Journal of Industrial and Systems Engineering
Vol. 15, No. 1, pp. 54-68
Winter (January) 2023

A novel two-level clustering algorithm for time series group forecasting

Ali Ghorbanian[1], Hamideh Razavi[1]*

[1]*Department of Industrial Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran*

ali.ghorbanian@mail.um.ac.ir, h-razavi@um.ac.ir

Abstract

Parametric models are considered the widespread methods for time series forecasting. Non-parametric or machine learning methods have significantly replaced statistical methods in recent years. In this study, we develop a novel two-level clustering algorithm to forecast short-length time series datasets using a multi-step approach, including clustering, sliding window, and MLP neural network. In first-level clustering, the time series dataset in the training part is clustered. Then, we made a long time series by concatenating the existing time series in each cluster in the first level. After that, using a sliding window, every long-time series created in the previous step is restructured to equal-size sub-series and clustered in the second level. Applying an MLP network, a model has been fitted to final clusters. Finally, the test data distance is calculated with the center of the final cluster, selecting the nearest distance, and using the fitted model in that cluster, the final forecasting is done. Using the WAPE index, we compare the one-level clustering algorithm in the literature regarding the mean of answers and the best answer in a ten-time run. The results reveal that the algorithm could increase the WAPE index value in terms of the mean and the best solution by 8.78% and 5.24%, respectively. Also, comparing the standard deviation of different runs shows that the proposed algorithm could be further stabilized with a 3.24 decline in this index. This novel study proposed a two-level clustering for forecasting short-length time series datasets, improving the accuracy and stability of time series forecasting.

Keywords: Time series, clustering, forecasting, sliding window, neural network

1-Introduction

The forecasting for time series data is based on the point that previous data have intricate patterns that can be used for forecasting future horizons (Boshnakov, 2016). Parmazan et al. performed a general classification and split time series forecasting methods into two main parts, parametric and non-parametric (Parmezan, Souza, & Batista, 2019). Parametric methods consist of statistical methods such as moving average (MA) (Lucas & Saccucci, 1990), ARIMA (Box, Jenkins, Reinsel, & Ljung, 2015), and smoothing methods (Box et al., 2015). In the traditional statistical methods to predict time series, knowing the statistical distribution of the data is required. In these methods, regulating the model parameters is complicated (Islam & Sivakumar, 2002), so non-parametric or machine learning techniques could replace statistical methods in recent years. The machine learning methods for forecasting time series include MLP neural networks (Borghi, Zakordonets, & Teixeira, 2021), LSTM (Abbasimehr & Paki, 2022), recurrent neural networks (Weerakody, Wong, Wang, & Ela, 2021), Combining multiplex networks (Pérez, Moral-Rubio, & Criado, 2023) and SVM (Pant & Kumar, 2022) and KNN (Parmezan & Batista, 2015; B. Yu, Song, Guan, Yang, & Yao, 2016) algorithms.

*Corresponding author
ISSN: 1735-8272,

Mixing parametric and non-parametric methods is an approach researchers use to improve the forecasting precision of time series (de Araújo Morais & da Silva Gomes, 2022; Hajirahimi & Khashei, 2022).

Parmazan et al., in a comprehensive study, demonstrated that artificial neural network models were used from 2009 to 2018 in 62% of the research on time series forecasting (Parmezan et al., 2019).

Our motivation for conducting this research is to address the existing research gap in the literature by presenting a novel time series group forecasting algorithm for short-length time series datasets, applying sliding window and neural network. Unlike previous studies, this algorithm uses two clustering levels to determine the final clusters for fitting a machine-learning model. The purpose is to improve the precision of the clusters created and, consequently, the learning precision of these clusters. The algorithm's results reveal improvement in the accuracy and stability of time series forecasting.

The remainder of this paper is organized as follows: section 1 presents the introduction and literature reviews of the topic. Section 2 describes the proposed algorithm and its steps in detail. Section 3 provides test results and summarizes the performance of the two-level algorithm based on different performance indices. Research conclusions were presented in section 4, and section 5 suggested future studies.

1-1-Related works

Mixing clustering and classification methods is one way to improve classification and forecasting problems' efficiency. Several studies in various fields, like recommendation systems (Koosha, Ghorbani, & Nikfetrat, 2022), health field forecasting (Al-Hiary, Bani-Ahmad, Reyalat, Braik, & Alrahamneh, 2011; Polat, 2012; Said, Abd-Elmegid, Kholeif, & Gaber, 2018; Udler et al., 2018), passenger flow prediction(P. Li, Wu, & Pei, 2023), and image processing (Mai, Ngo, & Trinh, 2018; C. Yu, Wang, Zhao, Hao, & Shen, 2020) have revealed that this approach may improve forecasting efficiency in various issues. This approach was applied to forecast time series in the past decade. This approach has been used in some studies to forecast a time series. For instance, Sfetsos and Siriplus combined neural networks and clustering algorithms to forecast the daily variations in the pound and dollar value (Sfetsos & Siriopoulos, 2004). Kadia et al. applied time series forecasting based on clustering to forecast the number of references to papers (Kedia, Thummala, & Karlapalem, 2005). Hu et al. have used a novel method for forecasting time series based on directed visibility graphs and improved random walk (Hu & Xiao, 2022). Arias and Bayi used the clustering of traffic distribution variables and weather conditions to forecast the electric charge of cars in South Korea (Arias & Bae, 2016). So, the time series length must be long enough so that the required number of sub-series can be used to cluster and learn neural networks in every cluster by segmenting it. Buck employed a fuzzy approach to cluster and forecast energy demand (Bock, 2018). Panapakidis developed four models based on clustering and forecasting neural networks to forecast the load of 10 buses (Panapakidis, 2016).

Dong et al. combined time series clustering and convolutional neural networks to forecast electric current. However, in the so-called approach, the data were first clustered and then split into training and test data, which is more or less different from the previous methods (Dong, Qian, & Huang, 2017). To solve this problem, researchers concurrently used a group of time series for clustering. For instance, Stewka et al. used the K-means clustering algorithm for group forecasting of time series (Astakhova, Demidova, & Nikulchev, 2015). Another research reported that selling three types of computer products was forecasted via mixing time series forecasting using neural networks and clustering. This study combined three clustering algorithms of K-means, SOM, GHSOM, and two neural network algorithms of SVR and ELM (Chen & Lu, 2017).

Li et al. applied a self-made clustering approach for time series group forecasting. In this method, they are selected in order of time series. If the time series is sufficiently close to a cluster, it is added to that cluster; otherwise, another cluster is developed (Lee, Su, Lin, & Lee, 2017). Joe et al. applied clustering forecasting to forecast medicine demand for different hospitals. In this research, demands from various hospitals in the city were first collected, then this dataset was clustered, and forecasting was made for every cluster (Xu, Chan, Ch'ng, & Tan, 2020). Li et al. applied this model to forecast the demands at various common bicycle stations (D. Li, Zhao, & Li, 2019).

According to the related literature, to forecast a time series group, these series became a long time series. Then the long-time series was segmented to develop a sub-series with constant length.

Afterward, the sub-series were clustered, and a statistical model or machine learning was assumed for every cluster. If the used time series group owns different structures, the clusters for fitting a statistical model or machine learning may not be sufficiently accurate. So, a model based on two-level clustering was developed to solve this problem. In the proposed algorithm, the time series group was first clustered; in the second stage, clustering was done for each cluster developed in the first stage. Hence, it is expected to witness more correlation among the within-created clusters, so the fitted model to each cluster becomes more precise. The equipped models may provide more efficient forecasting for each time series.

2-Two-level clustering algorithm

According to the literature review, clustering is a commonly used method for time series forecasting. In these methods, often applied for long time series, a long time series is first made by setting a time series dataset together. Then, the time series is divided into identical parts by a moving window with a fixed size. In the next stage, the sub-series are clustered, and a statistical model or machine learning is fitted for the clusters. Finally, the most suitable cluster is chosen using the test data distance from the clusters, and the final forecasting is done.

However, the dataset may have different structures in most real cases, reducing the sub-series clustering precision. So, fitting a proper model for every cluster will cause problems. Based on the two issues discussed, forecasting efficiency can also decline. Accordingly, a two-level clustering algorithm was developed to forecast short-length time series datasets. In this research, time series clustering is conducted at two levels. At the first level, the time series are clustered in one dataset; at the second level, each cluster's series are clustered again separately for the final forecast. The proposed algorithm has seven stages:

Stage 1 (data split): the time series dataset is split into two parts: train and test. So, 80% of the data's first part is considered a train, and 20% of the second part as a test.

Stage 2 (first-level clustering): the train time series dataset is clustered by a clustering algorithm.

Stage 3 (creating long time series): For each created cluster in stage 2, we concatenated the time series in each cluster and made a long time series.

Stage 4 (restructuring sub-series): Using a fixed-length moving window, the long time series for every cluster in stage 3 are restructured to equal-size sub-series.

Stage 5 (second-level clustering): The sub-series in the previous stage are clustered for every cluster in the first level.

Stage 6 (model fitting): A machine learning model is fitted for every cluster created in the second level.

Stage 7 (forecast): The most proper cluster is chosen based on the test data distance from the final clusters. Then, forecasting is done by applying the fitted model for the cluster and the test data.

Figure 1 accurately depicts the mentioned stages. In the following, the primary steps of the plan and the specific algorithms used for every stage are elaborated in detail.

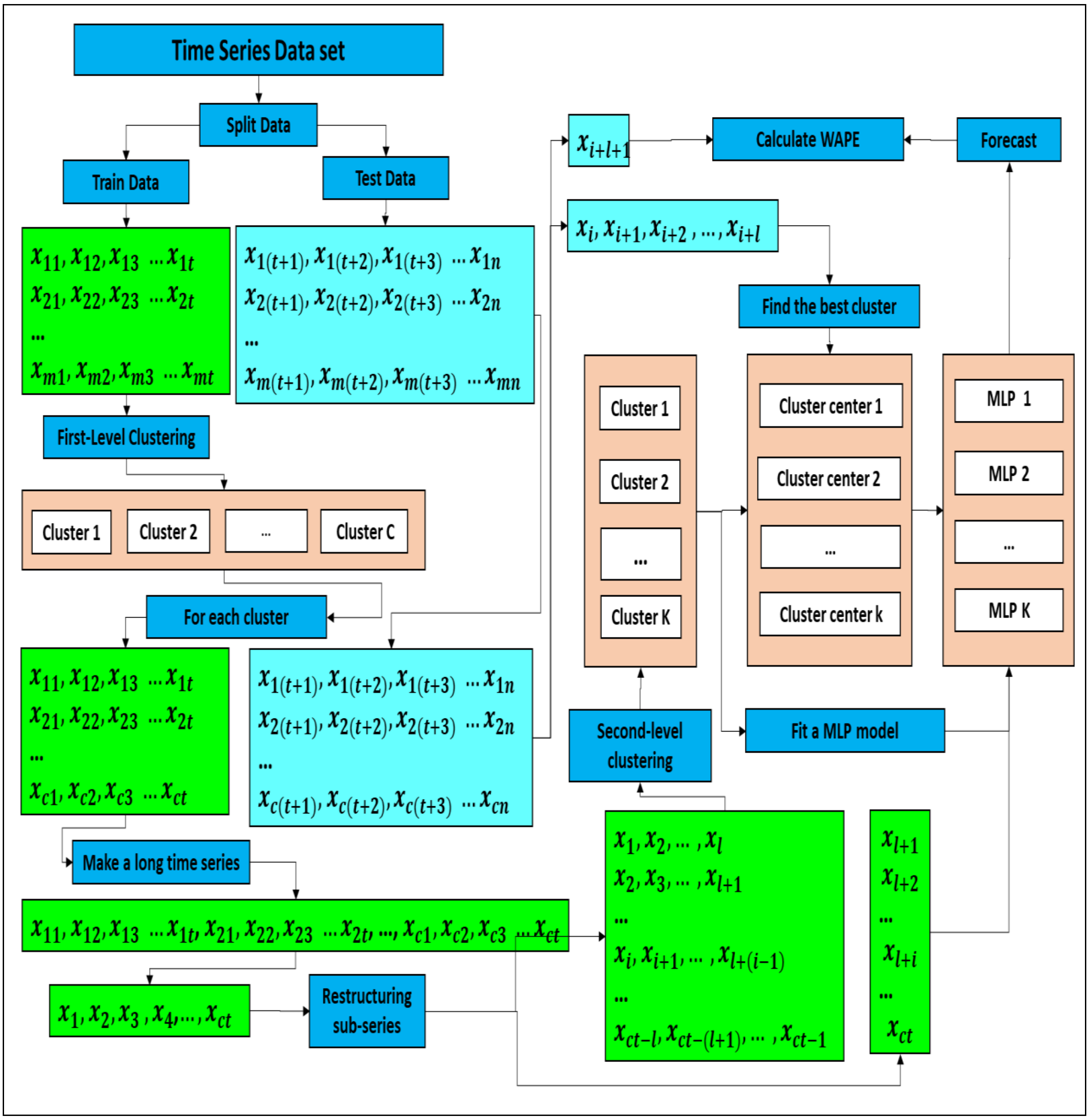


Fig1. The proposed plan of the two-level clustering algorithm

2-1-First and second-level clustering

The proposed algorithm has two clustering levels. At the first level, the dataset is split into a determined number of clusters. In the second clustering level, every cluster developed in the first level is clustered again. The hierarchical algorithm was applied for the first-level clustering, and the K-means algorithm was used for the second-level clustering.

2-1-1-Hierarchical algorithm

The algorithm used for the first-level clustering is agglomerative hierarchical with the complete distance ($d_{\max}$). Equation (1) shows the complete distance. A and B are the clusters; a and b are the clusters' objects, and d is the distance between two objects.

$$d_{\max} = \max\{d(a,b) : a \in A, b \in B\} \tag{1}$$

2-1-2-K-means algorithm

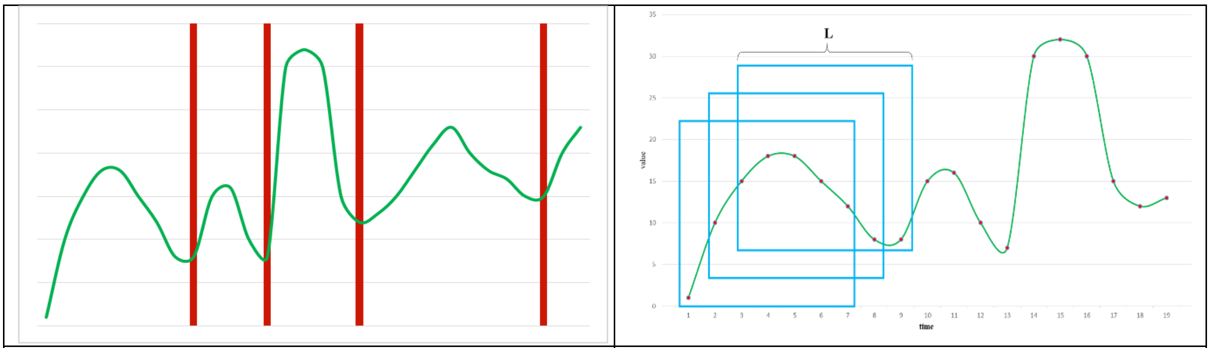
The k-means algorithm is a prevalent clustering algorithm (Huang et al., 2016; Kobylin & Lyashenko, 2020). It uses an iterative approach to decrease the data distance within a group and raise the distance between groups. Equation (2) shows the objective minimization function of this algorithm. In this relation, C is the center of each cluster; x and y are the objects; w is a 0 - 1 variable, which assesses every point within a cluster. In the proposed algorithm, this algorithm is used in the first and second clustering levels with Euclidean distance. Equation (3) shows the calculation procedure of the Euclidean distance in two n-dimensional spaces (Sinaga & Yang, 2020).

$$j = \sum_{i=1}^{m} \sum_{k=1}^{k} w_{ik} \left| x^{i} - C_{k} \right|^{2} \tag{1}$$

$$ED(X, Y) = \sqrt{\sum_{t=1}^{n} (x_t - y_t)^2} \tag{2}$$

2-2-Restructuring sub-series

A time series restructuring to sub-series is performed with two methods of special algorithms and sliding window (Keogh, Chu, Hart, & Pazzani, 2004; Norwawi, 2021). Figure 2a depicts restructuring by unique algorithms, and figure 2b shows restructuring by a sliding window.



a- Restructuring by special algorithms | **b-** Restructuring by a sliding window

Fig 2. Various types of time series restructuring to sub-series

This study uses the second method to restructure the sub-series. For clustering in the second level, at first, the series in every first-level cluster are set next to each other to make a long-time series .Then, a fixed-length L-moving window is applied for restructuring this series. Assuming that the time series is shown as $x_1,\ x_2, x_3\ , x_4,\ \ldots, x_{ct}$, table 1 displays the sub-series created by this method .The first sub-series is made by choosing the first L data of the main time series. Then, the first data is omitted from the time series set, and the second sub-series is the same as the prior form. This loop continues until L data to the end of the time series. If the time series length equals ct, $ct - L$ sub-series will be made. Moreover, the following data is applied as a response variable for training and model fitting in every data sub-series.

Table1. Time series segmentation using a fixed-size moving window

Sub-series	Response variable
$x_1, x_2, \ldots, x_l$	x_{l+1}
$x_2, x_3, \ldots, x_{l+1}$	x_{l+2}
…	…
$x_i, x_{i+1}, \ldots, x_{l+(i-1)}$	x_{l+i}
…	…
$x_{ct-l}, x_{ct-(l+1)}, \ldots, x_{ct-1}$	x_{ct}

MLP neural network is a popular and efficient model for forecasting time series (Borghi et al., 2021; Talkhi, Fatemi, Ataei, & Nooghabi, 2021). It has three main components, including the input, hidden, and output layers, in which the hidden layer can be made of one layer or several layers. This network has some problems, such as interpretability. But it is still a prevalent neural network due to its application for big data size and generalizability. A neural network's precision is indicated by three factors: the hidden layers' number, the neurons' number in each layer, and each layer's weight (de Jesús Rubio, 2017). An MLP neural network with one hidden layer is employed in the proposed plan. Also, the neurons' number in this hidden layer equals the input variable's size. Since the forecast input variable of each time series is identical to the segments' length, the neurons' number in the hidden layer is equal to L. Figure 3 depicts the network used in the proposed plan.

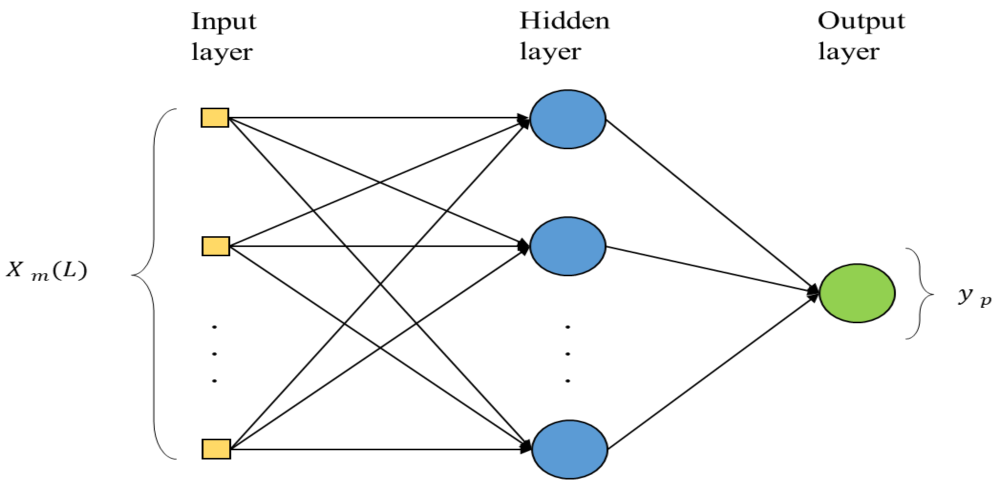


Fig 3. MLP network with one hidden layer used in the design

2-3-Forecast

L data posterior to the response data are first selected using the sliding window to forecast each test data. The most suitable cluster is picked when comparing the distance between this selected sub-series and the clusters' centers in the second stage. The final forecast is performed for the test data by applying the neural network model fitted for the selected cluster and the selected sub-series. Figure 4 shows the process of selecting the input data and the response variable using sliding windows with length L in detail. Also, the procedure of applying this data in a trained neural network for final forecasting and error calculation can be seen in this figure.

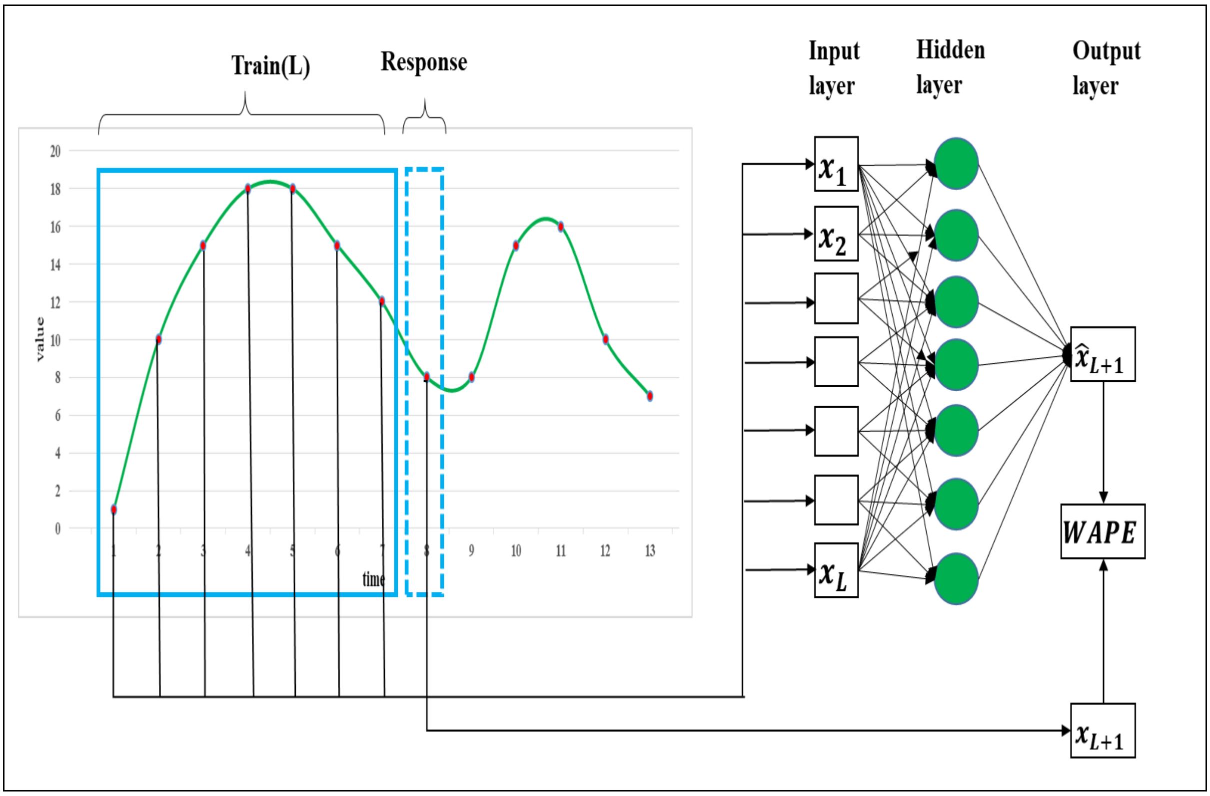


Fig 4. Final forecasting using MLP neural network

3-Test results

Different time series datasets can create a different number of clusters. Besides, the proper input variable size for learning a neural network to forecast can vary for different datasets. In the proposed algorithm, we considered three parameters, which are: the number of first-level clusters (K_1) with values of 2,3,4,5; the number of second-level clusters (K_2), with values of 2 and 3; the segments' length (L), with values of 15 and 20. By this, we mean that we assumed 16 permutations when running the presented algorithm for each dataset.

Table 2. Parameters used in the model and their values

Parameters	**Symbol**	**value**
Number of clusters in the first level	K_1	2-5
Number of clusters in the second level	K_2	2,3
Size of moving window	L	15,20
The number of neurons in the hidden layer of the MLP network	L	15,20

Figure 5 displays the pseudocode used for the proposed algorithm and three mentioned parameters with different values while evaluating the algorithm.

Input: Time series datasets
Output: Best WAPE
For each k_1 **do**
    For each L **do**
        For each k_2 **do**
            run a two-level clustering algorithm
            Calculate WAPE
Select best WAPE
	Input: Time series dataset,k_1,k_2, L
	Output: WAPE
	Split dataset to train and test
	For train data
	First level clustering (number of cluster=k_1)
	    For each first-level cluster **do**
	        Concatenate time series in cluster
	        Restructuring sub-series by sliding window (L)
	        Second-level clustering (number of cluster=k_2)
	            For each second-level clustering, do
	                MLP model
	                Calculate the center of cluster
	For test data
	Find the nearest cluster in the second-level clustering
	Forecast
	Calculate WAPE

b- Pseudo-code for 16 permutations	**a-** Pseudo-code of two-level clustering algorithm

Fig 5. Pseudocode of the proposed plan

3-1-Results

The efficiency of the proposed algorithm was assessed in terms of forecast efficiency and results consistency. The WAPE index was used to evaluate the proposed algorithm's forecast efficiency. Equation (4) shows this index, where A is the actual value, and F is the forecast value. The standard deviations of different runs were assessed to evaluate the proposed algorithm's stability.

$$WAPE = \frac{\sum_{i=1}^{n} |A_i - F_i|}{\sum_{i=1}^{n} |A_i|} \quad (4)$$

To assess its effectiveness, this model was applied to 35 datasets from the UCR archive, each with a length of less than 600. These datasets cover various topics, including health, finance, media, and engineering (Dau et al., 2019).

The proposed two-level clustering algorithm is implemented in two settings. For first-level clustering, we applied the hierarchical and k-means algorithms in the first and second settings. This aims to investigate the effect of different clustering algorithms on the final accuracy. First, the better setting will be selected based on the three criteria of the average WAPE, the best WAPE, and the standard deviation between these two settings. Then, in the following sections, the best-selected setting will be thoroughly examined and put up against the one-level clustering algorithm. Each setting of algorithm has been run on each dataset 10 times The table 3 depicts the performance results of both algorithm settings in the summary. According to the information in the table, it can be seen that the WAPE index of the Average of 10 runs for the first setting of the algorithm is equal to 27.5%, while this index for the second setting is 28.97%. The optimum value obtained for the WAPE index in a 10-time run is 24.66% for the first setting, which 1.61% is better than the second setting. Also, the standard deviation criterion of ten runs for both settings of the algorithm is close to each other, but despite this, the first setting of the algorithm has a lower value. In general, only the first setting of the suggested algorithm has been looked at and contrasted with the one-level clustering algorithm because it produces better results in the three key criteria introduced than the second setting. According to the obtained results, it can be concluded that the use of the hierarchical clustering algorithm for first-level clustering can bring better results.

Table 3. Comparison of the first and second settings of the proposed algorithm

	Two level clustering (Hierarchical)		Two level clustering (K-mean)	
	Average of 10 run	Best of 10 run	Average of 10 run	Best of 10 run
WAPE	27.50	24.66	28.97	26.27
Standard deviation	2.00	-	2.12	-

In the first state, The average WAPE for the first setting of the two-level clustering algorithm is compared with the one-level clustering algorithm. Table 5 and figure 6 show that the mean WAPE for 35 datasets in a 10-time run is 27.50. This value is 36.28 in the one-level clustering algorithm in the literature. The improvement in the mean value in this state in the WAPE index for 35 datasets is 8.78%. The first setting of the two-level clustering algorithm outperformed in 32 datasets out of 35 datasets. At its best, the first setting of the two-level algorithm could decrease the WAPE index by 31.34% in a specific dataset. At its worst, the algorithm could increase the index by 6.43%, which is negligible since it took place only in three datasets. Figure 6 displays the WAPE index for each dataset's mean of 10-time run and the average of all datasets for the two one-level and first setting of two-level clustering algorithms.

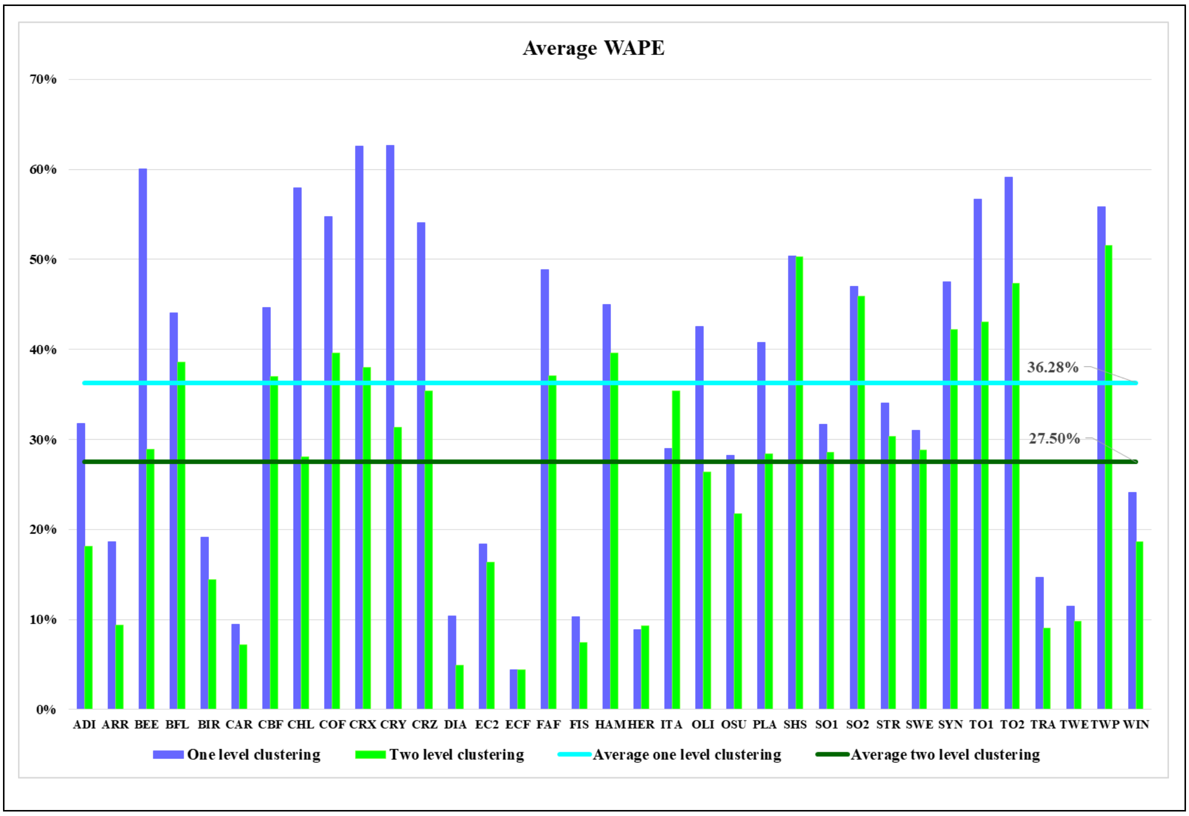


Fig 6. WAPE mean in two one-level and two-level clustering modes

In the second state, the optimum value obtained for the WAPE index in a 10-time run is also compared. The mean WAPE index for 35 datasets is 24.66% for the first setting of the proposed algorithm, while this value is 30.72% for one-level clustering. The mean value of improvement in this state for 35 datasets is 6.05%. In this state, the first setting of the two-level algorithm outperformed the

WAPE index in 31 of 35 datasets (table 5). Moreover, in the one-level clustering, the optimum WAPE value is 3.88%, while this value is 3.62% for the first setting of the two-level algorithm. At its best, the two-level algorithm could decrease the WAPE index by 32.17% in a specific dataset. Also, at its worst, the algorithm increased the index value by 8.76%. Figure 7 represents the optimum value of the WAPE index for a 10-time run of the datasets and their average value.

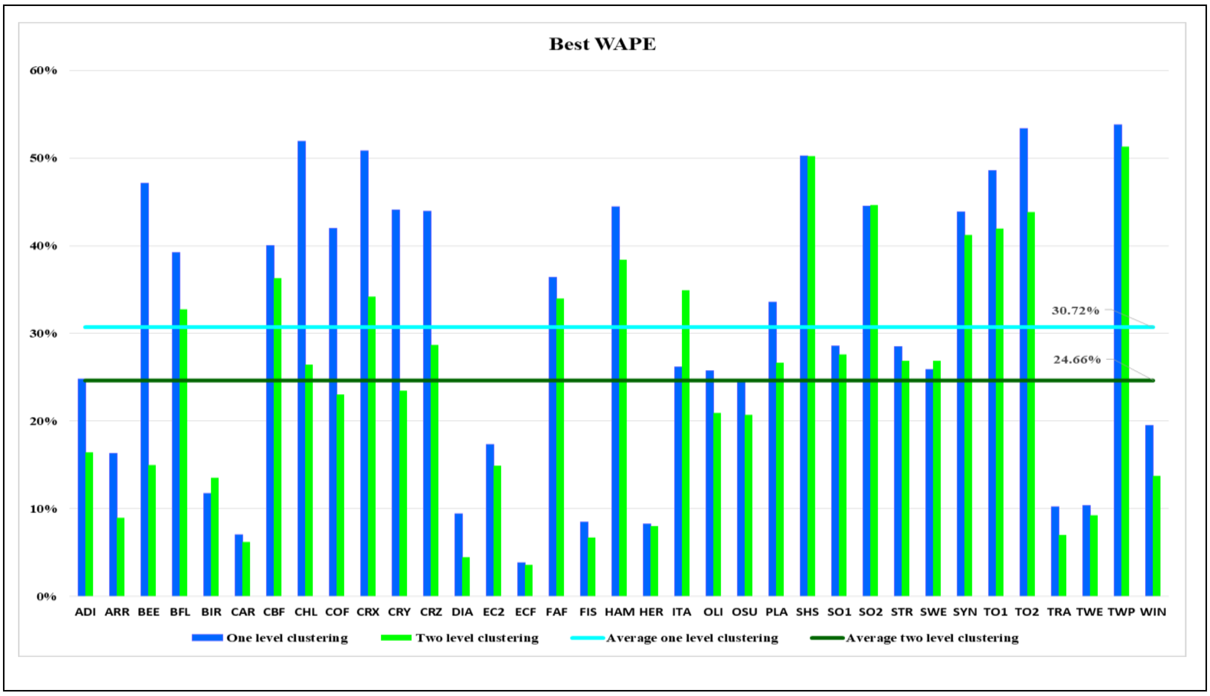


Fig 7. The optimum value of WAPE in two one-level and two-level clustering modes

The standard deviation of a 10-time run for each dataset was obtained in the WAPE index to evaluate the algorithm stability. Figure 8 and table 5 show the mean standard deviation of 2.00 for the datasets in the first setting of the two-level algorithm. However, this value for the one-level clustering forecast is 5.24, indicating a 3.24 decrease. The results also revealed that 31 out of 35 datasets had a lower standard deviation. Figure 8 shows the standard deviation of the WAPE index for a 10-time run of each dataset and the datasets' mean.

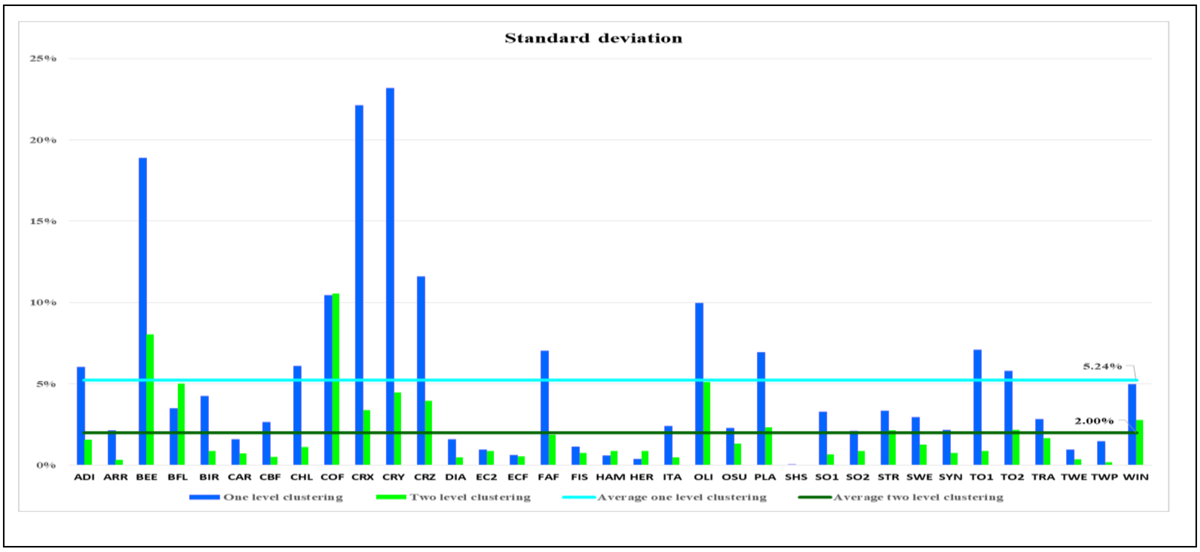


Fig 8. Standard deviation of 10-time run for two-level and one-level clustering

Table 4 provides the detailed WAPE index value and standard deviation of different runs for 35 datasets.

Table 4. WAPE index results and standard deviation for each dataset for the first setting

Data sets	The mean of the 10-time run						The Best of the 10-time run		
	WAPE(%)			Standard deviation(%)			WAPE(%)		
	one level clustering	two level clustering	difference	one level clustering	two level clustering	difference	one level clustering	two level clustering	difference
Adiac (ADI)	31.72	18.14	13.59	6.03	1.56	4.47	24.84	16.42	8.42
ArrowHead (ARR)	18.62	9.35	9.28	2.12	0.33	1.79	16.34	9.00	7.34
Beef (BEE)	60.01	28.91	31.10	18.88	8.04	10.84	47.18	15.01	32.17
BeetleFly (BFL)	44.07	38.56	5.51	3.50	5.03	-1.52	39.26	32.78	6.47
BirdChicken (BIR)	19.14	14.46	4.68	4.23	0.87	3.36	11.79	13.52	-1.72
Car (CAR)	9.48	7.17	2.31	1.60	0.72	0.88	7.08	6.19	0.89
CBF (CBF)	44.65	36.95	7.70	2.65	0.51	2.14	40.02	36.32	3.70
ChlorineConcentration (CHL)	57.92	28.05	29.87	6.09	1.12	4.97	51.91	26.47	25.44
Coffee (COF)	54.73	39.62	15.11	10.45	10.55	-0.10	41.98	23.02	18.96
Cricket_X (CRX)	62.56	38.02	24.54	22.12	3.37	18.75	50.82	34.18	16.64
Cricket_Y (CRY)	62.69	31.35	31.34	23.18	4.46	18.72	44.09	23.51	20.58
Cricket_Z (CRZ)	54.07	35.41	18.67	11.59	3.97	7.62	43.93	28.67	15.26
DiatomSizeReduction (DIA)	10.36	4.89	5.47	1.60	0.48	1.12	9.47	4.46	5.00
ECG200 (EC2)	18.36	16.33	2.03	0.94	0.87	0.07	17.33	14.93	2.40
ECGFiveDays (ECF)	4.38	4.40	-0.02	0.61	0.55	0.05	3.88	3.62	0.27
FaceFour (FAF)	48.85	37.03	11.82	7.04	1.87	5.17	36.38	34.01	2.38
FISH (FIS)	10.32	7.46	2.87	1.13	0.76	0.37	8.48	6.71	1.77
Ham (HAM)	45.01	39.58	5.43	0.58	0.87	-0.30	44.46	38.41	6.05
Herring (HER)	8.89	9.29	-0.39	0.38	0.86	-0.48	8.27	8.01	0.26
ItalyPowerDemand (ITA)	28.99	35.41	-6.43	2.39	0.48	1.91	26.16	34.92	-8.76
OliveOil (OLI)	42.50	26.36	16.14	9.97	5.12	4.85	25.75	20.96	4.79
OSULeaf (OSU)	28.24	21.74	6.50	2.27	1.33	0.94	24.43	20.71	3.73
Plane (PLA)	40.80	28.40	12.40	6.95	2.33	4.61	33.58	26.63	6.95
ShapeletSim (SHS)	50.38	50.26	0.12	0.05	0.02	0.03	50.30	50.25	0.05
SonyAIBORobotSurface (SO1)	31.71	28.60	3.11	3.28	0.67	2.60	28.61	27.63	0.98
SonyAIBORobotSurfaceII (SO2)	47.01	45.87	1.13	2.11	0.87	1.24	44.52	44.66	-0.14
Strawberry (STR)	34.02	30.36	3.67	3.35	2.15	1.20	28.52	26.91	1.61
SwedishLeaf(SWE)	30.98	28.85	2.13	2.93	1.26	1.67	25.88	26.85	-0.96
synthetic_control(SYN)	47.51	42.23	5.29	2.16	0.76	1.40	43.85	41.24	2.61
ToeSegmentation1(TO1)	56.67	43.05	13.62	7.09	0.88	6.21	48.58	41.99	6.59
ToeSegmentation2(TO2)	59.09	47.38	11.71	5.78	2.17	3.61	53.39	43.85	9.54
Trace(TRA)	14.64	9.05	5.59	2.83	1.67	1.16	10.27	7.01	3.26
TwoLeadECG(TWE)	11.45	9.76	1.68	0.95	0.37	0.58	10.37	9.23	1.14
Two_Patterns(TWP)	55.84	51.57	4.28	1.46	0.19	1.27	53.85	51.34	2.51
Wine(WIN)	24.09	18.65	5.44	4.97	2.78	2.20	19.55	13.79	5.76
Average	**36.28**	**27.50**	**8.78**	**5.24**	**2.00**	**3.24**	**30.72**	**24.66**	**6.05**

Table 5 summarizes the performance of the two-level algorithm based on different performance indices.

Table 5. Summary of the first setting of the proposed algorithm results in two modes of mean and the optimum answer

	One level clustering		Two level clustering (Hierarchical)	
	Average of 10 run	Best of 10 run	Average of 10 run	Best of 10 run
WAPE	36.28	30.72	27.50	24.66
Standard deviation	5.24	-	2.00	-
Maximum improvement in one dataset	-	-	31.34	32.17
The maximum amount of WAPE decreases in one dataset	-	-	-6.43	-8.76
The number of the dataset with the improved WAPE	-	-	32	31

4-Conclusion

This study proposed a two-level clustering algorithm for forecasting a group of short time series through seven significant steps. The purpose is to create more accurate clusters and improve the precision of the fitted models for each cluster. We have run the algorithm ten times on 35 datasets to assess its efficiency. We first implemented the algorithm according to the type of clustering algorithm (hierarchical, k-means) used in the first level in two different settings, selecting the best setting using different criteria. Then we compared the selected best setting with the one-level algorithm. The results showed that using the hierarchical algorithm in the first level of clustering could have better results. The mean in the 10-time run revealed that the first setting of the two-level clustering algorithm could increase the WAPE index value by 8.78% and reach 27.50% compared to the one-level algorithm. When selecting the optimum result of a 10-time run, the algorithm could decrease the WAPE index by 6.05% and get 24.66%. To assess the algorithm stability, the standard deviation of the 10-time run of the first setting of the proposed two-level algorithm was compared with the one-level algorithm. The results demonstrated that the two-level algorithm had a standard deviation of 2.00, which is 3.24 better than that of the one-level algorithm.

5-Future studies

The proposed two-level algorithm has two clustering levels. The hierarchical algorithm was used for the first-level clustering and the K-Means algorithm for the second level. Other clustering algorithms, such as Dbscan or Kmedoids, can be evaluated at different levels. Furthermore, various combinations of these algorithms can be applied for clustering at different levels. In addition, other time series clustering methods, like feature extraction, can be employed for first-level clustering. In the machine learning model fitting, only the MLP model was used; hence, the impact of deep learning models can be assessed in this algorithm.

Data availability.

Sequence data that support the findings of this study have been deposited in the http://www.cs.ucr.edu/~eamonn/time_series_data/

References

Abbasimehr, H., & Paki, R. (2022). Improving time series forecasting using LSTM and attention models. *Journal of Ambient Intelligence and Humanized Computing, 13*(1), 673-691.

Al-Hiary, H., Bani-Ahmad, S., Reyalat, M., Braik, M., & Alrahamneh, Z. (2011). Fast and accurate detection and classification of plant diseases. *International Journal of Computer Applications, 17*(1), 31-38.

Arias, M. B., & Bae, S. (2016). Electric vehicle charging demand forecasting model based on big data technologies. *Applied Energy, 183*, 327-339.

Astakhova, N. N., Demidova, L. A., & Nikulchev, E. V. (2015). Forecasting method for grouped time series with the use of k-means algorithm. *arXiv preprint arXiv:1509.04705*.

Bock, C. (2018). *Forecasting energy demand by clustering smart metering time series.* Paper presented at the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems.

Borghi, P. H., Zakordonets, O., & Teixeira, J. P. (2021). A COVID-19 time series forecasting model based on MLP ANN. *Procedia Computer Science, 181*, 940-947.

Boshnakov, G. N. (2016). Introduction to Time Series Analysis and Forecasting, Wiley Series in Probability and Statistics, by Douglas C. Montgomery, Cheryl L. Jennings and Murat Kulahci (eds). Published by John Wiley and Sons, Hoboken, NJ, USA, 2015. Total number of pages: 672 Hardcover: ISBN: 978-1-118-74511-3, ebook: ISBN: 978-1-118-74515-1, etext: ISBN: 978-1-118-74495-6. In: Wiley Online Library.

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control*: John Wiley & Sons.

Chen, I.-F., & Lu, C.-J. (2017). Sales forecasting by combining clustering and machine-learning techniques for computer retailing. *Neural Computing and Applications, 28*(9), 2633-2647.

Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C.-C. M., Zhu, Y., Gharghabi, S., . . . Keogh, E. (2019). The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica, 6*(6), 1293-1305.

de Araújo Morais, L. R., & da Silva Gomes, G. S. (2022). Forecasting daily Covid-19 cases in the world with a hybrid ARIMA and neural network model. *Applied Soft Computing, 126*, 109315.

de Jesús Rubio, J. (2017). A method with neural networks for the classification of fruits and vegetables. *Soft Computing, 21*(23), 7207-7220.

Dong, X., Qian, L., & Huang, L. (2017). *Short-term load forecasting in smart grid: A combined CNN and K-means clustering approach.* Paper presented at the 2017 IEEE International Conference on Big Data and Smart Computing (BigComp).

Hajirahimi, Z., & Khashei, M. (2022). A Novel Parallel Hybrid Model Based on Series Hybrid Models of ARIMA and ANN Models. *Neural Processing Letters*, 1-19.

Hu, Y., & Xiao, F. (2022). A novel method for forecasting time series based on directed visibility graph and improved random walk. *Physica A: Statistical Mechanics and its Applications, 594*, 127029.

Huang, X., Ye, Y., Xiong, L., Lau, R. Y., Jiang, N., & Wang, S. (2016). Time series k-means: A new k-means type smooth subspace clustering for time series data. *Information Sciences, 367*, 1-13.

Islam, M., & Sivakumar, B. (2002). Characterization and prediction of runoff dynamics: a nonlinear dynamical view. *Advances in water resources, 25*(2), 179-190.

Kedia, V., Thummala, V., & Karlapalem, K. (2005). *Time Series Forecasting through Clustering-A Case Study*. Paper presented at the COMAD.

Keogh, E., Chu, S., Hart, D., & Pazzani, M. (2004). Segmenting time series: A survey and novel approach. In *Data mining in time series databases* (pp. 1-21): World Scientific.

Kobylin, O., & Lyashenko, V. (2020). Time series clustering based on the k-means algorithm. *Journal La Multiapp, 1*(3), 1-7.

Koosha, H., Ghorbani, Z., & Nikfetrat, R. (2022). A Clustering-Classification Recommender System based on Firefly Algorithm. *Journal of AI and Data Mining, 10*(1), 103-116.

Lee, C.-H., Su, Y.-Y., Lin, Y.-C., & Lee, S.-J. (2017). *Time series forecasting based on weighted clustering*. Paper presented at the 2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA).

Li, D., Zhao, Y., & Li, Y. (2019). Time-series representation and clustering approaches for sharing bike usage mining. *IEEE access, 7*, 177856-177863.

Li, P., Wu, W., & Pei, X. (2023). A separate modeling approach for short-term bus passenger flow prediction based on behavioral patterns: A hybrid decision tree method. *Physica A: Statistical Mechanics and its Applications*, 128567.

Lucas, J. M., & Saccucci, M. S. (1990). Exponentially weighted moving average control schemes: properties and enhancements. *Technometrics, 32*(1), 1-12.

Mai, S. D., Ngo, L. T., & Trinh, H. L. (2018). *Satellite image classification based spatial-spectral fuzzy clustering algorithm.* Paper presented at the Asian Conference on Intelligent Information and Database Systems.

Norwawi, N. M. (2021). Sliding window time series forecasting with multilayer perceptron and multiregression of COVID-19 outbreak in Malaysia. In *Data Science for COVID-19* (pp. 547-564): Elsevier.

Panapakidis, I. P. (2016). Clustering based day-ahead and hour-ahead bus load forecasting models. *International Journal of Electrical Power & Energy Systems, 80*, 171-178.

Pant, M., & Kumar, S. (2022). Fuzzy time series forecasting based on hesitant fuzzy sets, particle swarm optimization and support vector machine-based hybrid method. *Granular Computing, 7*(4), 861-879.

Parmezan, A. R. S., & Batista, G. E. (2015). *A study of the use of complexity measures in the similarity search process adopted by knn algorithm for time series prediction.* Paper presented at the 2015 IEEE 14th international conference on machine learning and applications (ICMLA).

Parmezan, A. R. S., Souza, V. M., & Batista, G. E. (2019). Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. *Information Sciences, 484*, 302-337.

Pérez, S. I., Moral-Rubio, S., & Criado, R. (2023). Combining multiplex networks and time series: A new way to optimize real estate forecasting in New York using cab rides. *Physica A: Statistical Mechanics and its Applications, 609*, 128306.

Polat, K. (2012). Classification of Parkinson's disease using feature weighting method on the basis of fuzzy C-means clustering. *International Journal of Systems Science, 43*(4), 597-609.

Said, A. A., Abd-Elmegid, L. A., Kholeif, S., & Gaber, A. A. (2018). Classification based on clustering model for predicting main outcomes of breast cancer using hyper-parameters optimization. *International Journal of Advanced Computer Science and Applications, 9*(12).

Sfetsos, A., & Siriopoulos, C. (2004). Combinatorial time series forecasting based on clustering algorithms and neural networks. *Neural computing & applications, 13*(1), 56-64.

Sinaga, K. P., & Yang, M.-S. (2020). Unsupervised K-means clustering algorithm. *IEEE access, 8*, 80716-80727.

Talkhi, N., Fatemi, N. A., Ataei, Z., & Nooghabi, M. J. (2021). Modeling and forecasting number of confirmed and death caused COVID-19 in IRAN: A comparison of time series forecasting methods. *Biomedical Signal Processing and Control, 66*, 102494.

Udler, M. S., Kim, J., von Grotthuss, M., Bonàs-Guarch, S., Cole, J. B., Chiou, J., . . . Atzmon, G. (2018). Type 2 diabetes genetic loci informed by multi-trait associations point to disease mechanisms and subtypes: a soft clustering analysis. *PLoS medicine, 15*(9), e1002654.

Weerakody, P. B., Wong, K. W., Wang, G., & Ela, W. (2021). A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing, 441*, 161-178.

Xu, S., Chan, H. K., Ch'ng, E., & Tan, K. H. (2020). A comparison of forecasting methods for medical device demand using trend-based clustering scheme. *Journal of Data, Information and Management*, 1-10.

Yu, B., Song, X., Guan, F., Yang, Z., & Yao, B. (2016). k-Nearest neighbor model for multiple-time-step prediction of short-term traffic condition. *Journal of Transportation Engineering, 142*(6), 04016018.

Yu, C., Wang, L., Zhao, J., Hao, L., & Shen, Y. (2020). Remote sensing image classification based on RBF neural network based on fuzzy C-means clustering algorithm. *Journal of Intelligent & Fuzzy Systems, 38*(4), 3567-3574.