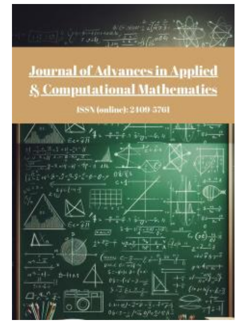
Journal of Advances in Applied & Computational Mathematics, 2024, 100-118

Published by Avanti Publishers

Journal of Advances in Applied & Computational Mathematics

ISSN (online): 2409-5761



Forecasting Non-stationary Time Series Using Deep Learning in a Fuzzy Time Series Framework and its Application to Stock Markets

Abdul J. Saleena*, John C. Jessy and M.C. Lineesh

Department of Mathematics, National Institute of Technology Calicut, NIT Campus, Kozhikode 673601, Kerala, India

ARTICLE INFO

Article Type: Research Article

Academic Editor: Sivasankar Shanmugam

Keywords:
Stock market
Deep learning
Fuzzy time series
Universe of discourse
Non-stationary time series
Variable-length interval partitioning

Timeline:
Received: August 11, 2024
Accepted: September 25, 2024
Published: October 09, 2024

Citation: Saleena AJ, Jessy JC, Lineesh MC. Forecasting non-stationary time series using deep learning in a fuzzy time series framework and its application to stock markets. J Adv Appl Computat Math. 2024; 11: 100-118.

DOI: https://doi.org/10.15377/2409-5761.2024.11.6

ABSTRACT

Non-stationary time series prediction is challenging due to its dynamic and complex nature. Fuzzy time series models offer a promising solution for forecasting such data, but a key challenge lies in partitioning the universe of discourse, which significantly impacts forecasting accuracy. Traditional fuzzy time series models often use equal-length interval partitioning, which is more suited for stationary data and limits their adaptability to non-stationary time series. This paper introduces a novel variable-length interval partitioning method designed specifically for non-stationary time series. The developed method combines a Long Short-Term Memory (LSTM) Autoencoder with K-means clustering, enabling dynamic, data-driven partitioning that adapts to the changing characteristics of the data. The LSTM Autoencoder encodes the time series, which is clustered using K-means, and intervals are defined based on cluster centers. Furthermore, the Variable Length Interval Partitioning-based Fuzzy Time Series model (VLIFTS) is developed by incorporating this partitioning method and the concepts of Markov chain and transition probability matrix. In this model, fuzzy sets are viewed as states of a Markov chain, and transition probabilities are used in the forecasting phase. The model is validated on stock market indices Nifty 50, NASDAQ, S&P 500, and Dow Jones. Stationarity and heteroscedasticity are tested using Augmented Dickey-Fuller (ADF) and Levene's tests respectively. Statistical forecast accuracy metrics Root Mean Squared Error (RMSE) and Mean Absolute Percent Error (MAPE) show that VLIFTS significantly improves forecasting accuracy over traditional models. This hybrid approach enhances fuzzy time series modelling and can be applied to various non-stationary time series forecasting problems.

*Corresponding Author
Email: saleenaabduljabbar@gmail.com
Tel: +(91) 8547705979

1. Introduction

Non-stationary time series are common in a wide range of real-world systems. For non-stationary time series, statistical properties like mean, variance, and higher-order moments vary over time. Non-stationarity is frequently caused by trends, seasonality, heteroscedasticity, and other irregular patterns over time. Therefore, utmost care should be taken when choosing a forecasting model based on the assumption of stationarity. Non-stationary time series forecasting plays a prominent role in analysing the performance time series of many real-world systems. Recently, fuzzy time series models have been used to predict non-stationary time series. The use of fuzzy logic [1] to analyse imprecise time series is called fuzzy time series.

In traditional fuzzy time series modelling, the Universe of Discourse (UoD) is partitioned into equal-length intervals. The length of intervals influences forecasting accuracy in fuzzy time series models. Huarng [2] extended Chen's [3] model by introducing distribution and average-based length. Yu [4] addressed two issues in fuzzy time series modelling: repeating and weighting of fuzzy logical relationships. Huarng [5] introduced a variable-length interval partitioning method for time series forecasting in which lengths of intervals are according to ratio-based lengths. Cheng et al. [6] used an adaptive expectation model in fuzzy time series modelling and applied it to predict stock prices. To evaluate the length of intervals, Yolcu et al. [7] applied a constrained optimization technique. Tsaur [8] introduced the Markov chain and transition probability matrix into fuzzy relations. Guney et al. [9] extended Tsaur [8] to a seasonal time series.

Seasonal time series occur frequently in a variety of real-world systems. In the literature, seldom work is done based on seasonal time series under fuzzy forecasting. Song [10] pioneered the use of a fuzzy time series model for seasonal time series. Egrioglu et al. [11] proposed Seasonal Autoregressive Integrated Moving Average (SARIMA) models with a partial high-order bivariate fuzzy model and neural networks. Suhartono and Lee [12] incorporated Winter's model with a weighted fuzzy time series model. Tseng and Tzeng [13] combined the potentialities of SARIMA [14] model and the fuzzy regression model. An improved FTS model [15] is introduced to tackle the seasonality aspect of time series.

Certain non-stationary time series occur as a result of unpredictable changes [16]. Choosing a suitable forecasting model is difficult in this situation. These implicit kinds of changes are called concept drifts [17]. In non-stationary time series analysis, transformation techniques were used initially, for example, the Autoregressive Integrated Moving Average (ARIMA) model [14]. Later, time-variant forecasting models like time-variant fuzzy time series models were introduced. A Time-Variant FTS model (TVFTS) was proposed [18] initially in the canon of literature in which training occurs in a moving window methodology. That is, the rule base of the model was built for every window of length *w* of the time series with the same UoD and fuzzy sets/knowledge base. Several time- variant models after [18] include hybrid approaches [19], transformation [20], seasonal techniques [13] and weighing techniques [15]. Non-stationary time series like stock prices and their forecasting is discussed in [21, 22].

The concept of neutrosophic sets [23], granular computing, and bio-inspired techniques [24] are utilized in FTS modeling. A review of FTS modeling is briefly discussed in [25, 26]. Variations in time series can be utilized and interpolated [27] in the future for forecasting non-stationary time series. In [28], non-stationary fuzzy sets are used to handle heteroscedastic time series having predictable variance over time. Recently, a Non-stationary Fuzzy Time Series (NSFTS) model [29] is introduced in which membership function parameters are updated according to the variations in a given time series. In [29], UoD is time-variant and the rule-base is time-invariant. The utility of Adaptive Auto-regressive (AR) modeling combined with Kalman filtering for managing non-stationary time series, particularly within biomedical applications, has been well established in [30]. This approach allows for real-time adjustment of model parameters, improving forecasting accuracy in environments where the underlying dynamics are constantly evolving. This work builds upon the smoothness priors approach for modeling non-stationary covariance in time series, ensuring a smooth evolution of AR coefficients [31]. A high dimensional time series is discussed and its forecasting technique which is based on fuzzy time series and embeddings is developed in [32] and [33]. The concept of intuitionistic fuzzy sets is utilized and discussed in [34-37] for forecasting non-stationary time series. The concept of hesitant fuzzy sets is used in time series forecasting [38] and [39]. Fixed points provide a mathematical framework for understanding how and why time series models stabilize, cycle, or diverge, which is crucial for making accurate predictions, especially in complex or nonlinear systems. Farid [40] developed an iterative procedure for solving fixed point problem in a Hilbert space. Some of the solutions to a fixed point problem include [41-43].

The fuzzy time series model is an effective tool for forecasting non-stationary time series. However, a critical challenge lies in the partitioning of the universe of discourse, which has a significant impact on forecasting accuracy. In this paper, a novel variable-length interval partitioning method is developed for partitioning the universe of discourse of a non-stationary time series into different intervals of varying length. This partitioning method progresses through the stages of encoding, clustering, and construction of intervals. Unlike other works in the literature, the developed hybrid method extracts the unique features of the time series using the LSTM Autoencoder. After that, the K-means clustering algorithm is performed to cluster this featured data. The intervals are therefore constructed based on the cluster centers obtained from the K-means clustering algorithm. Also, a fuzzy time series model is developed by incorporating the developed partitioning method, the notion of the Markov chain, and the transition probability matrix.

The remaining sections of this paper are structured in the following way: basic definitions related to fuzzy sets and fuzzy time series are discussed in Section 2. The developed partitioning method and fuzzy time series model are detailed in Sections 3 and 4 respectively. The application of the developed model to the stock markets and its comparison with existing models are discussed in section 5. Conclusions and future works are given in Section 6.

2. Preliminaries

2.1. Fuzzy Time Series

In light of works by Zadeh [44, 45], Song and Chissom [46, 47] introduced the idea of fuzzy time series. Fuzzy time series is an exceptionally powerful process with semantic qualities as its perceptions. The universe of discourse in Fuzzy Time Series (FTS) is subdivided and each division is linked to a fuzzy set. To represent the underlying nature of the time series, IF-THEN time patterns are identified from the training part of the fuzzy time series, and the rule base is built as fuzzy logical relationships. A first-order time-invariant fuzzy time series forecasting model was proposed by Song and Chissom [47] in which the max-min composition operation is utilized in FTS modelling. In Chen [3], fuzzy logical relationship groups are formed from fuzzy logical relationships, and the simple arithmetic mean is utilized for forecasting fuzzy time series.

Definition 1 [46] Let U be a universal set. A fuzzy set A in U is characterized by its membership function $\mu_A : U \rightarrow [0, 1]$ and $\mu_A(x)$ is interpreted as the degree of membership of element x in fuzzy set A for each $x \in U$.

Definition 2 [46] Let $Y(t)$, $t = 0, 1, \ldots$, a subset of R, be the universe of discourse on which fuzzy sets $f_i(t)$, $i = 1, 2, \ldots$ are defined and $F(t)$ is the collection of $f_i(t)$, $i = 1, 2, \ldots$. Then $F(t)$ is called a fuzzy time series on $Y(t)$, $t = 0, 1, \ldots$

Here $F(t)$ can be considered as a linguistic variable and $f_i(t)$, $i = 1, 2, \ldots$ as the possible linguistic values of $F(t)$. Fuzzy sets are the observations of a fuzzy time series. Fuzzy time series models are classified into first-order and higher-order fuzzy time series models depending on the number of previous observations considered in modelling. The first-order FTS model considers relationships among fuzzy sets with only immediate previous time points. Fuzzy time series models are further classified based on their dependency on time. In time-invariant FTS models, the knowledge base (fuzzy sets) and rule base remain stable over time, but in time-variant FTS models, either the knowledge base, the rule base, or both change over time.

2.2. Long Short-Term Memory Autoencoder

An LSTM autoencoder design uses Long Short-Term Memory (LSTM) [48] layers, a kind of recurrent neural network, to encode and decode sequential input. An LSTM autoencoder has the following parts:

1. LSTM encoder: LSTM encoders capture temporal dependencies and extract meaningful features from input sequences. The LSTM encoder generates a compressed representation known as the latent space.
2. Repeat Vector Layer: The repeat vector layer replicates the encoded representation over multiple time steps to prepare it for decoding by the LSTM.
3. LSTM Decoder: An LSTM decoder reconstructs a sequence closely matching the input sequence from the repeated encoded representation.

2.3. K-means Clustering Algorithm

K-means clustering [49] is a popular unsupervised learning algorithm used to partition a dataset into *k* distinct, non-overlapping clusters based on the features of the data points. The goal is to minimize the within-cluster variance, which is the sum of squared distances between each data point and the centroid of its assigned cluster.

Objective Function

The objective of K-means is to minimize the within-cluster sum of squares (WCSS), also known as the inertia:

$$WCSS = \sum_{J=1}^{K} \sum_{x_i \in C_j} || x_i - \mu_j ||^2$$

This function measures the compactness of the clusters. The algorithm iteratively reduces the WCSS until it reaches a local minimum. The main steps of K-means clustering algorithm are:

- **Initialization**

Choose the number of clusters *k*.

Randomly select *k* data points from the dataset as the initial cluster centroids $\mu_1, \mu_2, \ldots, \mu_k$.

- **Assignment Step**

For each data point x_i in the dataset, assign it to the nearest cluster centroid based on the Euclidean distance:

$$c_i = arg \min_j ||x_i - \mu_j||^2$$

where c_i is the index of the cluster assigned to data point x_i, and μ_j is the *j*-th cluster centroid.

- **Update Step**

Recalculate the centroids of the clusters by taking the mean of all data points assigned to each cluster:

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$

where C_j is the set of data points assigned to cluster *j*, and $|C_j|$ is the number of points in cluster *j*.

- **Convergence Check**

Repeat the assignment and update steps until the centroids no longer change or the changes are smaller than a predefined threshold, indicating that the algorithm has converged.

3. The Variable-length Interval Partitioning Method

In this section, the method developed for variable-length interval partitioning of UoD is presented. The devel- oped method comprises the following three stages:

1. **Encoding:** In the stage of encoding, the temporal dependencies of the given time series are captured and extract mean- ingful features from it. The encoder generates a compressed representation of the given data. In this stage, LSTM Autoencoder is utilized to encode the given time series.
2. **Clustering:** The encoded time series is clustered in this stage of clustering. In this stage, the entire time series is clustered into different groups, and cluster centers are generated. In this stage, K-means clustering algorithm is utilized to cluster the encoded version of a given time series.
3. **Construction of Intervals:** In this stage, intervals are constructed [50] using cluster centers obtained in the clustering stage. Let LB_i and UB_i denote the lower bound and upper bound of i^{th} interval respectively. Let $LB_1 = D_{min} - \sigma$, $UB_m = D_{max} + \sigma$ and C_i be the i^{th} cluster center. Then,

$$UB_i = \frac{C_i + C_{i+1}}{2} \tag{1}$$

$$LB_{i+1} = UB_i, i = 1, 2, ..., k \tag{2}$$

3.1. Algorithm of the Developed Partitioning Method

The steps involved in the developed variable-length interval partitioning method are given below:

Step 1: Input a non-stationary time series and the number of intervals, *m.*

Step 2: Input the non-stationary time series into the LSTM Autoencoder network.

Step 3: Extract the encoder part of LSTM Autoencoder such that the reconstruction error is minimum.

Step 4: Input this encoder to the K-means clustering algorithm and extract k cluster centers.

Step 5: Intervals $u_1, u_2, .., u_m$ are constructed based on cluster clusters and are given by

$$u_1 = [LB_1, UB_1], u_2 = [LB_2, UB_2], ..., u_m = [LB_m, UB_m] \tag{3}$$

The flowchart for the developed algorithm is given in Fig. (**1**).

4. The Developed VLIFTS Model

A fuzzy time series model called VLIFTS model is developed by incorporating the developed partitioning method discussed above, the notion of the Markov chain, and the transition probability matrix. The developed model progresses through the following stages:

1. **Partitioning:** In this stage, the universe of discourse is partitioned into intervals of varying length. The hybrid method developed based on the LSTM Autoencoder and K-means clustering algorithm discussed in Section 3 is utilized to achieve this.
2. **Fuzzification:** In the fuzzification stage, fuzzy sets are defined over the intervals of the universe of discourse. Each data point is fuzzified based on the degree of membership of that data point in a fuzzy set. As a result, we obtain a fuzzy time series corresponding to the given crisp time series.
3. **Rule-base building:** In this stage, first-order IF-THEN temporal relationships are built among fuzzy sets and grouped into classes according to the same base fuzzy set.
4. **Defuzzification:** Different fuzzy sets are considered as different states of a Markov Chain. A transition probability matrix is therefore formed to model both the occurrence and recurrent occurrence of fuzzy sets. The forecast is computed as the weighted average of midpoints of leading fuzzy sets with weights as transition probabilities.

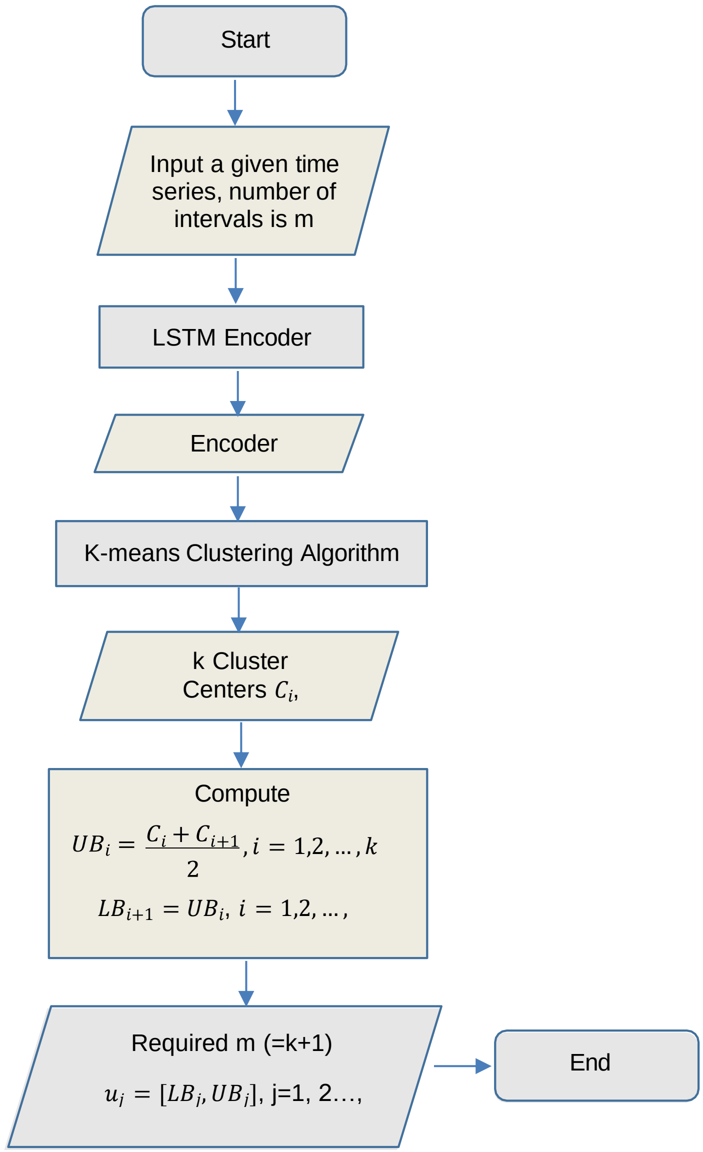


Figure 1: Flowchart of developed variable-length interval partitioning method.

4.1. Algorithm of the Developed VLIFTS Model

The algorithm of the developed VLIFTS model is given below:

Step 1: Input a given time series {Yt|t ∈ T }.

Step 2: Check for stationarity of a given time series. If it is non-stationary, go to step 3.

Step 3: Define the universe of discourse as $U = [D_{min} - \sigma, D_{max} + \sigma]$.

Step 4: Construct the intervals $u_1, u_2, \ldots, u_m$ using the hybrid method discussed in Section 3.

Step 5: Define fuzzy sets $A_j, j = 1, 2, \ldots, k$ over U as

$$\mu_{A_j}(y) = \begin{cases} 1, y \in u_j \\ 0.5, y \in u_{j-1} \, or \, y \in u_{j+1} \\ 0, \text{elsewhere} \end{cases} \quad (4)$$

Here $\mu_{A_j}(y)$ is the membership degree of a data point y in a fuzzy set A_j, $j = 1, 2, \ldots, k$.

Step 6: Fuzzification: If a data point's maximum membership falls under the fuzzy set A_k, then its fuzzified value is considered A_k.

Step 7: Set up IF-THEN Rules: First-order fuzzy sets' temporal relationships are obtained by the rule $A_L \rightarrow A_R$, where A_L and A_R are given as:

$$A_L = \underset{\text{Aj}}{\arg\max}\{\mu_{A_j}(y_{t-1})\} \tag{5}$$

$$A_R = \underset{\text{Aj}}{\arg\max}\{\mu_{A_j}(y_t)\} \tag{6}$$

Step 8: Rule-base building: Collect all temporal relationships with the base AL at time t − 1 into one class, AL → Ai, Aj, Here, the set of all possible outcomes at time t leading from the same AL are arranged and in this way, we have got several classes varying subject to the base AL.

Step 9: Construction of transition probability matrix: In this step, fuzzy sets are viewed as states of the Markov process. The transition probability matrix (TPM) P = (pij) is formed asbased on the first-order temporal relationships and the rule-base classes. Here pij, i, j = 1, 2, ...k, is the transition probability of the system from state Ai to state Aj and are computed by pij = nij on the basis of nij as the number of times the system visited the state Aj from state Ai and ni, the total number of times the system visited the state Ai.

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdot & \cdot & \cdot & p_{1k} \\ p_{21} & p_{22} & \cdot & \cdot & \cdot & p_{2k} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ p_{k1} & p_{k2} & \cdot & \cdot & \cdot & p_{kk} \end{bmatrix} \tag{7}$$

Step 10: Defuzzification: The forecast at (t + 1)th time point, yˆt+1 is computed based on the TPM from the rule-base classes. It is the weighted average of mid-points of the leading fuzzy sets with weights as pij's. If the base fuzzy set Aj leads to the same fuzzy set Aj, in rule-base class, mid-point of fuzzy set Aj is replaced by just the previous observation, that is yt.

Therefore,

$$\hat{y}_{t+1} = p_{j1}.m_1 + \ldots + p_{jj}.y_t + \ldots + p_{kk}.m_k \tag{8}$$

The flowchart of the algorithm of the developed VLIFTS model is given in Fig. (**2**).

5. Results and Discussion

In this section, the developed VLIFTS model is applied to stock markets and results are consolidated in the following subsections.

5.1. Data Description

In this section, the developed VLIFTS model presented in this paper is applied to four stock market in- dices, that is NSE-Nifty 50, NASDAQ, S&P 500, and Dow Jones. The data is available on www.nesindia.com, https://finance.yahoo.com/quote/%5EIXIC/, https://finance.yahoo.com/quote/%5EGSPC/, https://finance.yahoo.com/qu Time plots of the daily closing prices from NSE-Nifty 50, NASDAQ, S&P 500, and Dow Jones are given in Fig. (**3**-**6**) respectively. The number of fuzzy sets k chosen is 10.

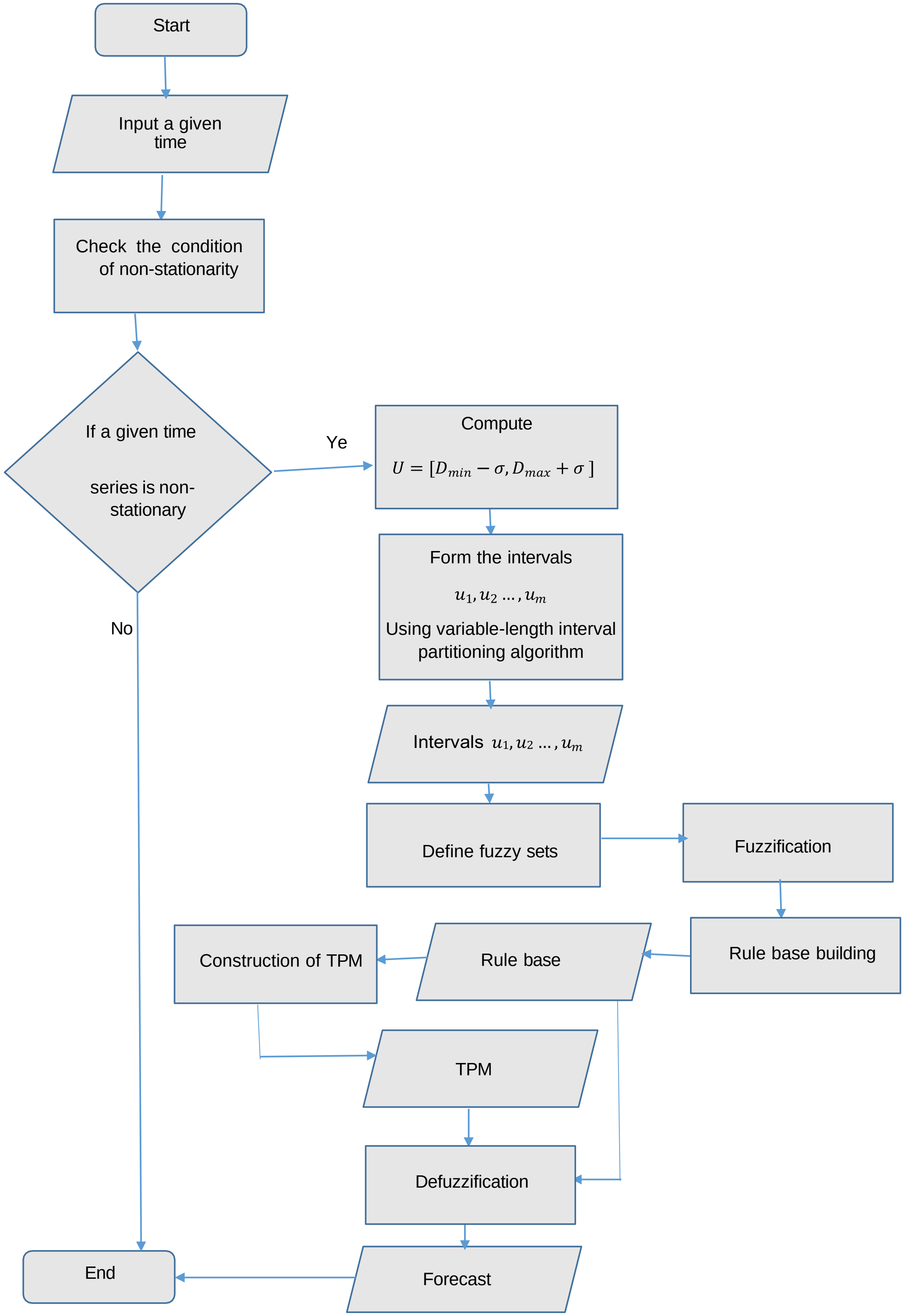


Figure 2: Flowchart of developed VLIFTS model.

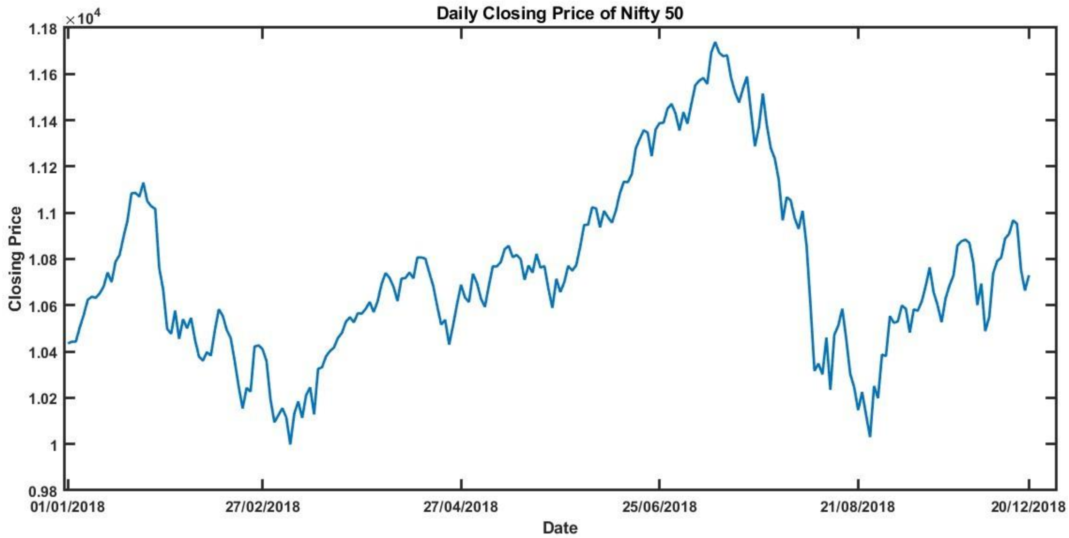


Figure 3: Daily closing price of Nifty-50 stock market index for the year 2018.

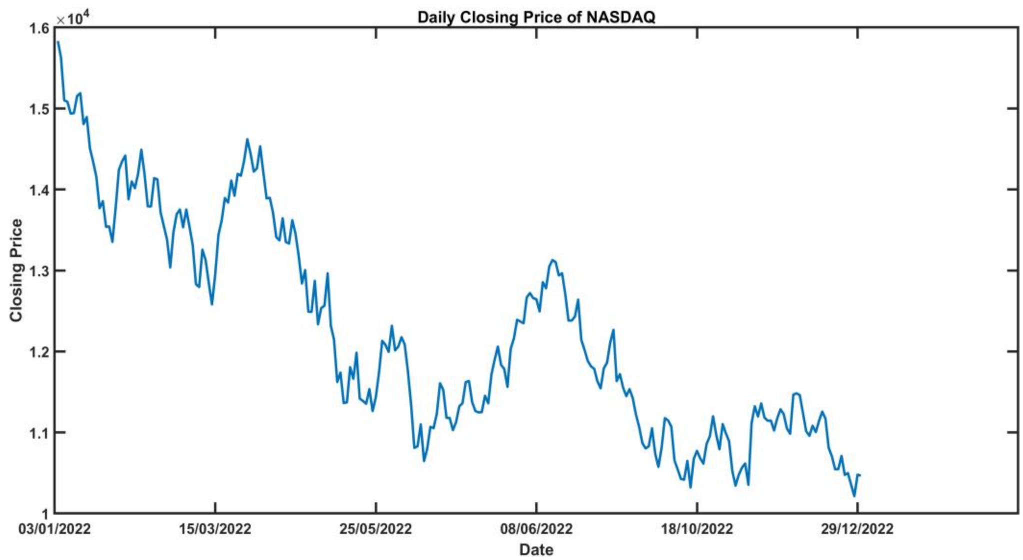


Figure 4: Daily closing price of NASDAQ stock market index for the year 2022.

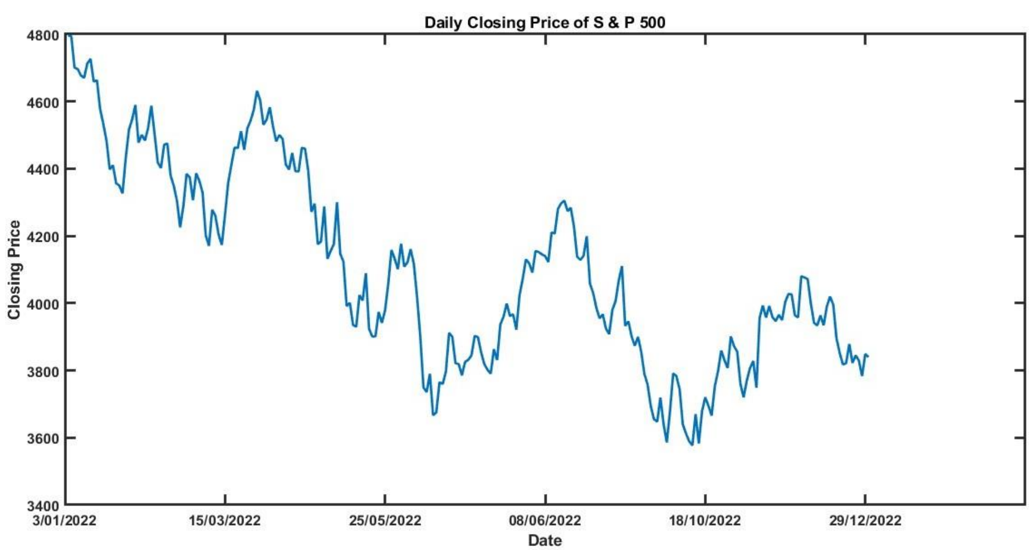


Figure 5: Daily closing price of S&P 500 stock market index for the year 2022.

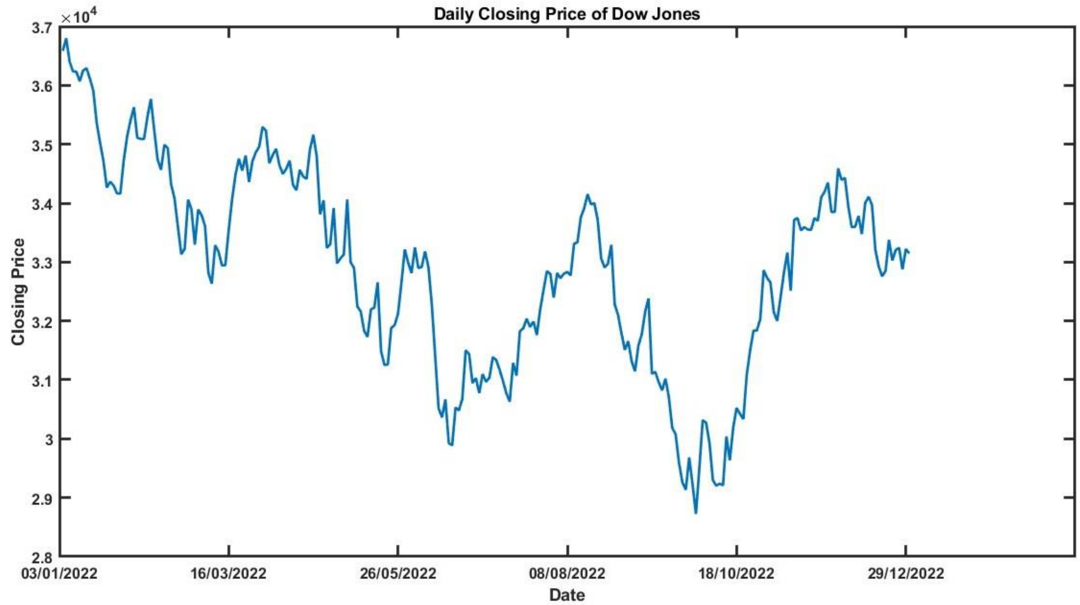


Figure 6: Daily closing price of Dow Jones stock market index for the year 2022.

5.2. Checking Stationarity and Heteroscedasticity

The test for stationarity of the daily closing price data is performed using the Augmented Dickey-Fuller (ADF) test. Under the ADF test, the null and alternative hypotheses are:

H_0 : The given time series is non-stationary

H_1 : The given time series is stationary

The test results are given in Table **1**. From the test results, it is concluded that all the time series considered are non-stationary.

Table 1: Testing of stationarity.

ADF Test			
Time Series	**Statistic**	**p-value**	***H*0**
NSE-Nifty 50	-2.3384	0.1598	Accept
NASDAQ	-2.4448	0.1295	Accept
S&P 500	-2.4359	0.1318	Accept
Dow Jones	-2.4883	0.1183	Accept

Levene's test tests the heteroscedasticity of considered time series of stock market indices. Under Levene's test, the null and alternative hypotheses are:

H_0 : The given time series is homoscedastic

H_1 : The given time series is heteroscedastic.

The test results are given in Table **2**. From the test results, it is concluded that all the time series except Dow Jones are heteroscedastic.

Table 2: Testing of heteroscedasticity.

Levene's Test			
Time Series	**Statistic**	**p-value**	***H0***
NSE-Nifty 50	44.0866	2.0648e−10	Reject
NASDAQ	36.6391	5.1975e−09	Reject
S&P 500	26.8525	4.5393e−07	Reject
Dow Jones	0.3839	0.5360	Accept

5.3. Illustration of the Developed Partitioning Method

In this section, an illustration of the developed partitioning method is discussed using the daily closing price of the Indian stock market index Nifty 50 of NSE. The daily closing price data is split into train set (earliest 70 %), validation set (next 15 %) and test set (last 15 %).

The training and validation loss while training the LSTM Autoencoder is given in Fig. (**7**). The training loss reflects how well the model fits the training data, whereas the validation loss reveals how well the model fits the new data. To check the model performance, it is usually plotted training loss and validation loss together in one figure. From this plot, the model performance to the data such as underfitting, overfitting, and good fit can be identified. Here, from Fig. (**7**), both the training loss and the validation loss start to decline and stabilize at a specific point. Therefore it indicates the good performance of the model.

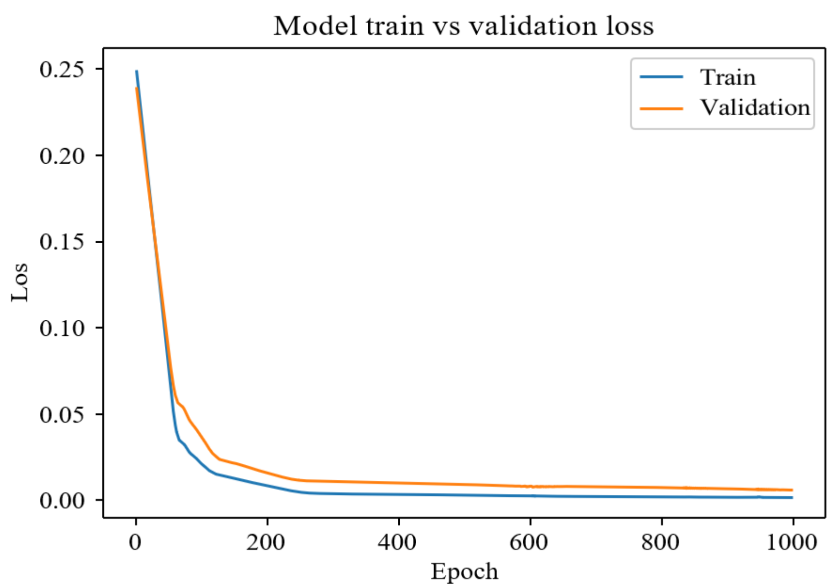


Figure 7: Nifty 50-train vs validation loss.

Once this LSTM Autoencoder model is fitted to the data, the encoder part alone is used for clustering along with the K-means clustering algorithm.

Interval Partitioning of UoD of Nifty 50

In this case, UoD is given by U = [9546.66, 12189.89]. The cluster centers are obtained from the K-means clustering algorithm by choosing the number of clusters 10. The cluster centers are given by: 9999.64, 10129.06, 10235.19, 10343.68, 10496.19, 10656.61, 10804.32, 10966.96, 11243.22, and 11515.45.

Therefore, the intervals are obtained from the equations 1, 2, and 3 as:

u_1 = [9546.66, 10064.35], u_2 = [10064.35, 10182.12]

u_3 = [10182.12, 10289.43], u_4 = [10289.43, 10419.935]

u_5 = [10419.935, 10576.4], u_6 = [10576.4, 10730.46]

u_7 = [10730.46, 10885.64], u_8 = [10885.64, 11105.09]

u_9 = [11105.09, 11379.33], u_{10} = [11379.33, 12189.89]

Defuzzification Process

This section details the determination of forecasts based on the defuzzification process (8) for Nifty 50.

The universe of discourse is given by $U = \{u_1, u_2, \ldots, u_{10}\}$.

Here we choose the number of fuzzy sets as 10. Therefore the fuzzy sets are defined by

$A_i = \{\ldots + 0.5/u_{i-1} + 1/u_i + 0.5/u_{i+1} + \ldots\}$

$i = 1, 2, \ldots, 10$. The rule base classes for the train set are given by

$A_1 \rightarrow A_2$

$A_2 \rightarrow A_1, A_2, A_3, A_4$

$A_3 \rightarrow A_2, A_3, A_5$

$A_4 \rightarrow A_3, A_4, A_5$

$A_5 \rightarrow A_4, A_5, A_6$

$A_6 \rightarrow A_5, A_6, A_7$

$A_7 \rightarrow A_6, A_7, A_8$

$A_8 \rightarrow A_7, A_8, A_9$

$A_9 \rightarrow A_8, A_9, A_{10}$

$A_{10} \rightarrow A_9, A_{10}$

The transition probability matrix P for the train set is given by

$$P=\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.125 & 0.375 & 0.375 & 0.125 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5714 & 0.2857 & 0 & 0.1486 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1667 & 0.6667 & 0.1667 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0.7 & 0.2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1428 & 0.6 & 0.2571 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.2857 & 0.6428 & 0.0714 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0555 & 0.8333 & 0.111 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0909 & 0.7272 & 0.1818 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.05 & 0.95 \end{bmatrix}$$

The first observation of the train set of the Nifty 50 dataset is 10435.55. The fuzzy set corresponding to this observation is A_5. The rule base class for this fuzzy set A_5 is $A_5 \rightarrow A_4, A_5, A_6$. The fitted value for the second observation of the train set is determined as:

$$\hat{Y}_2 = p_{54} \times m_4 + p_{55} \times pr_{value} + p_{56} \times m_6 \quad (9)$$

$$\hat{Y}_2 = 0.1 \times 10354.685 + 0.7 \times 10435.55 + 0.2 \times 10653.4325 \quad (10)$$

That is, $\hat{Y}_2$ = 10471.04. Here m_4, m_6 are midpoints of intervals u_4 and u_6 respectively. Also, pr_{value} is the previous observation, the first observation of the train set. Similarly, the first observation of the test set (172th observation) of the Nifty 50 is 11589.1. The fuzzy set corresponding to this observation is A_{10}. The rule base class for this fuzzy set is $A_{10} \rightarrow A_9, A_{10}$. The forecast value for the second observation (173th observation) of the test set is determined as:

$$\hat{Y}_{173} = p_{10,9} \times m_{10} + p_{10,10} \times prvalue \quad (11)$$

$$\hat{Y}_{173} = 0.05 \times 11242.2125 + 0.95 \times 11589.1 \quad (12)$$

That is, $\hat{Y}_{173}$ = 11571.755. The forecast of the test data and its original for the stock market index Nifty 50 is given in Fig. (**8**).

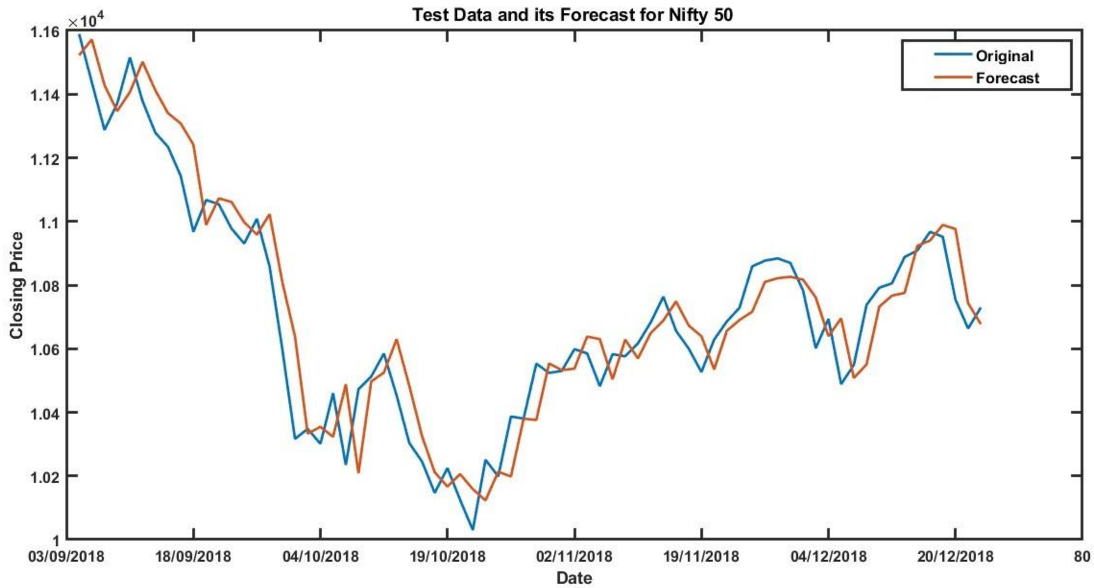


Figure 8: Comparison of forecasts of test data with original data for the stock market index Nifty 50.

5.4. Forecast of the Test Data for Other Selected Stock Market Indices

Forecasts of other selected stock market indices NASDAQ, S&P 500, and Dow Jones are discussed in this section. Similarly, as in the case of Nifty 50, other stock market indices are analysed. Training loss vs validation loss while training LSTM Autoencoder for stock market indices NASDAQ, S&P 500, and Dow Jones are given in Fig. (**9**, **11**, and **13**) respectively. Also, forecast of the test data for the stock market indices NASDAQ, S&P 500, and Dow Jones are given in Fig. (**10**, **12**, and **14**) respectively.

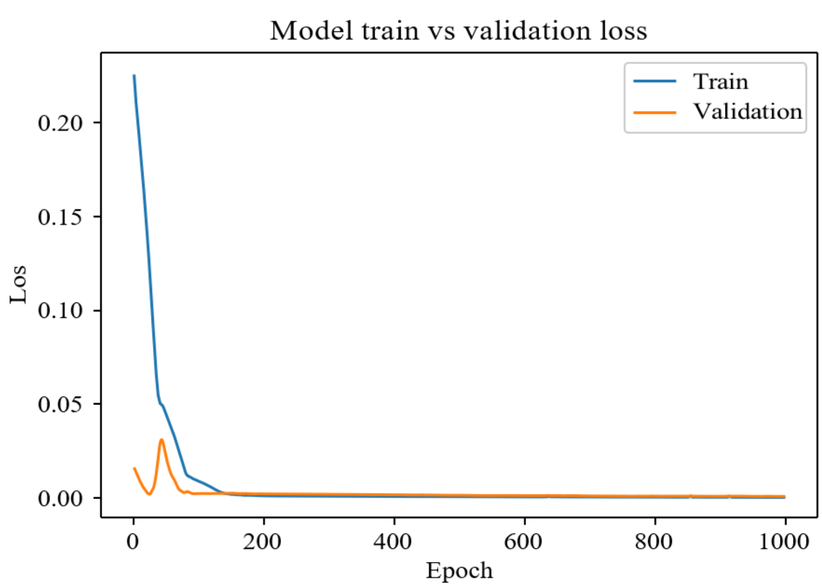


Figure 9: NASDAQ-train vs validation loss.

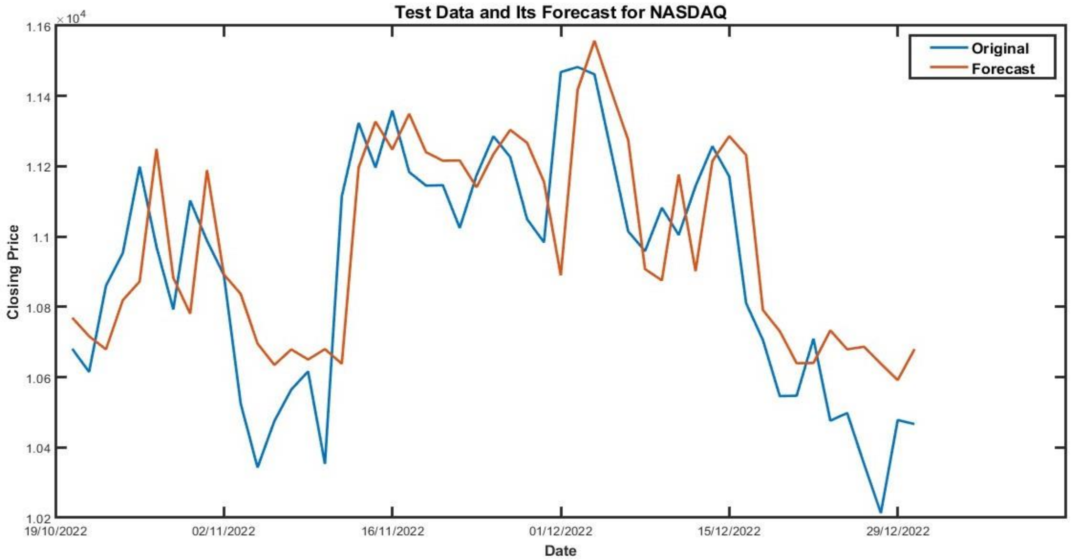


Figure 10: Comparison of forecasts of test data with original data for the stock market index NASDAQ.

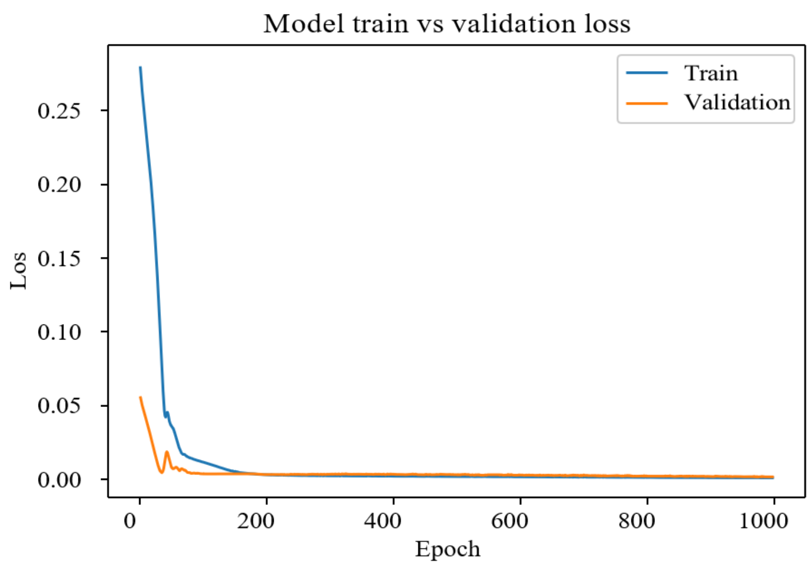


Figure 11: S&P 500-train vs validation loss.

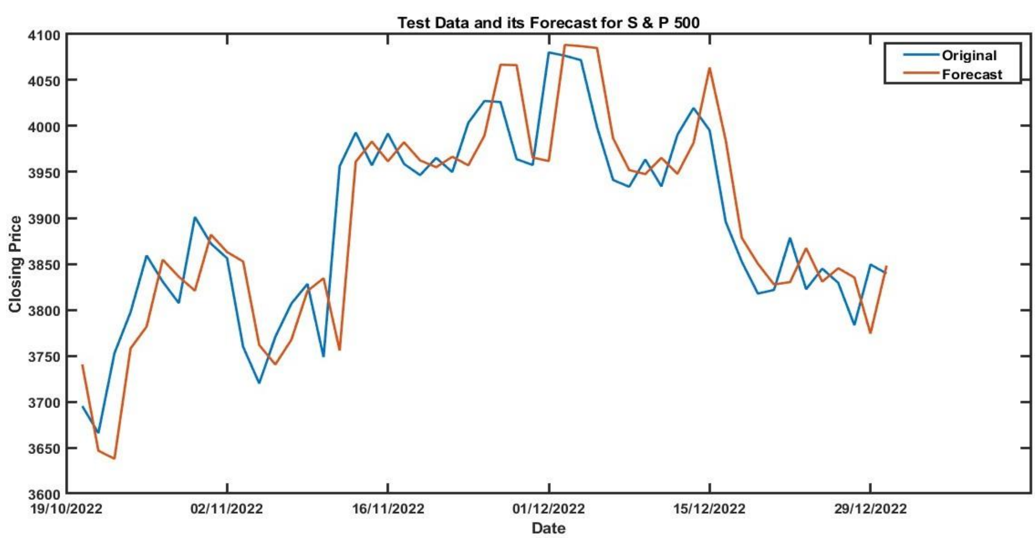


Figure 12: Comparison of forecasts of test data with original data for the stock market index S&P 500.

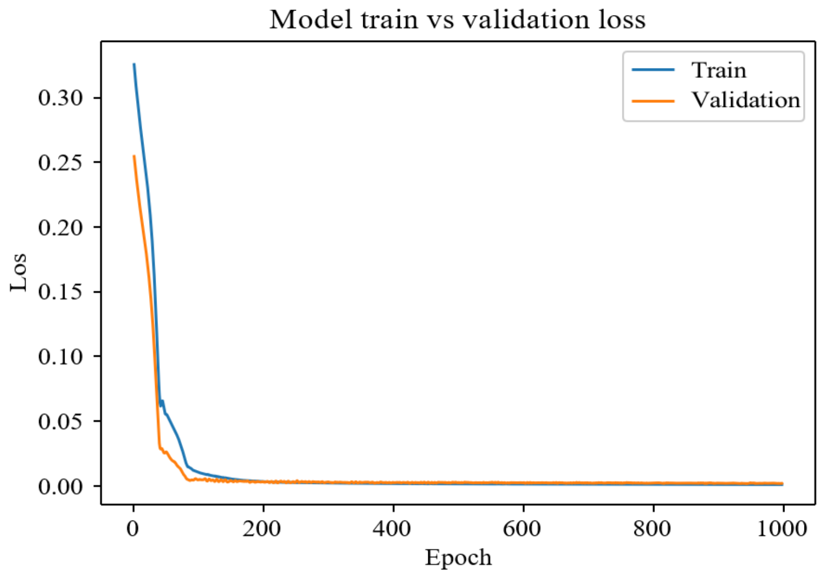


Figure 13: Dow Jones-train vs validation loss.

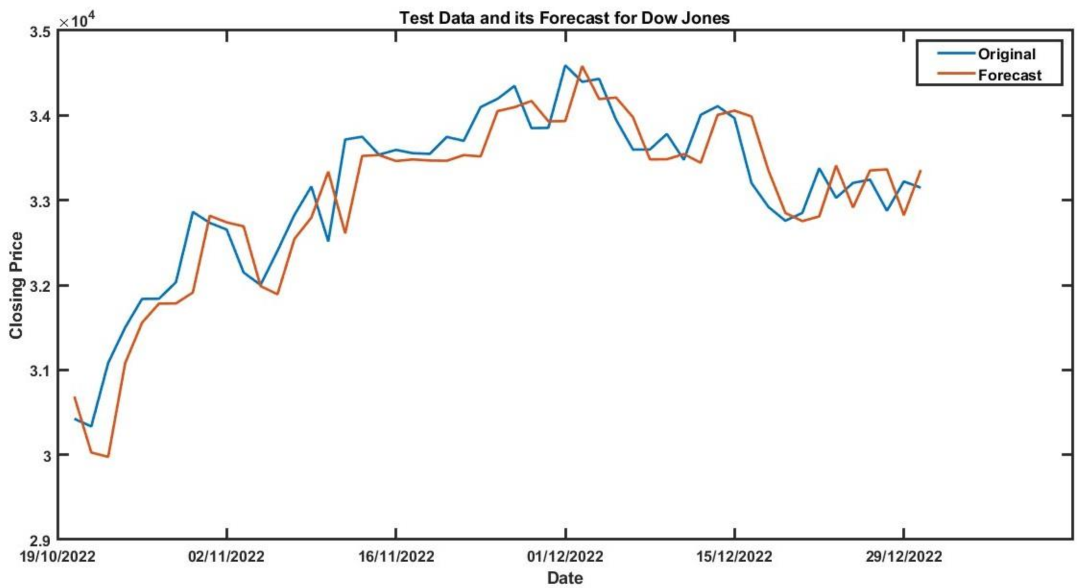


Figure 14: Comparison of forecasts of test data with original data for the stock market index Dow Jones.

5.5. Comparison of the developed VLIFTS Model for the Selected Stock Market Indices with Existing Benchmark FTS Models

The developed VLIFTS model is compared with the benchmark FTS model developed by Chen [3] and the FTS model based on the concepts of Markov chain and transition probability matrix developed by Tsaur [8].

The Root Mean Squared Error (RMSE) and the Mean Absolute Percent Error (MAPE) are the forecast accuracy metrics used in this paper to evaluate the forecast accuracy of the model. The RMSE and MAPE are defined in the following equations:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}{n}} \tag{13}$$

where n is the number of observations in the given data, y_t is the observation at t^{th} time point of a given study variable, and $\hat{y}_t$ is the corresponding forecast at t^{th} time point. A lower RMSE is an indication of a better predictability of a forecasting model. Since RMSE is sensitive to outliers, other forecast accuracy metrics such as mean absolute percentage error is also considered in this thesis to evaluate the predictability of a model.

$$MAPE = \frac{\sum_{t=1}^{n} \left| \frac{y_t - \hat{y}_t}{y_t} \right|}{n} \tag{14}$$

where *n* is the number of observations in the given data, y_t is the observation at t^{th} time point of a given study variable, and $y\hat{}_t$ is the corresponding forecast at t^{th} time point. A lower MAPE is an indication of a better predictive performance of a forecasting model.

Table 3: Estimate of the forecast accuracy metrics RMSE and MAPE for stock market indices Nifty 50, NASDAQ, S&P 500, and Dow Jones.

Time Series	Model	RMSE	MAPE
NSE-Nifty 50	Chen [3]	160.54	1.2507
	Tsaur [8]	125.03	0.9456
	VLIFTS	120.35	0.9117
NASDAQ	Chen [3]	402.98	3.3400
	Tsaur [8]	242.89	1.8721
	VLIFTS	220.74	1.6831
S&P 500	Chen [3]	106.90	2.2302
	Tsaur [8]	58.5987	1.2067
	VLIFTS	57.2725	1.1257
Dow Jones	Chen [3]	551.69	1.2613
	Tsaur [8]	443.41	1.0656
	VLIFTS	420.56	0.9865

The estimate of the forecast accuracy metrics shows that the forecasts due to the developed model called VLIFTS are more accurate in forecasting the time series of the closing price of considered stock market indices Nifty 50, NASADAQ, S & P 500, and Dow Jones. The results are given in Table **3**. Hence the developed VLIFTS model is more efficient than benchmark FTS models.

6. Conclusions

In this paper, the variable-length interval partitioning method is developed to solve the problem of partitioning the universe of discourse of a non-stationary time series into different intervals of varying lengths. This method is based on the LSTM Autoencoder and K-means clustering. LSTM Autoencoder compresses the given data and extracts essential information. The K-means clustering algorithm is applied to this compressed representation and *k* cluster centers are obtained. The intervals are formed using these *k* cluster centers. This hybrid methodology utilizes the unique features of the LSTM network, Autoencoder, and K-means clustering for partitioning UoD into intervals of variable length.

A fuzzy time series model called VLIFTS is developed for modelling and forecasting a non-stationary time series by incorporating this partitioning method for variable-length interval partitioning of the UoD. The developed model incorporates the concepts of the Markov chain and the transition probability matrix in the forecasting stage. The developed model is validated by applying it to the stock market indices NSE-Nifty 50, NASDAQ, S&P 500, and Dow Jones. The estimate of the forecast accuracy metrics shows that the forecasts due to the developed VLIFTS model are more accurate in forecasting the time series of the closing price of stock market indices NSE-Nifty 50, NASDAQ, S&P 500, and Dow Jones. Hence the developed VLFTS model is more efficient than benchmark FTS models. Also, figures of the test data along with its forecast are given for each stock market index.

In the future, it may be focused on extending the architecture of the VLIFTS model to high-order forecasting.

The application of this model can also be extended to other real-world non-stationary time series.

Conflict of Interest

The authors declare that there is no conflict of interest.

Funding

This research work is supported by National Institute of Technology Calicut.

Acknowledgments

Authors are thankful to National Institute of Technology Calicut for its support.

References

[1] Zadeh LA. Fuzzy sets. Inf Control. 1965; 8(3): 338-53. https://doi.org/10.1016/S0019-9958(65)90241-X

[2] Huarng K. Effective lengths of intervals to improve forecasting in fuzzy time series. Fuzzy Sets Syst. 2001; 123(3): 387-94. https://doi.org/10.1016/S0165-0114(00)00057-9

[3] Chen SM. Forecasting enrollments based on fuzzy time series. Fuzzy Sets Syst. 1996; 81(3): 311-9. https://doi.org/10.1016/0165-0114(95)00220-0

[4] Yu HK. Weighted fuzzy time series models for taiex forecasting. Physica A. 2005; 349(3-4): 609-24. https://doi.org/10.1016/j.physa.2004.11.006

[5] Huarng K, Yu TH. Ratio-based lengths of intervals to improve fuzzy time series forecasting. IEEE Trans Cybern. 2006; 36(2): 328-40. https://doi.org/10.1109/TSMCB.2005.857093

[6] Cheng C, Chen T, Teoh HJ, Chiang CH. Fuzzy time-series based on adaptive expectation model for taiex forecasting. Expert Syst Appl. 2008; 34(2): 1126-32. https://doi.org/10.1016/j.eswa.2006.12.021

[7] Yolcu U, Egrioglu E, Uslu VR, Basaran MA, Aladag CH. A new approach for determining the length of intervals for fuzzy time series. Appl Soft Comput. 2009; 9(2): 647-51. https://doi.org/10.1016/j.asoc.2008.09.002

[8] Tsaur RC. A fuzzy time series-markov chain model with an application to forecast the exchange rate between the taiwan and us dollar. Int J Innov Comput Inf Control. 2012; 8(7): 4931-42.

[9] Guney H, Bakir MA, Aladag CH. A novel stochastic seasonal fuzzy time series forecasting model. Int J Fuzzy Syst. 2018; 20: 729-40. https://doi.org/10.1007/s40815-017-0385-z

[10] Song Q. Seasonal forecasting in fuzzy time series. Fuzzy Sets Syst. 1999; 107(2): 235-6. https://doi.org/10.1016/S0165-0114(98)00266-8

[11] Egrioglu E, Aladag CH, Yolcu U, Basaran MA, Uslu VR. A new hybrid approach based on sarima and partial high order bivariate fuzzy time series forecasting model. Expert Syst Appl. 2009; 36(4): 7424-34. https://doi.org/10.1016/j.eswa.2008.09.040

[12] Suhartono, Lee MH, Javedani H. A weighted fuzzy integrated time series for forecasting tourist arrivals. In: Abd Manaf A, Zeki A, Zamani M, Chuprat S, El-Qawasmeh E, Eds. Informatics Engineering and Information Science. ICIEIS 2011. Communications in Computer and Information Science, vol 252. Berlin, Heidelberg: Springer; 2011, pp. 206-17. https://doi.org/10.1007/978-3-642-25453-6_19

[13] Tseng FM, Tzeng GH. A fuzzy seasonal arima model for forecasting. Fuzzy Sets Syst. 2002; 126(3): 367-76. https://doi.org/10.1016/S0165-0114(01)00047-1

[14] Box G. Box and jenkins: time series analysis, forecasting and control. In A Very British Affair: Six Britons and the Development of Time Series Analysis During the 20th Century. London: Palgrave Macmillan, Springer; 2013, pp. 161-215. https://doi.org/10.1057/9781137291264_6

[15] Hao-Tien Liu and Mao-Len Wei. An improved fuzzy forecasting method for seasonal time series. Expert Syst Appl. 2010; 37(9): 6310-8. https://doi.org/10.1016/j.eswa.2010.02.090

[16] Tsymbal A. The problem of concept drift: definitions and related work. Computer Science Department, Trinity College Dublin, 2004; 106(2): 58.

[17] Ditzler G, Roveri M, Alippi C, Polikar R. Learning in nonstationary environments: A survey. IEEE Comput Intell Mag. 2015; 10(4): 12-25. https://doi.org/10.1109/MCI.2015.2471196

[18] Song Q, Chissom BS. Forecasting enrollments with fuzzy time series-part ii. Fuzzy Sets Syst. 1994; 62(1): 1-8. https://doi.org/10.1016/0165-0114(94)90067-1

[19] A J Saleena and C Jessy John. New hybrid model based on triple exponential smoothing and fuzzy time series for forecasting seasonal time series. In: Deo N, Gupta V, Acu A, Agrawal P, Eds. Mathematical analysis ii: optimisation, differential equations and graph theory. ICRAPAM 2018. Springer Proceedings in Mathematics & Statistics, vol 307. Singapore: Springer; 2018, pp. 179-90. https://doi.org/10.1007/978-981-15-1157-8_16

[20] Singh SR. A simple time variant method for fuzzy time series forecasting. Cybern Syst. 2007; 38(3): 305-21. https://doi.org/10.1080/01969720601187354

[21] Wei LY. A hybrid anfis model based on empirical mode decomposition for stock time series forecasting. Appl Soft Comput. 2016; 42: 368-76. https://doi.org/10.1016/j.asoc.2016.01.027

[22] Jilani TA, Burney SMA. A refined fuzzy time series model for stock market forecasting. Physica A. 2008; 387(12): 2857-62. https://doi.org/10.1016/j.physa.2008.01.099

[23] Singh P. A novel hybrid time series forecasting model based on neutrosophic-pso approach. Int J Mach Learn Cybern. 2020; 11(8): 1643-58. https://doi.org/10.1007/s13042-020-01064-z

[24] Singh P, Dhiman G. A hybrid fuzzy time series forecasting model based on granular computing and bio-inspired optimization approaches. J Comput Sci. 2018; 27: 370-85. https://doi.org/10.1016/j.jocs.2018.05.008

[25] Singh P. A brief review of modeling approaches based on fuzzy time series. Int J Mach Learn Cybern. 2017; 8(2): 397-420. https://doi.org/10.1007/s13042-015-0332-y

[26] Bose M, Mali K. Designing fuzzy time series forecasting models: A survey. Int J Approx Reason. 2019; 111: 78-99. https://doi.org/10.1016/j.ijar.2019.05.002

[27] Vovan T. An improved fuzzy time series forecasting model using variations of data. Fuzzy Optim Decis Making. 2019; 18: 151-73. https://doi.org/10.1007/s10700-018-9290-7

[28] Alves MA, Silva PCL, Junior CAS, Vieira GL, Guimarães FG, Sadaei HJ. An extension of nonstationary fuzzy sets to heteroskedastic fuzzy time series. In ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018.

[29] Silva PCL, Junior CAS, Alves MA, Silva R, Cohen MW, Guimarães FG. Forecasting in non-stationary environments with fuzzy time series. Appl Soft Comput. 2020; 97: 106825. https://doi.org/10.1016/j.asoc.2020.106825

[30] Arnold M, Milner XHR, Witte H, Bauer R, Braun C. Adaptive ar modeling of nonstationary time series by means of kalman filtering. IEEE Trans Biomed Eng. 1998; 45(5): 553-62. https://doi.org/10.1109/10.668741

[31] Kitagawa G, Gersch W. A smoothness priors time-varying ar coefficient modeling of nonsta- tionary covariance time series. IEEE Trans Automat Contr. 1985; 30(1): 48-56. https://doi.org/10.1109/TAC.1985.1103788

[32] Bitencourt HV, Souza LAF, Santos MC, Silva R, Lima PC, Guimarães FG, et al. Combining embeddings and fuzzy time series for high-dimensional time series forecasting in internet of energy applications. Energy. 2023; 271: 127072. https://doi.org/10.1016/j.energy.2023.127072

[33] Bitencourt HV, Orang O, Souza LAF, Silva PCL, Guimarães FG. An embedding-based non-stationary fuzzy time series method for multiple output high-dimensional multivariate time series forecasting in iot applications. Neural Comput Appl. 2023; 35(13): 9407-20. https://doi.org/10.1007/s00521-022-08120-5

[34] Pattanayak RM, Behera HS, Panigrahi S. A novel probabilistic intuitionistic fuzzy set based model for high order fuzzy time series forecasting. Eng Appl Artif Intell. 2021; 99: 104136. https://doi.org/10.1016/j.engappai.2020.104136

[35] Dixit A, Jain S. Intuitionistic fuzzy time series forecasting method for non-stationary time series data with suitable number of clusters and different window size for fuzzy rule generation. Inf Sci. 2023; 623: 132-45. https://doi.org/10.1016/j.ins.2022.12.015

[36] Yolcu OC, Yolcu U. A novel intuitionistic fuzzy time series prediction model with cascaded structure for financial time series. Expert Syst Appl. 2023; 215: 119336. https://doi.org/10.1016/j.eswa.2022.119336

[37] Kocak C, Egrioglu E, Bas E. A new explainable robust high-order intuitionistic fuzzy time-series method. Soft Comput. 2023; 27(3):1783-96. https://doi.org/10.1007/s00500-021-06079-4

[38] Radha Mohan Pattanayak, Himansu Sekhar Behera, and Sibarama Panigrahi. A novel high order hesitant fuzzy time series forecasting by using mean aggregated membership value with support vector machine. Inf Sci. 2023; 626: 494-523. https://doi.org/10.1016/j.ins.2023.01.075

[39] Pant M, Kumar S. Fuzzy time series forecasting based on hesitant fuzzy sets, particle swarm optimization and support vector machine-based hybrid method. Granular Comput. 2022; 7(4): 861-79. https://doi.org/10.1007/s41066-021-00300-3

[40] Farid M. The subgradient extragradient method for solving mixed equilibrium problems and fixed point problems in hilbert spaces. J Appl Numer Optim. 2019; 1(3): 335-45. https://doi.org/10.23952/jano.1.2019.3.10

[41] Farid M, Kazmi KR. A new mapping for finding a common solution of split generalized equilibrium problem, variational inequality problem and fixed point problem. Korean J Math. 2019; 27(2): 297-327.

[42] Djafari-Rouhani B, Kazmi KR, Farid M. Common solutions to some systems of variational inequalities and fixed point problems. Fixed Point Theory. 2017; 18(1): 167-90. https://doi.org/10.24193/fpt-ro.2017.1.14

[43] Alansari M, Farid M, Ali R. An iterative scheme for split monotone variational inclusion, variational inequality and fixed point problems. Adv Differ Equ. 2020: 485: 1-21. https://doi.org/10.1186/s13662-020-02942-0

[44] Zadeh LA. Outline of a new approach to the analysis of complex systems and decision processes IEEE Trans Cybern. 1973; (1): 28-44. https://doi.org/10.1109/TSMC.1973.5408575

[45] Zadeh LA. The concept of a linguistic variable and its application to approximate reasoning-i. Infr Sci. 1975; 8(3): 199-249. https://doi.org/10.1016/0020-0255(75)90036-5

[46] Song Q, Chissom BS. Fuzzy time series and its models. Fuzzy Sets Syst., 1993; 54(3): 269-77. https://doi.org/10.1016/0165-0114(93)90372-O

[47] Song Q, Chissom BS. Forecasting enrollments with fuzzy time series-part i. Fuzzy Sets Syst. 1993; 54(1): 1-9. https://doi.org/10.1016/0165-0114(93)90355-L

[48] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997; 9(8): 1735-80. https://doi.org/10.1162/neco.1997.9.8.1735

[49] MacQueen J. Some Methods for Classification and Analysis of Multivariate Observations. Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, Oakland, CA, USA: 1967, pp. 281-97.

[50] Chen SM, Tanuwijaya K. Fuzzy forecasting based on high-order fuzzy logical relationships and automatic clustering techniques. Expert Syst Appl. 2011; 38(12): 15425-37. https://doi.org/10.1016/j.eswa.2011.06.019