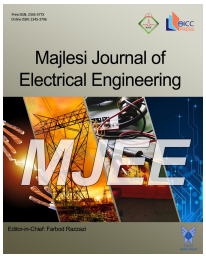

Volume 18, Issue 3, 182442 (1-10)

Majlese Journal of Electrical Engineering (MJEE)

https://dx.doi.org/10.57647/j.mjee.2024.180342

Comparison of data filtering methods effects on smart grid load forecasting

Vikash Kumar[1,*], Rajib Kumar Mandal[2]

National Institute of Technology Patna, Patna, Bihar, India.

*Corresponding author: vikashk.phd19.ee@nitp.ac.in

Original Research

Received:
23 June 2024
Revised:
10 July 2024
Accepted:
25 July 2024
Published online:
3 September 2024

Abstract:

The integration of advanced metering technology in power systems has enabled real-time data access for every node in a smart grid. As a result, the power system can now access large volumes of data. This vast amount of data requires an alternative method of analysis. Machine learning-based load forecasting technologies are being applied in this scenario. However, this massive data collection needs to be processed through the appropriate data pre-processing method, such as the removal of noise, outliers, and erroneous data, the detection of missing data, the normalization of widely divergent datasets, etc., to improve the effectiveness of the load forecaster. Thus, to eliminate the various kinds of errors and outliers present in the data that was directly obtained from smart meters, this study analyses and compares the efficacy of eight distinct smoothing and filtering techniques as a novel contribution of this work. Using the processed data acquired, a neural network-based load forecasting model was developed to compare the efficacy of the various pre-processing approaches. This study makes use of real-time data obtained from the smart meter placed at a node within the NIT Patna campus. The proposed moving average filter surpasses the other methods for filtering and smoothing the raw data by an average MAPE of 2.66, according to the load forecasting results that were obtained.

Keywords: Smart grid; Data pre-processing; Normalization; Neural network; Load forecasting

1. Introduction

In many areas of knowledge, from management to process control, from science to engineering, data analysis serves as the foundation for research. Attributes that are both symbolic and numerical are used to collect data on a certain subject. This data comes from a variety of sources, including sensors with various levels of reliability and complexity. An improved knowledge of the phenomenon of interest results from the analysis of these data. Finding information that can be used to solve issues or make decisions is the primary goal of any data analysis. But, issues with the data can make this difficult. Most often, data flaws are not discovered until after data analysis has begun. Before using data, pre-processing is necessary. The idea of transforming unclean data into clean data is known as data pre-processing. The algorithm must be able to quickly analyze the characteristics of the data for a model to be accurate and dependable in its predictions.

The bulk of real-world datasets applied for machine learning are very likely to contain missing data, inconsistent results, and noise due to their different origins. Thus, improving the general level of data quality requires data processing. The overall statistics of the data may not accurately reflect the presence of missing or duplicate values. Outliers and inconsistent data points commonly cause false predictions by interfering with the model's general learning process. To make quality assessments, one needs high-quality data. Data pre-processing is essential to obtain this high-quality data; otherwise, it would be a situation of "garbage-in, garbage-out." [1]. Data pre-processing is a data mining technique for transforming raw data into a usable and efficient format. Data pre-processing refers to the steps involved in transforming or encoding data so that it may be easily interpreted by a computer [2]. Data pre-processing can have a variety of objectives. One may be interested in learning more about the nature of the data or modifying the data's structure in addition to fixing data issues like corrupted data

Abbreviations

Abbreviation	Explanation
μ	Mean of data
σ	Standard deviation
FFNN	Feed-forward neural network
GF	Gaussian Filter
LOESS	Locally estimated scatter plot smoothening
LOWESS	Locally weighted scatter plot smoothening
MAF	Moving average filter
MAPE	Mean absolute percentage error
MMF	Moving median filter
NN	Neural network
RLOESS	Robust Locally estimated scatter plot smoothening
RLOWESS	Locally weighted scatter plot smoothening
RMSE	Long-term load forecasting
SGOLAY	Mean absolute percentage error
THI	Multiple linear regression

or irrelevant or missing attributes in data sets (e.g., degrees of granularity) to better prepare the data for analysis. In this paper few steps of data pre-processing are discussed. A few steps of data pre-processing are data reduction, data cleaning, dimensionality reduction, and, data integration.
The actual load data's normalization and filtration processes are both included in the pre-processing. Normalization is a method for effectively organizing data in a database [3]. The normalization process has two basic goals: to minimize redundant data (data that is stored in several tables) and to guarantee that data relationships make sense (only storing related data in a table). Normalization, in simple terms, ensures that all data looks and reads the same way across all records. Filtering is another method for removing undesirable components or noise from a signal. Before the data is used to train a load forecaster, it can be filtered using a variety of filtering techniques [4]. Data filtration is extensive. Data filtering is a solution that deals with basic issues like incorrect data at one end of the range. It deals with noisy data at the other end. A variety of data preparation methods rely on data filtering to get rid of unwanted information in the time, frequency, or time-frequency domain. The optimum filtering method should eliminate redundant characteristics with the least amount of distortion of the relevant signal characteristics [5].
In almost every industry, forecasting plays an important role. The power grid considered the most complicated man-made system on the earth, is controlled by electric utilities to provide electricity to more than five billion people around the world [6]. However, the process of producing electrons and transmitting them to electrical appliances is not easy. The product of the electric power industry, energy, cannot be held in significant quantities using current technologies, unlike many other firms that use inventories to store and buffer their goods and services. Therefore, it is necessary to produce and distribute electricity as soon as possible. In other words, utilities must consistently maintain supply and demand balance [7]. Since load forecasting is so important in utility operations, erroneous load forecasts can put a utility company in a financial bind or even lead to bankruptcy. While load forecasting is an important input for power system operations and planning, incorrect load projections can result in equipment failures or even a system-wide blackout. Therefore, to obtain effective load forecasting performance and consequently, effective energy management, quantitative analyses of data filtering strategies acquired in various forecasting systems are needed. This paper presents a trustworthy comparison of several filtering methods that, when used in conjunction with any load forecasting model, show to be more useful for a hypothetical future of smart grids. In this regard, the contribution of this work may be summarized as follows:
(i) Implementation of various data filtering methods for preprocessing the load data of a campus grid to forecast the load at various nodes.
(ii) Forecasting load at multiple nodes inside a campus grid consisting of a variety of loads including academic, residential, commercial, and mixed loads.
(iii) Application of a Moving average filter for data preprocessing as a novel contribution in this work to eliminate the outliers present in the dataset.
The remaining portions of the paper are structured as follows: Section 2 discusses the literature review followed by the proposed normalization method and different filtering techniques used for load prediction explained in Section 3. The forecast results and their comparisons using different filtering techniques are given in Section 4. Finally, the conclusion is presented in Section 5.

2. Literature review

Numerous studies have been done on the generation of electricity in a smart grid and the analysis of specific meteorological circumstances. The forecast approaches for solar energy and photovoltaic electricity have been reviewed in [8]. The authors of [9] performed forecast tactics for power distribution disruptions in a smart grid; this procedure is based on weather patterns and energy consumption, which shows that it rises with time. The authors of [10] used learning algorithms for smart energy meters to anticipate energy consumption. These studies demonstrate a strong correlation between the filtering techniques, forecasting of meteorological factors, and load forecasting for smart elec-

trical grids. Making an accurate prediction using sensors and transducer data is not an easy task. Due to the noise and measurement variations present in the signals obtained from these devices, a filtering approach may not be the best choice for a particular scheme or prediction technique [11]. Applying the different prediction algorithms to this noisy data would produce poor results since they would be unable to recognize the original trend in the data. As a result, data pre-processing is essential for improving the overall data quality [12]. The need and function of data pre-processing have been explained in many different ways. The required information can be represented along with fluctuations in the data brought on by variations in process or system variables, as well as in data collection and transmission. These impacts can be eliminated ahead with proper data pre-processing, leading to more efficient models. These models are considered to be more reliable, although they may not always be better predictors [13]. Filtering is another method for removing undesirable components or noise from a signal. Before the data is used to train a load forecaster, it can be filtered using a variety of filtering techniques [14].

Different techniques have been proposed by researchers for filtering and smoothening the raw dataset to minimize the effects of outliers on the input dataset [15] which includes a simple moving average [16, 17], Savitzky-Golay filter [18], LOWESS and LOESS filters [19], Autoregressive moving average [20], etc. In addition to these, ensemble-based data preprocessing is very popular these days and efforts are being made towards the amalgamation of two different preprocessing methods with the increased computational complexity [21]. The work proposed in this paper is motivated by the application of different filtering strategies to transform the raw data into usable data which enhances its performance many times. Here, various filtering techniques have been used and their performances have been compared for load forecasting at a node located inside a campus grid.

3. Methodology

In this section, the proposed forecasting methodology is presented as load forecasting. The detailed steps of the forecasting procedure are presented in Fig. 1. The load data is collected using smart meters installed at various nodes inside the campus. The raw data is then normalized to reduce wide variation in the range of load data over different seasons and weather variables. The normalized data is then pre-processed using the proposed eight different filters to remove the outliers without altering the original characteristics of the load. Load at any node is highly dependent on weather parameters like THI, time, and day, as the load on weekends is less as compared to the weekday loads. Therefore, these features are considered to be important attributes for training the forecasting model. The complete dataset is divided into two parts: training and testing datasets. Subsequently, the proposed NN-based load forecasting technique is used to predict the load at a given node inside the cam-

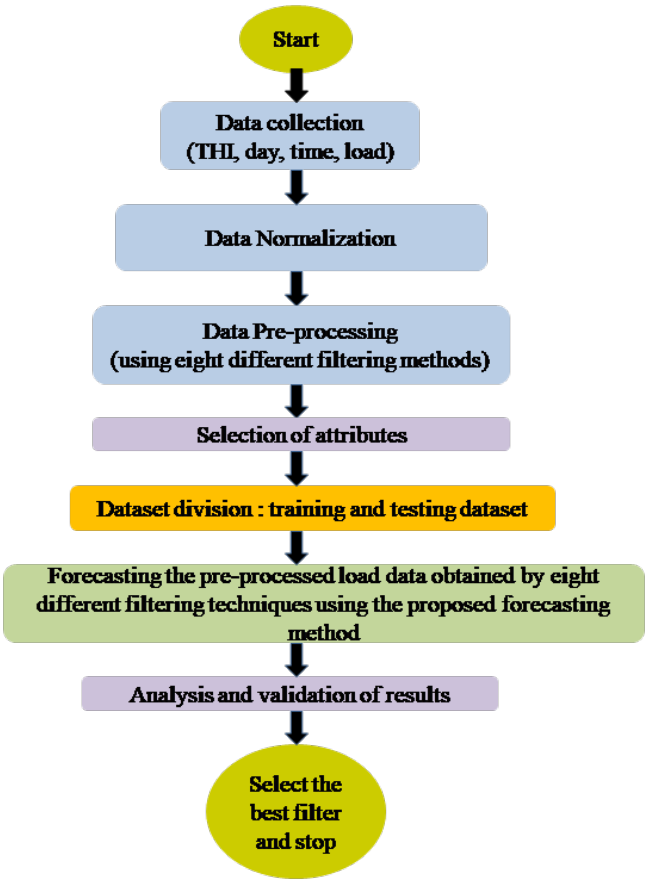

Figure 1. Pictorial representation of the forecasting methodology.

pus. Finally, the MAPE and RMSE values are calculated by comparing the forecasted load and the actual load from each filtering method to select the best pre-processing technique with minimum error values.

3.1 Data normalization

Normalization is important because of the significant seasonal variation in the range of load data brought on by changes in the weather, and model training loses this pattern of broad variation. The data is standardized on a scale of [-1, 1] to reduce the disparity. The z-score normalization technique [22] is used in this work for normalization, and its mathematical equation is presented in Eq. 1.

$$x_{new} = (x - \mu) \div \sigma \tag{1}$$

where; x represents the initial load at that time, μ is the mean of the data, σ is the standard deviation of the data, and x_{new} is the normalized load value at that instant. The z-score is used to determine how far a data point is from the mean. It calculates the standard deviations that are below or above the mean. The advantage of z-score normalization is that it enables a data administrator to understand the likelihood that a score will occur within the data's normal distribution. After normalization, the normalized data is filtered to eliminate any outliers from the dataset using various filters, which are covered in the section below.

3.2 Data filtration

Data filtration is a method of reducing noise in a dataset. The data can be filtered using a variety of filtering approaches before being used to train a load forecaster [13]. The different filters used in this work to preprocess the load dataset before using it to train the load forecaster are described below:

3.2.1 Moving Average Filter (MAF)

An MAF is a fundamental low-pass filter that combines m input samples into one output data set. It is a finite impulse response filter that has the same weighing effect throughout the selected window length of the filter. The output becomes smoother as the filter length increases because the strong transitions in the data are reduced while maintaining their original characteristics. The mathematical equation of the moving average filter is given in Eq. (2) below.

$$y_s(i) = \frac{1}{2N+1}(y(i+N) + y(i+N-1) + ... + y(i-N)) \tag{2}$$

Wwhere; $y_s(i)$ is the filtered load data value for an ith data point, N denotes the number of neighboring samples on either side $y_s(i)$, and $2N+1$ represents the data span.

3.2.2 Moving Median Filter (MMF)

The median filter is a non-linear digital filtering method that is frequently used to eliminate noise from a picture or signal. This noise removal is a common pre-processing measure to enhance the outcomes of subsequent processing. The fundamental principle of the median filter is to iteratively replace each entry in the signal with the median of its immediate neighbors. The "window" is the neighborhood pattern, which moves entry by entry over the entire signal. Due to its effective performance for some particular noise types, such as Gaussian, random, and salt and pepper sounds, the median filter is one of the well-known order-statistic filters [4].

3.2.3 Gaussian Filter (GF)

In electronics, a GF is a filter that has a Gaussian function as its impulse response. One advantage of a GF is that its Fourier transform has a Gaussian distribution with a zero frequency center (with positive and negative frequencies at both sides). The low pass functionality of the filter can be simply controlled by modifying the filter's width. Based on the Gaussian distribution, the Gaussian Smoothing Operator computes a weighted average of the surrounding pixels. It is mainly effective in reducing the Gaussian noise. Mathematically, a Gaussian filter alters the input signal via convolution with a Gaussian function, another name for this mathematical process is the Weierstrass transform. The impulse response of a one-dimensional Gaussian filter is denoted by Eq. (4) and the frequency response is expressed by Eq. (4) respectively.

$$g(x) = \sqrt{\frac{\alpha}{\beta}} e^{-\alpha x^2} \tag{3}$$

$$\bar{g}(f) = e^{\frac{-\pi^2 f^2}{\alpha}} \tag{4}$$

where; x represents the horizontal distance from the origin, α is the standard deviation, and f is the ordinary frequency.

3.2.4 LOWESS filter

Since both techniques use locally weighted linear regression to smooth the data, the names "LOWESS" and "LOESS" are derived from the phrase "locally weighted scatter plot smooth". The LOWESS filter smoothes the data by computing a linear regression in each window length. The regression weight function is specified by the toolbox for data points in the span, which causes the procedure to be weighted. Though computationally expensive, this method yields fewer discontinuities.

3.2.5 LOESS filter

This filter is similar to the 'LOWESS' filter, but the main difference is that it uses local quadratic regressions for smoothening the data in each window length. Since each smoothed value is derived by nearby data points identified within the span, similar to the moving average method, the smoothing process is regarded as local. It is mainly designed to tackle non-linear relationships which are not addressed well by the linear methods. One major drawback of using LOESS is that it does not provide a regression function that can be further represented by a mathematical equation or formula.

3.2.6 Robust LOWESS Filter (RLOWESS)

The averaged values may become deformed and no longer accurately reflect the behavior of the majority of the adjacent data points if the dataset contains any outliers. To solve this issue, the data can be smoothened by a robust

method that is unaffected by a tiny percentage of outliers. The RLOWESS smoothes the data using LOWESS but with the added advantage that it is more robust to outliers at the cost of computation.

3.2.7 Robust LOESS Filter (RLOESS)

Robust LOESS is also similar to LOESS as it smoothes the dataset using loess but it has a more robust nature for the outliers with the increased computational complexity. The robustness component of the Loess algorithm downweights observations with relatively large residuals to reduce the impact of "outliers" in the data.

3.2.8 Savitzky-Golay Filter (SGOLAY)

Savitzky-Golay filter is a type of generalized moving average filter where the filter coefficients are derived by performing unweighted linear least squares fit with the given polynomial. Because of this, an SGOLAY filter is also referred to as a digital smoothing polynomial filter or a least-squares smoothing filter. A polynomial with a higher degree yields a high level of smoothening without compromising the features of the data. The SGOLAY method of data filtering is typically employed mainly with spectroscopic or frequency data. The technique successfully preserves the high-frequency signal components for frequency data. For spectroscopic data, this method effectively preserves higher moments of the peak, like the line width. The centroid, on the other hand, can only be preserved by the MAF, which also removes a substantial amount of the high-frequency component of the signal. The SGOLAY filtering can, however, be less effective in eliminating noise than a moving average filter.

4. Case studies

4.1 Datasets description

This paper uses the load dataset of a smart metered distribution system set up at the residential cum academic campus of NIT Patna. The data includes the load consumption data of an academic load (library building) for the six months of August 2019 to January 2020 with a sampling period of one day [22]. All the various load data parameters, such as active power, reactive power, line and phase voltages, phase angle, frequency, power factor, current, etc., can be obtained at any node at any time, with a period ranging from a few minutes to many hours, depending on the requirement. The complete dataset is divided in the ratio of 7:1:2 in terms of training set, validation set, and testing set.

4.2 Error metrics

In this paper two error metrics are used to indicate the performance of the proposed load forecasting model and its comparison with the different pre-processing methods applied. The metrics are MAPE and RMSE which are mathematically expressed as:

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\bar{L_f}(i) - L_{act}(i)|}{L_{act}(i)} \tag{5}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\bar{L_f}(i) - L_{act}(i))^2} \tag{6}$$

where; N is the number of samples, $\bar{L_f}(i)$ is the forecasted load at hour i, and $L_{act}(i)$ is the actual load at that particular hour i.

4.3 Proposed load forecasting method

This paper uses a simple FFNN with a single input, output, and hidden layer. The ability of a neural network to handle nonlinear load patterns to create a multivariate model of meteorological variables is one of its key advantages. During training, the neural network is repeatedly exposed to varied input and output patterns, and weights between different layers are modified to determine the precise mapping between input and output. After training, the network is anticipated to learn the relationship between input and output behavior and, eventually, produce the required output for an unknown input pattern [22]. The number of neurons in the hidden layer is selected after the iterative evaluation for the least MAPE and RMSE values. Bayesian regularization algorithm is used to train the neural network.

4.4 Results and analysis

This section includes the load forecasting results obtained by using different filtering techniques on the smart metered data obtained from the campus test system. Also, it includes the comparative analysis of the diverse pre-processing techniques to select the best filtering technique which yields the minimum forecasting error by removing the maximum possible outliers present in the dataset without altering the original characteristics of the data. The input attributes used to train the developed forecasting model include weather parameters like temperature and humidity, day, and time index.

4.4.1 Forecasting results using different filtering techniques for data pre-processing

To remove errors and outliers from the data, the actual load data received from the smart meter is first normalized using the z-score normalization approach. The normalized data is then filtered using several filtering techniques. The proposed load forecasting model is then trained using the filtered data. The load forecasting performance obtained by using the moving average filter for data pre-processing with varying window sizes is given in Table 1. The number of data points a moving average filter uses for averaging is referred to as the window size. With a larger window size, the signal smoothening is more as compared to a small window size, but it may also compromise the filter's accuracy as it can be the case of loss of information since the original characteristics of the data are altered. Therefore, the window size is selected after the iterative evaluation to get the best forecasting results. It also depends upon the nature and type of the dataset used.

As it is known that the window size is a very important parameter that is responsible for the amount of smoothening in the signal. In this work, the widow size is varied between 2 to 20, and the consequent results are calculated as given in Table 1. In general, larger window sizes produce improved results but can add more computational costs and be less realistic as the original nature of the signal is hindered. Hence, the window size is selected based on the required

accuracy of the filter. Table 1 shows that forecasting results obtained by using a MAF for data pre-processing are very good and comparable for all the window sizes greater than 2, i.e. window length 5 can be proposed for this case as it fulfills both the requirements of the filter; high accuracy along with less alteration with the original signal. This can also be demonstrated by the comparison plot between the actual and predicted load shown in Fig. 2.

Table 2 shows the forecasting performance obtained by using GF for data pre-processing with the window size varied between 2 to 20. From the table, it is noted that the values of MAPE and RMSE for all the window lengths are slightly more as compared to the error values obtained in the case of MAF in Table 1 for the test dataset used. One of the major drawbacks of Gaussian models is that they are not well suited to categorical data since they function under the assumption that all the characteristics are regularly distributed. GF is a non-linear low pass filter mainly used for spatial data like blurring an image by removing the high spatial frequency components from an image. The graph obtained between the actual and predicted load for the test data is depicted in Fig. 3. As seen in the figure, the GF has a lower correlation between the actual and expected load than the MAF.

The result obtained by using the moving median filter for data smoothening before training the load forecasting model is given in Table 3. The comparison plot for the same is shown in Fig. 4. From the table, it can be concluded that the performance of the MMF is similar to that of the moving average filter for the applied dataset. One of the major differences between the moving average and moving median filter is that the MMF preserves the shifts in the level while the MAF blurs the sharp edges. The limitation of MMF is that such filters have the drawback of breaking up image edges and producing fake noise edges in the region with low signal-to-noise ratios, and they are unable to suppress Gaussian noise distributions.

Tables 4 and 5 express the forecasting performance obtained by using LOWESS and LOESS smoothening filters respectively. A nonparametric method for smoothing a set of data without making any assumptions about the data's underlying structure is known as locally estimated scatter plot smoothing, or LOESS. LOWESS is frequently used to fit a line to a scatter plot or time plot when it's difficult to see a line of best fit due to noisy data values, sparse data points, or weak interrelationships, or where least squares fitting fails to produce a line of good fit or is too time-consuming to use in linear regression. LOESS and LOWESS are developed on "traditional" methods, such as linear and nonlinear least squares regression. They deal with circumstances where the conventional methods don't work well or where their use would require excessive work.

From Tables 4 and 5, it is seen that the LOWESS and LOESS smoothening methods yield high MAPE and RMSE values as compared to the above-discussed smoothening methods like moving average, moving median, and Gaussian method for our load dataset. The reason behind this is LOESS uses data less effectively than other least squares techniques. The development of a good model needs large, highly sampled datasets (big data) to be generated. The comparison plots between actual and predicted load using LOWESS and LOESS smoothening methods are shown in Fig. 5 and 6 respectively.

LOWESS is used for univariate smoothening i.e. for fitting a smooth curve to a scatter plot while Loess is used for multivariate smoothening. Both algorithms work on locally weighted polynomial regression. The toolbox specifies a regression weight function for the data points included in the span, which causes the technique to be weighted. The resistant option additionally includes a weight function that

Table 1. Forecasting performance using MAF.

Window Size	MAPE	RMSE
2	10.52	0.0.627
5	0.0443	4.04E-04
10	0.0429	3.34E-04
20	0.044	3.28E-04

Table 2. Forecasting performance using GF.

Window Size	MAPE	RMSE
2	15.2614	1.2798
5	3.256	0.2587
10	1.76	0.0124
20	0.2112	6.85E-04

Table 3. Forecasting performance using MMF.

Window Size	MAPE	RMSE
2	10.665	0.6674
5	0.0502	4.57E-04
10	0.0416	4.05E-04
20	0.0444	4.32E-04

Table 4. Forecasting performance using LOWESS Filter.

Window Size	MAPE	RMSE
2	13.538	1.3521
5	14.011	1.3632
10	13.938	1.3587
20	12.3263	1.3934

Table 5. Forecasting performance using LOESS Filter.

Window Size	MAPE	RMSE
2	13.452	1.3229
5	12.412	1.3413
10	13.29	1.3542
20	14.237	1.3383

Table 6. Forecasting performance using LOWESS Filter.

Window Size	MAPE	RMSE
2	13.16	1.3638
5	13.131	1.353
10	13.644	1.3627
20	13.709	1.3505

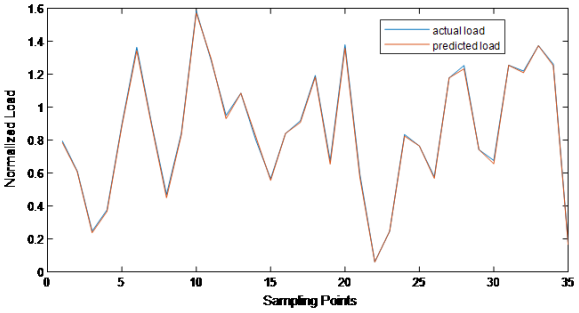

Figure 2. Comparison plot for MAF.

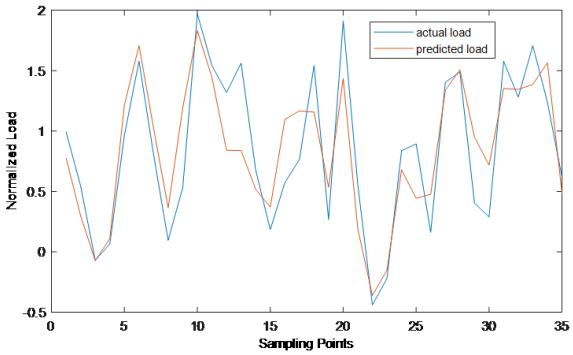

Figure 3. Comparison plot for GF.

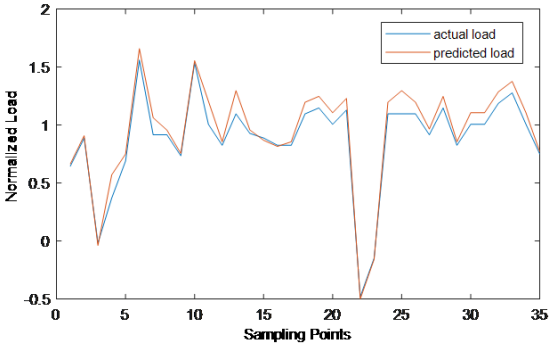

Figure 4. Comparison plot for MMF.

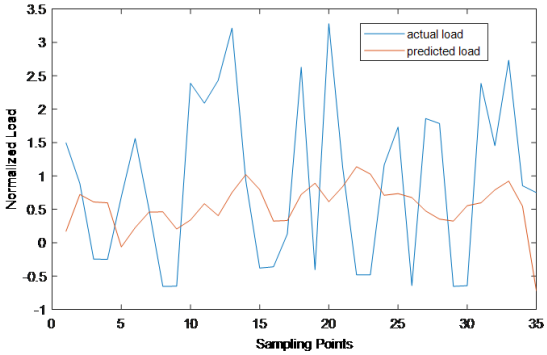

Figure 5. Comparison plot for LOWESS Filter.

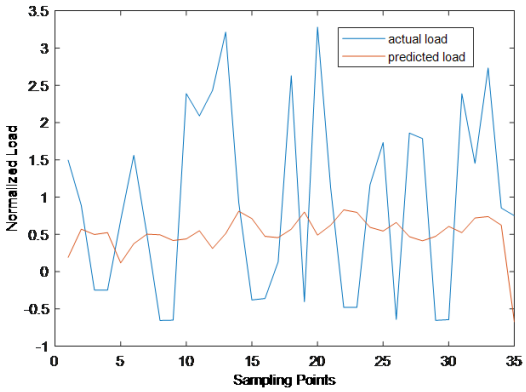

Figure 6. Comparison plot for LOESS Filter.

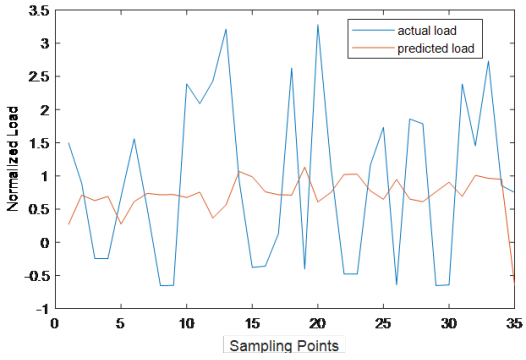

Figure 7. Comparison plot for RLOWESS Filter.

can make the process resistant to outliers in addition to the regression weight function. The robustness component of the LOESS algorithm downweights observations with relatively large residuals to lessen the effect of "outliers" in the data. The forecasting results obtained by RLOWESS and RLOESS smoothening techniques are shown in Tables 6 and 7 respectively along with the comparison plots depicted in Fig. 6 and 7.

The load forecasting result by using the SGOLAY filter for pre-processing is given in Table 8 and the comparison plot of actual and forecasted load is shown in Fig. 9. The SGOLAY filters simply work by fitting a polynomial to each sample in the filtered sequence's direct neighborhood of N neighbors. Simply evaluate the polynomial at point 0, which serves as both the polynomial's center and the neighborhood's center and move on to the following neighborhood. This filter has extensive application in the field of biomedical signal processing. The basic principle behind this filter is to find a $2n+1$ equidistant point for representing a polynomial of degree p.

4.4.2 Performance comparison of different filtering techniques for data pre-processing

The load forecasting performance comparison by the aforementioned filters for pre-processing the data is given in Table 9. The obtained results show that MAPE and RMSE

Table 7. Forecasting performance using LOWESS Filter.

Window Size	MAPE	RMSE
2	13.093	1.3601
5	13.941	1.3612
10	13.644	1.3627
20	14.014	1.3453

Table 8. Forecasting performance using SGOLAY Filter.

Window Size	MAPE	RMSE
2	13.761	1.3637
5	12.362	1.3697
10	13.063	1.3437
20	13.852	1.3591

Table 9. Forecasting results using different filters for window size 5.

S.NO	Filter	MAPE	RMSE
1	MAF	0.0443	4.04E-04
2	GF	3.256	0.2587
3	MMF	0.0502	4.57E-04
4	LOWESS	14.011	1.3632
5	LOESS	12.412	1.3413
6	RLOWESS	13.131	1.353
7	RLOESS	13.941	1.3612
8	SGOLAY	12.362	1.3697

values are the least for MAF compared to the other filtering methods used. For the window length variation between 2 to 20, the MAPE varies in the range of 10% to 0.05%. By using the Gaussian smoothening filter, the MAPE lies from 15% to 0.2%, which is more compared to the moving average filter. However, for our test dataset, the MMF performs similarly to the MAF with very little increment in MAPE obtained by using an MAF as given in Table 3. Now, it can be noted from the results obtained in Table 9 that the LOWESS and LOESS filters have poor load forecasting performance with the MAPE variation in the range of 13% to 12% for our test dataset as compared to the moving average, Gaussian and moving median filters. The reason behind this can be the size of the dataset used, as these methods need a large amount of sampled data for building a good model, which is not available in our case. The same performance is seen for RLOWESS, RLOESS, and SGOLAY filters also as tabulated in Table 9. T+he SGOLAY filter is mainly comfortable for the frequency data rather than the time series data. Many studies show that the SGOLAY filter is less successful than the moving average filter in the rejection of noisy components. The box plot shown in Fig. 10 can also be used to demonstrate how different filtering techniques for data pretreatment result in different comparative MAPE values.

The boxes drawn in this plot display the MAPE range that has been calculated using different filtering techniques by the proposed load forecasting algorithm. Each box contains a red line that represents the median of the range of MAPE values. The box plot shows that MAPE for the MAF and

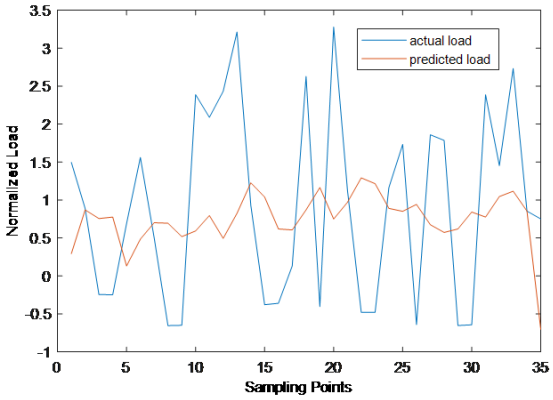

Figure 8. Comparison plot for RLOESS Filter.

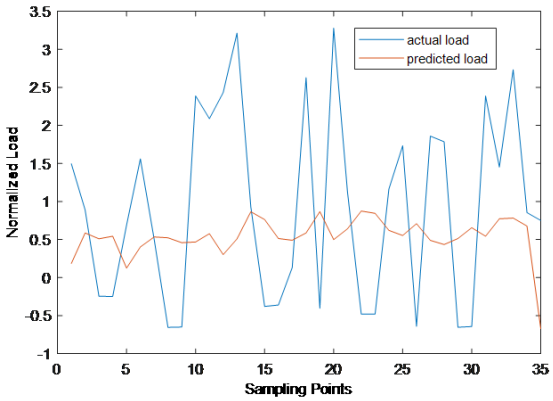

Figure 9. Comparison plot for SGOLAY Filter.

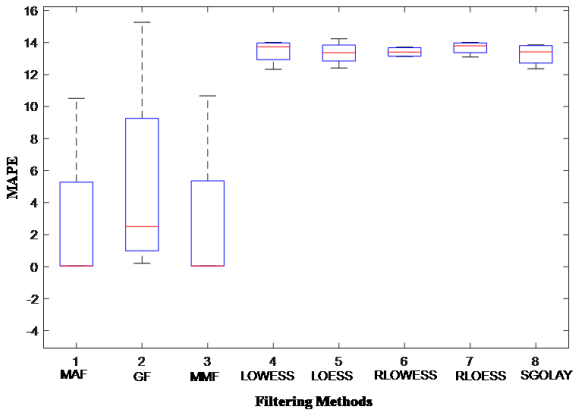

Figure 10. Box plot depicting MAPE values using different filtering techniques.

MMF filters is lower and within a similar range of 0.05 to 10.52%. By using the MAF filter for pre-processing, the least MAPE is obtained for all the window sizes varied from 2 to 20 as compared to the other filters used, which concludes that the MAF filter has the best performance for the test case used.

5. Conclusion

In a smart grid framework, this study demonstrates the effects of several filtering strategies in a load forecasting scenario with real-time data obtained from a smart meter. For normalization purposes, the z-score normalization technique is used, and the normalized data is again filtered using different methods to obtain the processed data which can be fed to the load forecasting model. The comparison of different filtering strategies for the application of load forecasting using a neural network model is presented here. From the load forecasting results obtained by using different filtering techniques, it was found that the moving average filter yields better performance as compared to the other filters proposed in this work with the average MAPE of 2.66% and RMSE of 0.157 for the test dataset used. The filtering and prediction performance, however, varies depending on the dataset's fluctuation, therefore this should also be taken into consideration, i.e., it highly depends upon the size, sampling interval, nature, and trend of the dataset used for testing. Therefore, the filter which is giving best performance for this case doesn't need to give equally good results for the variation in the dataset.

Acknowledgement:
This work is supported by the Science & Engineering Research Board, Department of Science and Technology, Government of India, under grant number ECR/2017/001027.

Authors contributions
All authors have contributed equally to prepare the paper.

Availability of data and materials
The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflict of interests
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Ahmad, X. Xiao, H. Mo, and D. Dong. "**Tuning data preprocessing techniques for improved wind speed prediction**.". *Energy Reports*, 11:pp. 287–303, 2024. DOI: https://doi.org/10.1016/j.egyr.2023.11.056.

[2] A. Parashar, A. Parashar, W. Ding, M. Shabaz, and I. Rida. "**Data preprocessing and feature selection techniques in gait recognition: A comparative study of machine learning and deep learning approaches**.". *Pattern Recognition Letters*, 172:pp. 65–73, 2023. DOI: https://doi.org/10.1016/j.patrec.2023.05.021.

[3] B. Boashash and Ed. "**Chapter 11 - Time-Frequency Synthesis and Filtering**.". *in Time-Frequency Signal Analysis and Processing (Second Edition), Oxford: Academic Press*, pages pp. 637–691, 2016. DOI: https://doi.org/10.1016/B978-0-12-398499-9.00011-X.

[4] S. Rai and M. De. "**Effect of Filtering in Big Data Analytics for Load Forecasting in Smart Grid in Machine Learning, Image Processing, Network Security and Data Sciences, A. Bhattacharjee, S. Kr. Borgohain, B. Soni, G. Verma, and X.-Z. Gao, Eds., in Communications in Computer and Information Science.**". *Singapore: Springer*, pages pp. 125–134, 2020. DOI: https://doi.org/10.1007/978-981-15-6315-7-10.

[5] M. Aouad, H. Hajj, K. Shaban, R. A. Jabr, and W. El-Hajj. "**A CNN-Sequence-to-Sequence network with attention for residential short-term load forecasting**.". *Electric Power Systems Research*, 211:p. 108152, 2022. DOI: https://doi.org/10.1016/j.epsr.2022.108152.

[6] V. Chinta, G. Song, and W. Zhang. "**Validation of the medium-range and sub-seasonal forecast of solar irradiance and wind speed using ECMWF**.". *Energy Reports*, 10:pp. 3908–3913, 2023. DOI: https://doi.org/10.1016/j.egyr.2023.10.058.

[7] A. S. F. Rocha, F. K. de O. M. V. Guerra, and M. R. B. G. Vale. "**Forecasting the Performance of a Photovoltaic Solar System Installed in other Locations using Artificial Neural Networks**.". *Electric Power Components and Systems*, 48(1-2):pp. 201–212, 2020. DOI: https://doi.org/10.1080/15325008.2020.1736211.

[8] J. W. Taylor and R. Buizza. "**Neural network load forecasting with weather ensemble predictions**.". *IEEE Transactions on Power Systems*, 17(3):pp. 626–632, 2002. DOI: https://doi.org/10.1109/TPWRS.2002.800906.

[9] S. Chapaloglou and et al. "**Smart energy management algorithm for load smoothing and peak shaving based on load forecasting of an island's power system**. ". *Applied Energy*, 238:pp. 627–642, 2019. DOI: https://doi.org/10.1016/j.apenergy.2019.01.102.

[10] P. Mishra, A. Biancolillo, J. M. Roger, F. Marini, and D. N. Rutledge. "**New data preprocessing trends based on ensemble of multiple preprocessing techniques**.". *TrAC Trends in Analytical Chemistry*, 132:p. 116045, 2020. DOI: https://doi.org/10.1016/j.trac.2020.116045.

[11] E. Escobar-Avalos, M. A. Rodríguez-Licea, H. Rostro-González, A. G. Soriano-Sánchez, and F. J. Pérez-Pinal. "**A Comparison of Integrated Filtering and Prediction Methods for Smart Grids**.". *Energies*, 14 (7), 2021. DOI: https://doi.org/10.3390/en14071980.

[12] E. R. Davies. "**CHAPTER 3 - Basic Image Filtering Operations**.". *in Machine Vision (Third Edition), E. R. Davies, Ed., in Signal Processing and its Applications. Burlington: Morgan Kaufmann*, pages pp. 47–101, 2005. DOI: https://doi.org/10.1016/B978-012206093-9/50006-X.

[13] E. Hussein. "**Preprocessing of Measurements**.". pages pp. 97–123, 2011. DOI: https://doi.org/10.1016/B978-0-12-387777-2.00009-4.

[14] C. Becker and U. Gather. "**The Masking Breakdown Point of Multivariate Outlier Identification Rules**.". 1997. DOI: https://doi.org/10.17877/DE290R-15061.

[15] D. J. Robb and E. A. Silver. "**Using Composite Moving Averages to Forecast Sales**.". *The Journal of the Operational Research Society*, 53(11):pp. 1281–1285, 2002.

[16] "**Introduction to Modern Time Series Analysis**.". *SpringerLink*, 2023. URL https://link.springer.com/book/10.1007/978-3-540-73291-4.

[17] J. Luo, K. Ying, and J. Bai. "**Savitzky-Golay smoothing and differentiation filter for even numberdata**.". *Signal Process*, 85(7):pp. 1429–1434, 2005. DOI: https://doi.org/10.1016/j.sigpro.2005.02.002.

[18] W. S. Cleveland and S. J. Devlin. "**Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting**.". *Journal of the American Statistical Association*, 83(403):pp. 596–610, 1988. DOI: https://doi.org/10.1080/01621459.1988.10478639.

[19] G. E. P. Box. "**Time Series Analysis**.". *5th edition*, 2015.

[20] Z. Qu, J. Xu, Z. Wang, R. Chi, and H. Liu. "**Prediction of electricity generation from a combined cycle power plant based on a stacking ensemble and its hyper parameter optimization with a grid-search method**.". *Energy*, 227:p. 120309, 2021. DOI: https://doi.org/10.1016/j.energy.2021.120309.

[21] S. Raschka. "**About Feature Scaling and Normalization**.". 2014. URL https://sebastianraschka.com/Articles/2014_about_feature_scaling.html.

[22] S. Rai and M. De. "**Analysis of classical and machine learning based short-term and mid-term load forecasting for smart grid**.". *International Journal of Sustainable Energy*, 40(9):pp. 821–839, 2021. DOI: https://doi.org/10.1080/14786451.2021.1873339.