techniques like correlation, regression, and moving averages to improve the accuracy of load demand forecasting. AnMA is designed to correct for biases and unforeseen changes in load demand and offers several desirable attributes, such as high accuracy, speed, robustness, low maintenance, repeatability, and a low computational cost. The study evaluates the performance of AnMA against Naïve, exponential smoothing, and Autoregressive Moving Average (ARMA) benchmarks for forecasting horizons ranging from five minutes to two hours multi-step ahead, using data from the preceding four months. The results show that AnMA is competitive with the benchmarks in terms of accuracy while offering dramatically lower computational costs, making it an efficient and highly attractive method.**

***Index Terms*—real-time, very short-term load forecasting, analogue, green algorithms.**

I. Introduction

Real-time load demand forecasting is vital for efficient and reliable electrical systems globally. Accurate forecasts enable decision-making regarding dispatching generating units and scheduling transmission system maintenance and upgrades. Very Short-Term Load Forecasting (VSTLF) predicts load behavior over a few minutes to several hours, widely used in operating electricity markets, including open and controlled environments [1]. VSTLF aims to predict the near-term behavior of electrical systems, supporting informed decision-making, including economic dispatch, price setting, renewable energy coordination, system analysis, and interchange scheduling [2].

Accurate load demand forecasting is crucial for maintaining power systems' economic viability and security. Deviations in the forecast that underestimate demand can result in additional costs associated with maintaining generator operation through ancillary services such as regulation and reserves. Conversely, overestimating demand can lead to unnecessary energy purchases and over-generation, compromising the delicate balance between generation and load and jeopardizing the system's security [3].

The challenge of estimating a VSTLF involves finding a model that accurately predicts expected electricity demand and providing forecasts on time. The promptness of the forecast is of equal importance as its accuracy in scenarios where sudden changes in demand may occur, such as cascading failures or extreme weather events [4]. The relationship between accuracy and calculation time remains unresolved, and efforts to improve one aspect often compromise the other. To mitigate this trade-off, solutions have been proposed that involve increasing computational capacity, processing costs, and energy consumption.

Therefore, in the context of a real-time Electricity Market (RTM), system operators require efficient and reliable methods for VSTLF that can automatically forecast load demand on a 24/7 basis, providing results every five minutes and delivering them almost instantaneously. Such models must meet the following criteria:

- High accuracy: It is crucial, as it directly impacts operational efficiency and the reliability of market results.
- Speed: The model must provide results on time, ideally within ten seconds or less per series.
- Adaptability: the model must adapt to changes in patterns and sudden shifts in load demand caused by external factors by integrating the most up-to-date real-time data.
- Low computational cost: Essential when computing resources are limited and multiple forecasting methods are running on the same computer or shared with other processes.
- Robustness: The model must consistently converge and provide accurate forecasts for the market, even in cases of non-invertible matrices and numerical inaccuracies during regression processes.
- Low maintenance: The model parameters should require minimal manual intervention, and the number of tuning parameters should be kept to a minimum.
- Repeatability: The model should be deterministic, resulting in consistent results even when run multiple times. Retaining all input data is crucial for achieving reproducible results and obtaining approval from regulatory agencies during audits.

Uriel I. Lezama-Lope is with Universidad Autónoma de Nuevo León and Instituto Nacional de Electricidad y Energias Limpias, Mexico, e-mail: uriel.lezamaop@uanl.edu.mx.

Alberto Benavides-Vázquez and Roger Z. Ríos-Mercado are with Universidad Autónoma de Nuevo León, Mexico, e-mail: jose.benavidesvz@uanl.edu.mx, roger@yalma.fime.uanl.mx.

Guillermo Santamaría-Bonfil is with BBVA Mexico, Data Portfolio Manager Department, Unique experience and data general directorate, e-mail: guillermo.santamaria@bbva.com.

A broad discussion of forecasting requirements for the electricity markets is discussed by Dannecker [1].

This paper's primary scientific contribution is the development of the Analogies Moving Average (AnMA) approach for predicting real-time load demand. By integrating the advantages of based-analogies (An) and moving averages (MA) methods, AnMA offers a trustworthy and precise forecast that can rapidly respond to recent changes in the load. This innovative solution delivers a rare and coveted mix of speed, adaptability, and robustness, making it an ideal candidate for use in electricity market operations.

In addition, the proposed method design in this paper meets the specified requirements of RTM forecasting by utilizing continuous retraining with incoming real-time data for immediate adaptation to changes in data behavior. It is specifically designed for very short-term load forecasting across seasons, each with distinctive demand patterns. Furthermore, the method optimizes training using a "first-in, last-out" approach that discards the latest training data when new real-time data is introduced.

The five-minute and two-and-a-half hours test results demonstrated that the AnMA variants using Principal Component Regression (PCR), Lasso, and Ridge were more precise and faster than the other methods. This study's proposal not only outlines a method but also suggests a comprehensive framework that integrates various correlation and regression tools from both classical statistics and Machine Learning (ML). This sets a benchmark for future research aiming to improve load demand forecasting methods' efficiency.

The paper is structured as follows: Section II presents a thorough overview of previous research on VSTLF, and works related to the analogue method in other fields are present. Section III outlines the proposed AnMA method. The test procedures are explained in section III-B. The main test results are summarized in section IV. Section V discusses the key findings of the tests. Finally, Section VI summarizes all the research and outlines potential lines for future research.

II. Related work

Load demand forecasting has been the subject of extensive research [5], [6]. This section provides an overview of some of the most relevant studies, including those based on ML techniques such as Artificial Neural Networks (ANN) [7], [8] and Support Vector Regression (SVR).

A noteworthy study is documented by Capuno et al. [9], who presented a method for a real-time operation that employs a hybrid of algebraic prediction and SVR. Their method initially generates a first forecast and then employs SVR to rectify any variations caused by significant temperature and humidity changes. The proposed method in our work also generates a baseline forecast and adjusts it with real-time data.

ML algorithms can deliver high-accuracy results but require high computational cost during the training phase [3], [10], [11]. In addition, changes in load patterns may necessitate frequent retraining of ML models, leading to increased resource consumption and a carbon footprint [12]. More sophisticated methods, such as those incorporating Deep Learning (DL) models, have been proposed for load demand forecasting [13], [14]. However, these methods face lengthy training times, further exacerbated with each retraining due to adding new demand data. Therefore, statistical learning should balance accuracy and computational efficiency, reducing resource usage and the carbon footprint as much as possible. On the contrary, linear regression-based methods, which are computationally inexpensive but may not be as accurate due to their linear features, frequently employ environmental factors as regression variables to model electricity demand [15].

The Analogies method, initially used in meteorology and climatology, has been proposed as another approach to time series forecasting by Monache et al. [16], This method assumes that forecasting errors between similar days will likely occur again. The grouping of days is performed by calculating the distance between all the days to select the most similar, resulting in an analog space consisting of days with similar features and subsequent error data. This model has been applied successfully in forecasting renewable resources in photovoltaic plants and wind farms by Alessandrini et al. [17], [18]. This approach makes a forecast from a set of the most similar past predictions. Azevedo et al. [19] and Santis et al. [20] describe its dynamic time scan forecasting based on analogies as a fast method for large datasets to forecast wind speed time series. The scan procedure uses polynomial regression models as a distance measure to identify similar patterns throughout the time series. The author's innovative idea was to use different similarity metrics as variables in the An method.

The multi-seasonal characteristics of electricity demand have been thoroughly studied by Gould et al. [21] and Livera et al. [22], who have identified multiple seasonalities in the time series. These seasonalities define demand patterns between days and seasons. Dudek [23] conducted a study on the seasonal characteristics of demand and determined certain patterns using tools such as Nearest Neighbors (NN). He found the criteria between days of the same season by minimizing the distances between a window of the latest data and the time series.

In a follow-up work, Dudek [24] proposes a one-day-ahead load demand forecasting model based on the search for patterns using NN and combining sample selection with Ordinal Least Square (OLS). This model refines the regression using variable reduction techniques such as stepwise regression, Lasso, and PCR. The study's central aim is to accurately identify days with seasonal cycle patterns and classify these days into weekdays, Saturdays, Sundays, and holidays. The ideas presented by Dudek [24] are comparable to those in our proposed method, including the selection of samples, the regression of these samples to determine a model, and the forecast calculation based on the subsequent data from the samples.

The work by Ngo et al. [25] proposes a method for ultra-short-term load demand forecasting based on the robust Holt-Winters double seasonal model. The authors evaluated the performance of the proposed method on distribution feeder loads and found that it had good prediction accuracy. Exponential Smoothing (ETS) methods are widely recognized for

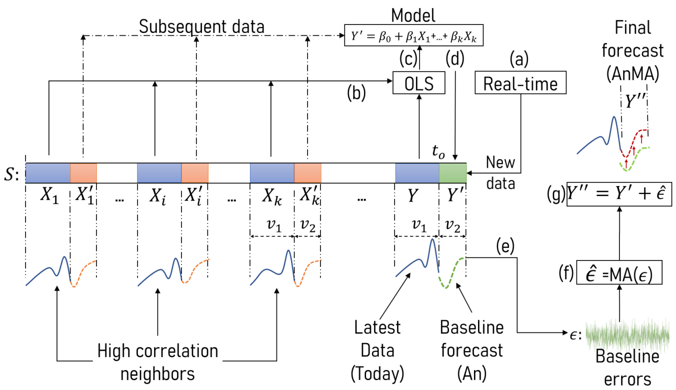

Fig. 1. Main steps of AnMA method. (a) Data collected in real-time, (b) select neighbors, (c) get regression model, (d) calculate forecast An, (e) record the An series error, (f) build a MA model from ϵ, (g) calculate forecast AnMA.

their efficiency, performance, and low computational cost and are commonly used as a benchmark in forecasting.

A novel approach for VSTLF is presented in our study, which combines an analogue approach with a moving averages model. This is the first time that such a combination of similarity metrics has been reported in a unified framework, to the best of our knowledge. A significant contribution to the field has been made by utilizing regression models, selecting high-correlation days, and correcting baseline forecast errors in real-time. The novel methodology has the potential to enhance the accuracy and efficiency of VSTLF in the electricity markets.

III. Materials and Methods

The AnMA method presented in this work is a comprehensive approach with five distinct stages. i) the collection of load demand data is performed; ii) the forecasting is conducted using an analogue based (An) method; iii) the residuals are obtained from the An forecasts; iv) calculating a Moving Average (MA) model from the residual time series; v) the forecast is estimated by combining the An and MA models. A visual representation of the key steps of the AnMA method is provided in Fig. 1.

Stage i. Load demand dataset

This study uses a time series load demand data from a representative region in Mexico. The data covers a span of one year with a sampling frequency of five minutes. The data have seasonal patterns with increased load demand during the hot seasons and decreased load demand during the cold seasons.

Stage ii. Analogies (An)

The time series, represented by the notation $S = (s_1, s_2, \ldots, s_p)$, consists of p consecutive periods. A subsequence is a collection of consecutive periods within the time series.

The Analogies (An) method starts by selecting the most recent data, denoted as

$$Y = y_j : j \in [|S| - v_1, |S|], \tag{1}$$

where v_1 is a significant time lag in the series S. This lag, v_1, is determined using the Auto-correlation Function (ACF) [26]. It is important to note that the value of v_1 must be greater than the forecast horizon, denoted as v_2, for Y'.

A sequence of the time series S, represented as $S' = (s_1, s_2, \ldots, s_n)$, is selected where $s_n = |S| - (v_1 + v_2)$. From this sequence, a group of n smaller sequences, each having a length of v_1, is extracted and referred to as the set X. These smaller sequences in the set X are called the neighborhood, and each individual sequence is called a neighbor.

Afterward, the k-nearest neighbors (k-NN) method [27] is utilized to select a subset of subsequences, referred to as the k nearest neighbors, from X. These k nearest neighbors, represented by X_k, are the k subsequences in X with the highest correlation with Y. X_k can be visualized as the black-colored subsequences in Fig. 1. The Pearson correlation coefficient is used as the similarity metric to measure the correlation between Y and each sequence in X. The selection process calculates the correlation between Y and each subsequence Xi of series S', going back one period at a time and storing the resulting coefficients. The neighbors are sorted by correlation level, and the k neighbors with the highest correlation are selected. To avoid selecting neighbors with high correlation and a few-period delay between positions i and j, only neighbors whose distance $|i-j|$ exceeds $v_1 \cdot \delta$ are chosen, where δ is a parameter close to 1. Therefore, if a neighbor in position i is chosen for its high correlation, only neighbors at a distance greater than $v_1 \cdot \delta$ from position i may be selected in position j. This practice ensures a more diverse selection and avoids biases from considering practically identical neighbors. The value of k is user-defined.

The OLS regression model with backward stepwise elimination is utilized to determine the model that explains the dependent variable Y by utilizing the X_k neighbors as independent variables.

The An method then produces a new set of subsequences, X_k', from the previously selected subsequences X_k. X_k' includes the next v_2 inmediate consecutive data for each neighbour X_k, and can be visualized as the orange-colored subsequences immediately following the neighbors in Fig. 1.

The regression model, obtained through the OLS, is then applied to X_k' to generate the baseline forecast Y' for the next time period in the time series.

A result of the selection process of the X_k neighbors and their resulting series X_k' can be visualized in Fig. 2. The figure displays the X_k neighbors on the left of a vertical line, along with the latest data point Y. The subsequent series X_k' from these neighbors and the forecast Y' are displayed on the right side of the line. There is a strong correlation between the current data Y and the selected X_k neighbors. It's worth noting that there is no observed load demand data following Y on the right side of the figure, only the forecasted value Y'.

Stage iii. An residuals estimation

The An method identifies repeating patterns in the data and uses those patterns to make a baseline forecast. However, over time, this baseline forecast may become biased and susceptible

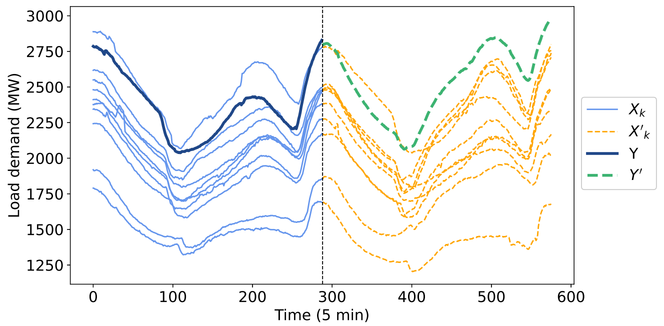

Fig. 2. Left: neighbors X_k with the highest correlation to current data Y. Right: subsequent data X'_k and the forecast Y'.

to errors if there are sudden changes in the demand data. To avoid this, making real-time adjustments while the forecast is being calculated is necessary.

From the baseline forecast results obtained by the An method, a series of errors between Y' and the actual demand S is created. The error series is referred to as $\epsilon = \epsilon_t : t \in [-v, 0]$ and contains the most recent v errors of the An method. The error series is important because it corrects the new forecasts being calculated by An. It is essential to keep the results of the baseline forecast obtained using the An method, as they will be used to make these corrections.

Stage iv. MA estimation

In the fourth stage, a model MA of order q is set from the errors (ϵ) of An to calculate the baseline forecast errors $\hat{\epsilon}$.

Stage v. AnMA forecast

Finally, the An and MA composite method computes the final forecast, denoted as Y'', by adding the calculated $\hat{\epsilon}$ to the result of the An forecast Y', as expressed by the equation

$$Y'' = Y'(t) + \hat{\epsilon}(t), \; \forall t = 1, \ldots, w. \tag{2}$$

A. AnMA variants

The AnMA methodology initially used purely statistical tools but later incorporated tools from the statistical and machine learning fields in neighbor selection and regression. The basic version of AnMA uses the Pearson coefficient for distance metric and OLS for regression. However, the first variation altered the neighbor distance measure from Pearson correlation to Euclidean distance in the selection phase. The second variation used linear dimensional reduction methods such as PCR and Partial Least Square (PLS) and shrinkage techniques like Lasso and Ridge regression to minimize the penalized residual sum of squares in the regression phase. Other models used include Random Forest (RF), Bagging, and Boosting, which employ the concept of ensembles to produce more accurate predictions. Tests of the AnMA method were conducted using the combinations of these tools.

B. Method Validation

Two tests will be conducted. The first will compare five-minute forecasts using AnMA and variants to Holt-winters Additive (HWA), Holt-winters Multiplicative (HWM), and Naïve models. The second test (prefix `X`) will compare two-and-a-half-hour forecasts with thirty periods in five-minute intervals using AnMA and variants, HWA, HWM, and Autoregressive Moving Average (ARMA), based on accuracy, calculation time, and CPU usage. A subsequence size of 288 data points will be used for v_1, with AnMA variants using five neighbors and MA using 15 periods for baseline forecast error correction. AnMA variants will use PLS, PCR, Lasso, Ridge, RF, Bagging, and Boosting regression models with two different neighbor selection metrics: Pearson's coefficient (default) and Euclidean distance. Some variants will omit the MA correction, indicated by the prefix `An`.

Auto-ARIMA was attempted [28], [29], but it was unfeasible due to the high processing time per real-time forecast. Instead, a faster 7-day lag ARMA (288×7 periods) to capture daily/weekly electricity demand seasonality is used [22] and implement an error correction using 15 periods lag MA model. For validation in time series forecasting, k-fold cross-validation will be used, following Bergmeir et al. [30]. All methods were tested with a single processing core except for ARMA, which utilized the `statsmodel` [31] library to exploit all available computer resources for faster processing.

The forecast will be for the summer of 2010, using four months of training data to predict the next 30 data periods. When new real-time data is added, the last training data will be trimmed with a first-in, last-out approach.

The accuracy metrics will be expressed in the Mean Percentage of Absolute Error (MAPE) and Mean Absolute Error (MAE). The computation time will be measured in seconds. And finally, the usage of CPU in percentage, which is recorded on a minute-by-minute basis.

A factorial design of experiments is recommended to explore the impact of varying the number of neighbors k, subsequence size v_1, and the q order or MA. The optimal parameter values were determined to be $k = 10$, $v_1 = 288$, and MA=15. Specifically, these parameter values correspond to a number of neighbors equivalent to ten different days, a subsequence length of one day, and an MA order of one- and-a-half-hours.

IV. RESULTS

The summary of two tests shows that the AnMA variants using PCR, Lasso, and Ridge methods with Pearson's coefficient as the similarity metric were the fastest. The Naïve method had minimal time and CPU usage, while HWM, HWA, and ARMA were slower. AnMA with PCR, Lasso, and Ridge had lower MAPE and MAE compared to the benchmarks. Although ARMA achieved the highest accuracy, it used all eight CPU cores, whereas AnMA produced similar results using only one core. The benchmarks Naïve, HWA, and HWM also utilized one core with low CPU usage. The results are presented in Tables I and II.

TABLE I

RESULTS FOR THE FIVE-MINUTE TEST. BENCHMARK METHODS ARE MARKED WITH AN ASTERISK.

Method	MAE	MAPE	Time (seconds)
ARMA ($\times 8$ cores)*	4.4549	0.1633	12.6203
AnMA-PCR	6.8055	0.2512	1.6022
AnMA-Lasso	7.8350	0.2892	1.5335
AnMA-Ridge	7.8380	0.2893	1.6184
AnMA-OLS	8.1875	0.3027	1.4989
AnMA-Lasso-euclidian	8.7921	0.3250	1.9696
AnMA-Ridge-euclidian	8.7946	0.3251	1.9688
AnMA-OLS-euclidian	8.8928	0.3285	1.8224
Naïve*	9.0309	0.3310	0.1000
HWM*	9.8225	0.3604	5.5842
HWA*	9.9119	0.3637	3.5475
AnMA-RF	10.4505	0.3902	1.6139
AnMA-Boosting	12.1386	0.4535	1.5299
AnMA-RF-euclidian	12.2511	0.4579	1.8325
AnMA-Bagging	12.5724	0.4693	1.5949
AnMA-Boosting-euclidian	14.2913	0.5325	1.7915
AnMA-Bagging-euclidian	14.8624	0.5541	1.9805
An-RF	22.5132	0.8336	1.6108
An-Boosting	22.5968	0.8349	1.5268
An-Bagging	23.2854	0.8623	1.5916
An-RF-euclidian	27.7836	1.0306	1.8295
An-PLS-euclidian	28.0604	1.0374	2.0078
An-Boosting-euclidian	28.0604	1.0374	1.7885
AnMA-PLS-euclidian	28.0604	1.0374	2.0112
An-Bagging-euclidian	28.8176	1.0682	1.9771
An-Ridge	30.2281	1.1127	1.6151
An-Lasso	30.2285	1.1127	1.5303
AnMA-PLS	31.5625	1.1638	1.5670
An-PLS	31.5625	1.1638	1.5638
An-OLS	32.4370	1.1929	1.4958
An-Ridge-euclidian	34.2231	1.2681	1.9654
An-Lasso-euclidian	34.2239	1.2681	1.9663
An-PCR	35.3237	1.3029	1.5989
An-OLS-euclidian	36.2647	1.3453	1.8193

Statistical tests were conducted to determine significant differences in the means of absolute errors between AnMA-PCR, Naïve, HWA, HWM, and ARMA in the five-minute and two-and-a-half-hour tests. AnMA-PCR had lower absolute errors on mean compared to benchmark methods, as supported by Figs. 3 and 4. However, in the two-and-a-half-hour test, the Wilcoxon paired test showed that the absolute errors of ARMA were significantly smaller than those of AnMA-PCR, with a p-value $= 0.0000$, indicating that ARMA has, on mean, lower absolute errors than AnMA-PCR, which was unexpected based on the initial comparison with the other benchmark methods.

The mean CPU execution time of AnMA-PCR was 1.6022 seconds, used into 1.5959 seconds for neighbor selection, while 0.0018 seconds were used for regression. It should be noted that the time for baseline correction was negligible.

The carbon footprint for each method was calculated following the method described by Lannelongue et al. [32], which involved multiplying the CPU usage and core count and factoring in the location and computer specifications. The amount of carbon used to generate electricity varies between different countries. The results, which display energy consumption (in kWh) and CO_2 emissions, can be found in Table III.

The benchmark models HWA, HWM, and ARMA were implemented using `statsmodels` developed by Seabold and Perktold [31] in Python. The AnMA was coded in Python 3.10.

TABLE II

RESULTS FOR THE TWO-AND-A-HALF-HOURS TEST. BENCHMARK METHODS ARE MARKED WITH AN ASTERISK.

Method	MAE	MAPE	Time (seconds)
XARMA ($\times 8$ cores)*	29.2242	1.0739	12.6362
XAnMA-PCR	30.5737	1.1259	1.6041
XAnMA-Lasso	32.5393	1.1977	1.5350
XAnMA-Ridge	32.5470	1.1980	1.6196
XAnMA-OLS	32.5020	1.1987	1.5000
XAnMA-Lasso-euclidian	34.8934	1.3012	1.9728
XAnMA-Ridge-euclidian	34.8994	1.3014	1.9676
XAnMA-OLS-euclidian	35.0092	1.3068	1.8227
XHWA*	37.7099	1.3732	3.5390
XHWM*	39.6985	1.4297	5.5912
XAn-Lasso	47.9497	1.7618	1.5340
XAn-Ridge	47.9530	1.7619	1.6186
XAnMA-RF	48.0756	1.7740	1.6161
XAn-PLS	48.4071	1.7805	1.5699
XAnMA-PLS	48.4071	1.7805	1.5709
XAn-OLS	48.6235	1.7839	1.4993
XAnMA-Bagging	48.4061	1.7852	1.5959
XAnMA-Boosting	48.8399	1.7997	1.5308
XAn-PCR	51.3148	1.8908	1.6030
XAn-Lasso-euclidian	53.3446	1.9792	1.9713
XAn-Ridge-euclidian	53.3475	1.9793	1.9661
XAn-euclidian	53.3562	1.9845	1.8220
XAnMA-RF-euclidian	53.4545	1.9859	1.8340
XAnMA-Boosting-euclidian	54.2142	2.0129	1.7923
XAnMA-Bagging-euclidian	54.7376	2.0325	1.9889
XAn-Boosting	60.1638	2.2109	1.5301
XAn-RF	60.2697	2.2162	1.6154
XAn-Bagging	60.5981	2.2280	1.5948
XAn-Boosting-euclidian	68.9374	2.5559	1.7915
XAnMA-PLS-euclidian	68.9374	2.5559	2.0128
XAn-PLS-euclidian	68.9374	2.5559	2.0113
XAn-RF-euclidian	69.1077	2.5612	1.8333
XAn-Bagging-euclidian	69.4539	2.5739	1.9874

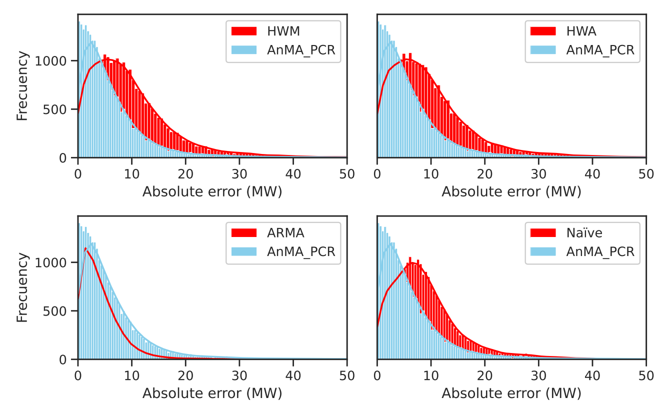

Fig. 3. Comparison of absolute error distributions between `AnMA-PCR` and benchmarks, five-minute test.

TABLE III

CARBON FOOTPRINT PER METHOD [32].

Method	Runtime	Cores	% CPU	kWh	kg-CO_2e
ARMA	92:53:00	8	92.0	7.7500	3.3400
HWM	41:06:00	1	17.1	1.0400	0.4472
HWA	26:07:00	1	16.8	0.6581	0.2839
AnMA-Ridge	11:54:00	1	18.0	0.3010	0.1298
AnMA-PCR	11:47:00	1	15.0	0.2952	0.1273
AnMA-Lasso	11:17:00	1	27.5	0.2941	0.1268
Naïve	00:44:00	1	1.0	0.0078	0.0180

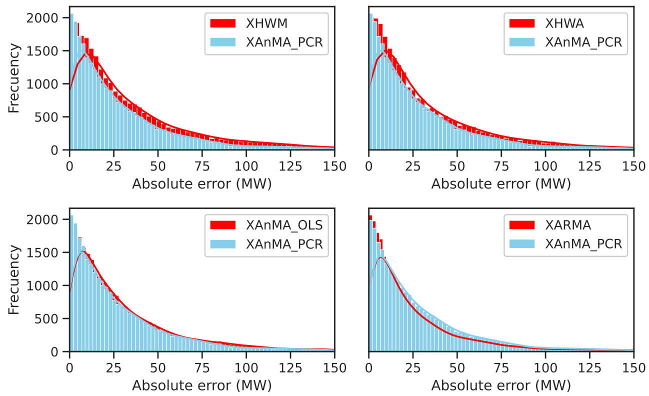

Fig. 4. Comparison of absolute error distributions between `AnMA-PCR` and benchmarks, two-and-a-half hours' test.

The hardware used in all tests was a 64-bit system with 64GB RAM and Intel(R) i7(R) CPU 11700 @ 2.50GHz, 65W.

A. Data Availability

This work's data and results are in the repository: [33]. Also, the code is available in [34].

V. Discussion

The Naïve forecast had the lowest CPU time in the five-minute test, while AnMA with its PCR, Lasso, and Ridge variants was faster on mean than HWA, HWM, and ARMA in both the five-minute and two-and-a-half-hour tests. AnMA with PCR, Lasso, and Ridge were more accurate than the Naïve forecast, HWA, and HWM, in terms of MAPE and MAE. However, the ARMA was more accurate than An-MA but relatively more expensive. In contrast, AnMA obtained its results in a dramatically shorter time of fewer than two seconds versus twelve seconds of ARMA.

AnMA was also very efficient, using only one CPU core and less than one kilowatt-hour, while ARMA required eight cores and more than seven kilowatt-hours. The benchmark methods (HWA, HWM, and Naïve) used a core too.

The most accurate variants of AnMA used Pearson's coefficient as a similarity metric and employed dimension reduction and shrinking in regression techniques, resulting in the highest accuracy in PCR, Lasso, and Ridge.

Due to its calculation speed, AnMA is particularly suitable for real-time applications and can be shared with other processes, providing an indirect advantage.

The method's speed is attributed to its design, which combines classical statistical tools, such as Pearson correlation, linear regression, and moving average, optimizing time and resources.

VI. Conclusions

The AnMA, a real-time load demand forecasting method that merges Analog and Moving Average methods in a flexible framework, was outlined. This method provides accurate results using less energy and less time than the benchmarks. Additionally, it has the advantages of low computational cost and adaptability to new data. Therefore it is suitable for real-time operation.

Furthermore, while the energy differences between the methods in this study may be negligible in practice, some machine-learning approaches are much more expensive and require long training periods in data centers with high energy consumption. In contrast, AnMA exemplifies an efficient, cost-effective, and energy-efficient algorithm.

Future improvements will include selecting neighbors from pre-selected subsets of days and parallelizing neighbor searches in the algorithm while ensuring its speed and computational efficiency. Additionally, a study is planned to compare forecasting techniques utilizing pre-trained deep learning models and similarity pattern methods with our AnMA, with the primary objective of evaluating these algorithms' adaptability to changing patterns, such as seasonal periods, holidays, and unforeseen events like the SARS-COV-2 outbreak. We anticipate that our AnMA algorithms will inspire the development of more accurate and cost-effective green algorithms.

Acknowledgment

A scholarship for graduate studies from the Mexican Council for Science and Technology CONACyT supported the research of UILL and ABV authors. Also, UILL acknowledges the support of INEEL (FICYDET 220819-la-EXT-385).

Disclaimer: The third author wants to clarify that the present work represents his own opinion and research lines, and not necessarily the opinion of BBVA Mexico.

References

[1] L. Dannecker, *Energy Time Series Forecasting: Efficient and Accurate Forecasting of Evolving Time Series from the Energy Domain*. Springer Fachmedien Wiesbaden, 2015.

[2] J. Y. Fan and J. D. McDonald, "A real-time implementation of short-term load forecasting for distribution power systems," *IEEE Transactions on Power Systems*, vol. 9, no. 2, pp. 988–994, 1994.

[3] S. Yadav, B. Tondwal, and A. Tomar, "Models of load forecasting," in *Prediction Techniques for Renewable Energy Generation and Load Demand Forecasting* (A. Tomar, P. Gaur, and X. Jin, eds.), vol. 956 of *Lecture Notes in Electrical Engineering*, ch. 7, pp. 111–130, Singapore, Singapore: Springer, 2023.

[4] Y. Yang, T. Nishikawa, and A. E. Motter, "Small vulnerable sets determine large network cascades in power grids," *Science*, vol. 358, no. 6365, p. 3184, 2017.

[5] I. K. Nti, M. Teimeh, O. Nyarko-Boateng, and A. F. Adekoya, "Electricity load forecasting: a systematic review," *Journal of Electrical Systems and Information Technology*, vol. 7, no. 13, pp. 2314–7172, 2020.

[6] A. Azeem, I. Ismail, S. M. Jameel, and V. R. Harindran, "Electrical load forecasting models for different generation modalities: A review," *IEEE Access*, vol. 9, pp. 142239–142263, 2021.

[7] I. Barić, R. Grbić, and E. K. Nyarko, "Short-term forecasting of electricity consumption using artificial neural networks – an overview," in *42nd International Convention on Information and Communication Technology, Electronics And Microelectronics (MIPRO)*, (Opatija, Croatia), pp. 1076–1081, May 2019.

[8] J. Gunawan and C.-Y. Huang, "An extensible framework for short-term holiday load forecasting combining dynamic time warping and LSTM network," *IEEE Access*, vol. 9, pp. 106885–106894, 2021.

[9] M. Capuno, J.-S. Kim, and H. Song, "Very short-term load forecasting using hybrid algebraic prediction and support vector regression," *Mathematical Problems in Engineering*, vol. 2017, p. 8298531, 2017.

[10] B. Ibrahim, L. Rabelo, E. Gutierrez-Franco, and N. Clavijo-Buritica, "Machine learning for short-term load forecasting in smart grids," *Energies*, vol. 15, no. 21, 2022.

[11] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M4 competition: 100 000 time series and 61 forecasting methods," *International Journal of Forecasting*, vol. 36, no. 1, pp. 54–74, 2020.
[12] S. Tiwari, A. Jain, N. M. O. S. Ahmed, Charu, L. M. Alkwai, A. K. Y. Dafhalla, and S. A. S. Hamad, "Machine learning-based model for prediction of power consumption in smart grid-smart way towards smart city," *Expert Systems*, vol. 39, no. 5, p. e12832, 2022.
[13] C. Tong, L. Zhang, H. Li, and Y. Ding, "Temporal inception convolutional network based on multi-head attention for ultra-short-term load forecasting," *IET Generation, Transmission & Distribution*, vol. 16, no. 8, pp. 1680–1696, 2022.
[14] A. M. N. C. Ribeiro, P. R. X. do Carmo, P. T. Endo, P. Rosati, and T. Lynn, "Short- and very short-term firm-level load forecasting for warehouses: A comparison of machine learning and deep learning models," *Energies*, vol. 15, no. 3, p. 750, 2022.
[15] H. Nano, S. New, A. Cohen, and B. Ramachandran, "Load forecasting using multiple linear regression with different calendars," in *Distributed Energy Resources in Microgrids* (R. K. Chauhan and K. Chauhan, eds.), ch. 16, pp. 405–417, Greater Noida, India: Academic Press, 2019.
[16] L. D. Monache, F. A. Eckel, D. L. Rife, B. Nagarajan, and K. Searight, "Probabilistic weather prediction with an analog ensemble," *Applied Energy*, vol. 141, pp. 3498–3516, 2013.
[17] S. Alessandrini, L. Delle Monache, S. Sperati, and G. Cervone, "An analog ensemble for short-term probabilistic solar power forecast," *Applied Energy*, vol. 157, pp. 95–110, 2015.
[18] S. Alessandrini, L. Delle Monache, S. Sperati, and J. Nissen, "A novel application of an analog ensemble for short-term wind power forecasting," *Renewable Energy*, vol. 76, pp. 768–781, 2015.
[19] M. Azevedo, R. Ruiz-Cárdenas, L. Brioschi, and M. Oliveira, "Dynamic time scan forecasting for multi-step wind speed prediction," *Renewable Energy*, vol. 177, pp. 584–595, 2021.
[20] R. B. D. Santis, T. S. Gontijo, and M. A. Costa, "Dynamic time scan forecasting: A benchmark with m4 competition data," *IEEE Latin America Transactions*, vol. 21, no. 2, pp. 320–327, 2023.
[21] P. G. Gould, A. B. Koehler, J. K. Ord, R. D. Snyder, R. J. Hyndman, and F. Vahid-Araghi, "Forecasting time series with multiple seasonal patterns," *European Journal of Operational Research*, vol. 191, no. 1, pp. 207–222, 2008.
[22] A. M. D. Livera, R. J. Hyndman, and R. D. Snyder, "Forecasting time series with complex seasonal patterns using exponential smoothing," *Journal of the American Statistical Association*, vol. 106, no. 496, pp. 1513–1527, 2011.
[23] G. Dudek, "Pattern similarity-based methods for short-term load forecasting — Part 1: Principles," *Applied Soft Computing*, vol. 37, pp. 277–287, 2015.
[24] G. Dudek, "Pattern-based local linear regression models for short-term load forecasting," *Electric Power Systems Research*, vol. 130, pp. 139–147, 2016.
[25] V. Ngo, W. Wu, B. Zhang, Z. Li, and Y. Wang, "Ultra-short-term load forecasting using robust exponentially weighted method in distribution networks," in *2015 IEEE Power & Energy Society General Meeting*, (Denver, USA), pp. 1–5, July 2015.
[26] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*. New York: Wiley, 5th ed., 2015.
[27] F. Martínez, M. P. Frías, M. D. Pérez, and A. J. Rivera, "A methodology for applying k-nearest neighbor to time series forecasting," *Artificial Intelligence Review*, vol. 52, no. 3, pp. 2019–2037, 2019.
[28] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: the forecast package for R," *Journal of Statistical Software*, vol. 26, no. 3, pp. 1–22, 2008.
[29] F. Garza, M. M. Canseco, C. Challú, and K. G. Olivares, "StatsForecast: Lightning fast forecasting with statistical and econometric models." PyCon Salt Lake City, Utah, 2022. https://github.com/Nixtla/statsforecast.
[30] C. Bergmeir, R. J. Hyndman, and B. Koo, "A note on the validity of cross-validation for evaluating autoregressive time series prediction," *Computational Statistics & Data Analysis*, vol. 120, pp. 70–83, 2018.
[31] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with Python," in *Proceedings of the 9th Python in Science Conference (SciPy 2010)* (S. van der Walt and J. Millman, eds.), (Austin), pp. 92–96, 2010.
[32] L. Lannelongue, J. Grealey, and M. Inouye, "Green algorithms: Quantifying the carbon footprint of computation," *Advanced Science*, vol. 8, no. 12, p. 2100707, 2021. http://calculator.green-algorithms.org/.
[33] "https://bit.ly/3RZm79z,"
[34] "https://github.com/urieliram/analog,"
[35] "http://yalma.fime.uanl.mx/r̃oger/,"

1st author Uriel earned a Bachelor of Science degree in Electrical Engineering, specializing in Electrical Power Systems, in 2003. Later, in 2010, he completed a Master's in Systems Engineering from the National Polytechnic Institute in Mexico. In 2009, he joined the National Institute of Electricity and Clean Energy as a researcher, where he developed advanced software for power system operations and contributed to the development of tools for managing the Mexican Electricity Market, including real-time systems. Currently, Uriel is a doctoral candidate in Systems Engineering at UANL in Mexico.

2nd author Alberto has a bachelor's degree in Philosophy and another in Multimedia. After that, he got a master's degree in System Engineering and is currently a doctoral student in the Department of Mechanical and Electrical Engineering at UANL, Mexico. His areas of interest are data science, time series forecasting, natural language processing, and data visualization for academic and business applications.

3rd author Guillermo holds a Ph.D. in Computer Science from Tecnologico de Monterrey. He has extensive experience in time series forecasting, feature selection, machine learning, and complexity analysis using information-based measures, with a decade of research experience at CONACYT-UADY. Currently, he holds an Expert Data Scientist position at BBVA, where he leverages his expertise to provide key insights and data-driven decision-making models and algorithms.

4th author Roger is a professor of Operations Research at UANL in Mexico, holding a Ph.D. in the same field from the University of Texas at Austin. He has held visiting scholar positions at the University of Colorado, Barcelona Tech, and University of Texas at Austin. His research focuses on designing efficient solutions to complex discrete optimization problems. His work covers districting, location analysis, healthcare, transportation systems, and scheduling, and has been published in top journals in the field. He is an Area Editor for *Computers & Operations Research* and a Fellow of the Mexican System of Research Scientists, the Mexican Academy of Sciences, and the Mexican Academy of Computer Sciences. More information on his work can be found at [35].