SCTE CABLE-TEC EXPO FAST FORWARD 2021
ATLANTA, GA
OCTOBER 11-14
2021 Fall Technical Forum
SCTE • NCTA • CABLELABS

Cluster-Based Network Traffic Prediction Pipeline For Big Data Time Series

A Technical Paper prepared for SCTE by

Wei Cai
Network Planning Engineer IV
Cox Communication
6205-B Peachtree Dunwoody Rd, Atlanta, GA 30328
678-645-0000
wei.cai@cox.com

Table of Contents

Title **Page Number**

List of Figures

Title **Page Number**

List of Tables

Title **Page Number**

1. Introduction

Upstream broadband usage and network capacity have been increasing sharply in recent years, particularly under global pandemic lockdowns since March 2020 [NCTA/covid-19-overview]. The spread of COVID-19 around the entire world has placed upstream traffic growth on an extremely irregular pattern with fluctuations, posing great challenges to use conventional methods such as Auto-Regressive Integrated Moving Average (ARIMA) and other classical statical models to ensure accurate forecasts because those classical methods have been proven to be weak and inadequate for modeling non-stationary traffic flows.

In practical forecasting applications, the use of different modeling methods has become a popular research by many scholars [Chen, 2011; He and Zeng, 2021]. However, with the rapid development of cable companies' network, network traffic data scale is increasingly important in modern network traffic world. Just combining forecasts from different models with weighs allocated does not satisfy the need to model big traffic flow data more effectively.

In this study, we propose a clustering-based two-stage approach to build network traffic forecasting models using ARIMA, eXtreme Gradient Boosting (XGBoost) and Long Short-term Memory (LSTM) that includes data preprocessing and model building, as shown in Fig. 1.

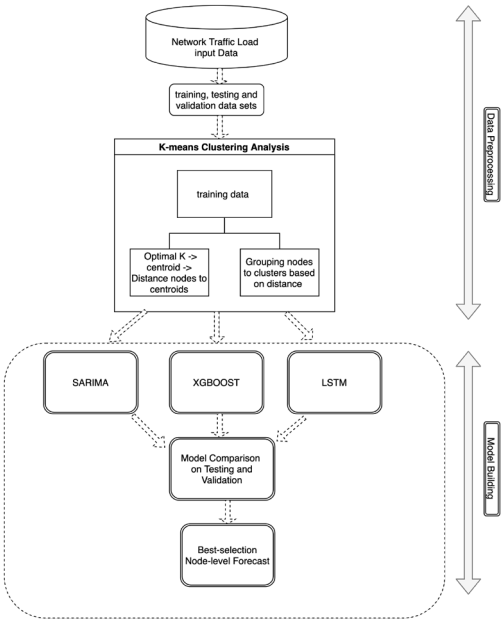

Figure 1 - The flowchart of the proposed best-selection forecasting

In the data preprocessing stage, using the training subset, we apply the K-means clustering method to find the K cluster centers to group a collection of network nodes into homogenous subsets based on intra-cluster similarity in traffic data [Vujicic et al., 2006]. K-means clustering is one of the most-commonly used data clustering algorithms, originally from signal processing. It aims to partition n observations into k clusters in which each observation belongs to the cluster with nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster [Anderberg, 1973, Kaufman and Rousseeuw, 1990]. A key assumption of applying forecasting models on clusters is the similarity of behavior patterns within a cluster. This algorithm is helpful to obtain computational efficiency when it comes to big data time series forecasting.

The paper is structured as follows: Section 2 explores the related studies and research gaps in existing literatures. Section 3 describes the methodology. Section 4 discusses the experimental results and Section 5 concludes the paper.

2. Background

Network traffic-related data are collected as time-series, hence time series analysis and forecasting techniques such as ARIMA modelling, XGBoost and LSTM can be employed for the traffic forecasting. In this section, we first review three modeling techniques applied: ARIMA, XGBoost and the deep learning-based Long Short-Term Memory (LSTM) model, then followed by review of related work.

2.1. Learning methods

2.1.1. ARIMA

The ARIMA model was pioneered by Box and Jenkins [Pankratz, 2008]. It is one of the most classical statistical models in time series prediction. This model can be presented as a linear regression function, in which lags and the lagged forecast errors are used for prediction. A standard ARIMA model can be expressed by three terms: autoregressive order (AR, p), moving average part (MA, q), and difference order (differencing, d). A seasonal ARIMA is an expanded version of a standard ARIMA model with information extracted from the seasonal parts that cannot be processed by the standard ARIMA model. To build a seasonal ARIMA model, finding the non-seasonal part (p, d, q) of ARIMA is the first step, and tuning the seasonality components (P, D, Q) is the second. Diagnostic checking of stationary and residual uncorrelation is vital for a good fit model. A simple ARIMA structure is illustrated in Figure 2 as follows:

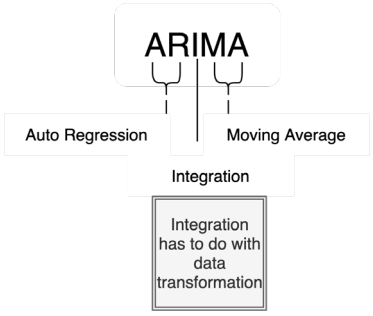

Figure 2 – ARIMA Structure

2.1.2. XGBoost

Extreme Gradient Boosting as one of the boosting algorithms in ensemble learning was first proposed by Tianqi Chen in 2015 and Carlos Guestrin in 2011. It is proved in the literature [Chen and Guestrin, 2016] that the XBGoost model has the characteristics of being efficient and scalable [Saeed, 2016]. The basic idea of the Boosting algorithm is to combine many weak learners to form a strong model that can predict accurately. Extreme gradient lifting tree [Wang et al., 2019] essentially is an integrated learning algorithm in which the number of decision tree keeps being added with each iteration till a strong classifier is found [Fabricius, 2000]. The XGBoost model has proved to have many advantages in model prediction, such as the lack of a need to preprocess the data, a fast operation speed, complete feature extraction, a good fitting effect and high prediction accuracy [Alim et al., 2020]. A simple XGBoost structure is illustrated in Figure 3 as follows:

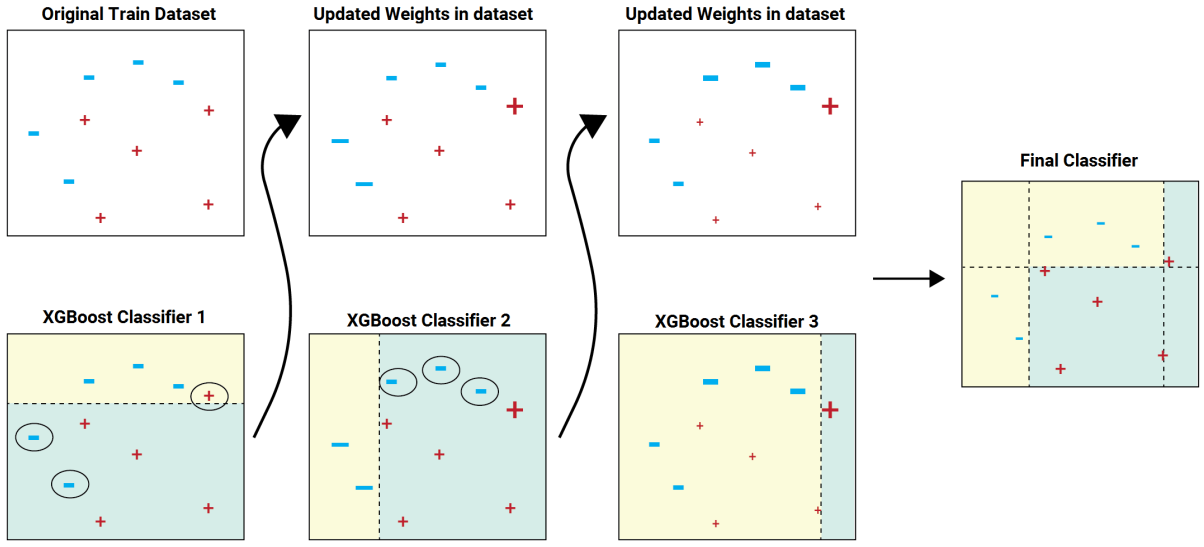

Figure 3 - XGBoost structure (https://blog.quantinsti.com/xgboost-python/)

2.1.3. LSTM

Long short-term memory networks (LSTM), proposed by (Hochreiter & Schmidhuber, 1997) is a special kind of recurrent neural network (RNN). Different from other RNN models that could easily suffer from the problems of getting small gradient loss (i.e., less than one), and multiplication of those gradient losses would vanish during training process, LSTM overcomes these problems by memorizing the prior stages' internal trend through a few different gates (i.e., input gate, forget gate, control gate and output gate) and optionally let data pass through or dispose of based on a sigmoidal neural network layer to predict future patterns. A simple LSTM network is illustrated in Figure 4 as follows:

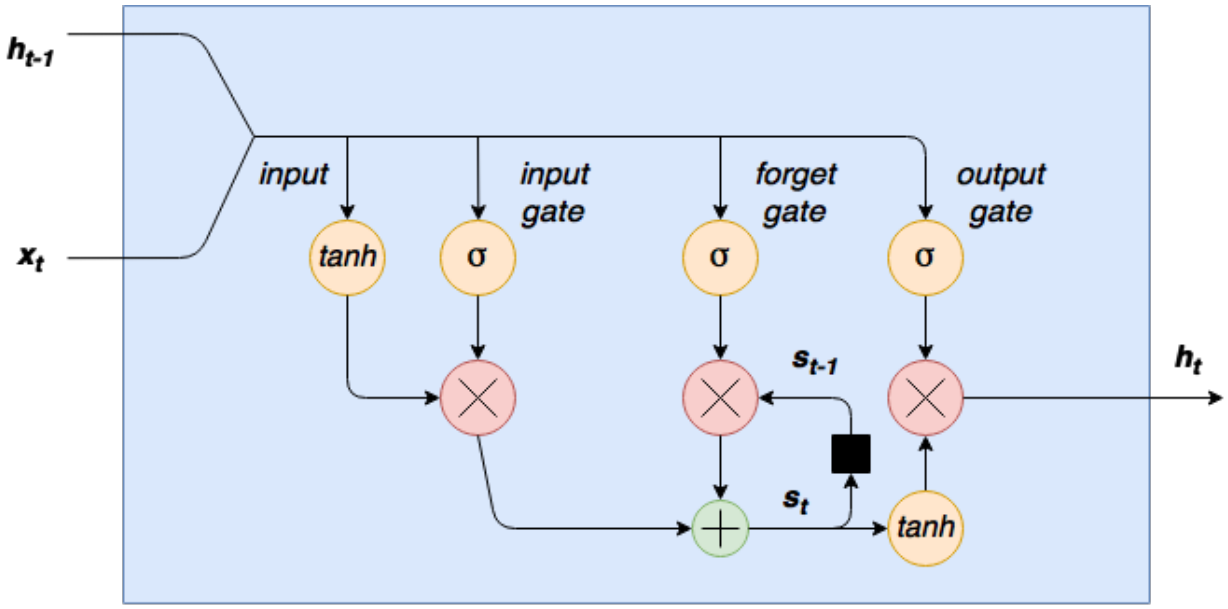

Figure 4 - LSTM structure (https://adventuresinmachinelearning.com/keras-lstm-tutorial/)

2.2. Related work

There are several works that have focused on comparing the performance of different forecasting techniques. He and Zeng (2021) have put forward a forecasting method of XGBoost-LSTM combination model based on a weighting method in forecasting product sales. The two years' sales time series data are modeled by using XGBoost and LSTM neural network models, respectively. They prove that the performance of using XGBoost-LSTM model to deal with time series is much higher than that of the original XGBoost single model, which maximizes the advantages of the two prediction models. Chen (2011) proposed to use the combination models that combine the linear model and the nonlinear model to predict tourism demand time series data. The results showed that the combination is superior to the individual models for the test cases of tourism demand time series. Alim et al. (2020) compared the performance of the XGBoost model a seasonal ARIMA model for human brucellosis in mainland China and used them to make short-term predictions. The results showed that the prediction accuracy of the XGBoost model was much better than that of the ARIMA model.

2.3. Research gap

From the above-mentioned research works, we have identified a couple of major research gaps. To the best of our knowledge, first major research gap is that there are very few research papers that have employed a clustering approach to perform classification of large quantities of network traffic data, grid search parameters or train model within a cluster. Clustering-based modeling approach can be very effective in terms of reducing computational cost for big data but still obtain decent prediction accuracy. Researchers in one study prove training the classical SARIMA models on clusters of public safety network users identified by the K-means algorithms performs well compared to the prediction based on the overall aggregate traffic [Vujicic et al., 2006]. With the data we have, at an individual node level, time series are very volatile, include a high amount of noise, and are unevenly and nonlinearly affected by different effects. However, most nodes in this study follow similar pattern and many series move in tandem, suggesting possibility to group those nodes into homogeneous clusters for hyperparameter search and model training.

Secondly, different from many relevant research studies recommending using combination of forecasting model: selecting appropriate weights to weight and average the results obtained from several different model, we propose to build different models separately and let three models compete to always select the best individual-level forecast in practical forecasting application. This will provide more reliable traffic prediction to the network management and the network capacity planning.

3. Methodology

11,738 network nodes' weekly upstream traffic load time series were selected into input for this study to test the performance of three models: ARIMA, XGBoost and LSTM. After replacing missing values with linear interpolation for each time series, 174 weekly data polls between 2018 and May 2021 were collected as the input data. During model development, 70% of the input data were used to train models, 20% is used as the testing data set, and the remaining 10% is used as the verification data set.

3.1. Data description and Pre-processing

The upstream traffic dataset used in the study was obtained from a cable company's network. It contains information regarding *site id*, *node description*, and *traffic loads* in weekly intervals, gathered from 11, 738 nodes. *Site id* and *node description* were concatenated to a string to label each network node in this study.

The significant problem forecasting is facing is the presence of missing values and outlier values in the observed historical traffic load data. In the dataset we collected, there exist missing values in weekly timestamp and traffic loads due to occurrences of data collection failures. Therefore, those missing values were linear interpolated based on historical trend. Linear interpolation is illustrated in Figure 5. In Figure 5, the left plot shows raw data weekly trend for one sample node but with missing polls highlighted with yellow arrows. After interpolation, those missing polls were filled with imputed values which can be seen on the right plot.

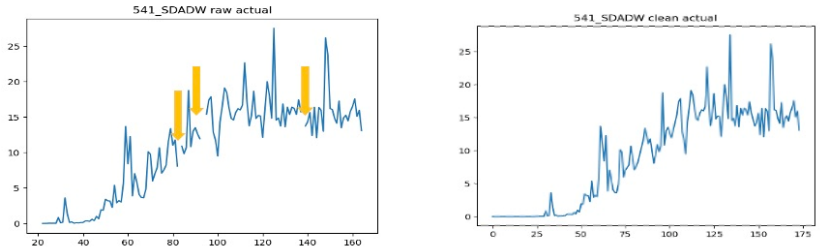

Figure 5 - Missing value linear interpolation

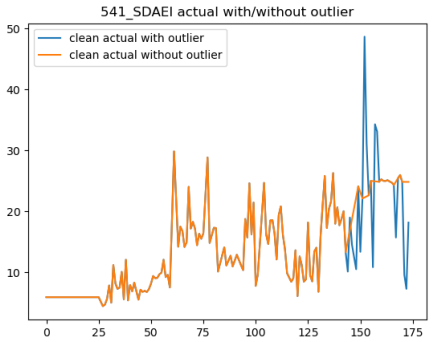

Figure 6 - Data with/without outliers

Outliers in time series can be grouped into two types: outliers affecting a single observation and outliers affecting a single observation and subsequent observations, which is true in our input data. In this study, we use a combination of statistical method to detect those anomalies and then apply linear interpolation based on time to replace them with imputed values. The effect is depicted in Figure 6.

3.2. Forecasting model set-up

Three different models based on the same input are proposed and fitted to predict future traffic loads. 70%-20%-10% of data partition is used for training, testing and validation purposes. The performance of all three models is presented based on the criteria of Mean Absolute Percentage Error (MAPE) as shown below:

$$M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \quad (1)$$

where A_t is the actual value, F_t is the forecast value mean, n is the number of times the summation iteration happens. M stands for mean absolute percentage error used to evaluate prediction accuracy and is expressed in percentage. In this study, a value lower than 10 is said to be a fit model. The evaluation of all three models is done by preparing a model on a training dataset and by making predictions on a test dataset and a validation dataset. Training, test and validation MAPEs are calculated to measure model performance.

Before training each model, K-means clustering method was used to partition 11,738 nodes into 4 optimal clusters. In this study, we use Euclidean as a distance metric because we have normalized each time series to have the same length for all nodes selected after preprocessing. the node count distribution by cluster is summarized as follows: Cluster 0: 10,082 nodes; Cluster 1: 392 nodes; Cluster 2: 71 nodes; Cluster 3: 1,193 nodes. To intuitively observe the data characteristics of each cluster's traffic load, the sum of traffic loads by each cluster is plotted in the following figure:

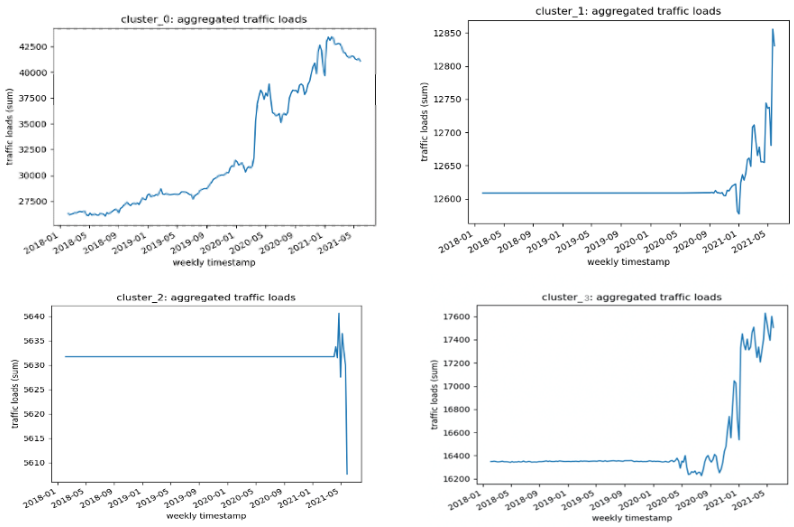

Figure 7 - Upstream Traffic Loads Trend by Cluster

As we can see form Figure 7, Cluster 0 shows much more volatilities and fluctuations since the start of COVID-19 compared to three other clusters. Clusters 1 and 3 experience more seasonal effect compounded with COVID-19 since late 2020. Cluster 2 has a relatively more peaceful trend compared to three others.

After K-means clustering, the main steps followed for the ARIMA model fitting as follows:

- Grid search was used to automatically discover the optimal order of non-seasonal and seasonable parameters at a cluster level. The combination of (p, d, q) that returns the lowest Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) values are selected as the most optimal parameters for each cluster.
- Dataset is split into 70%, 20%, 10% training, test and validation sets, respectively.
- Stationarity of the series are checked using the Adfuller function with the P-value along with the autocorrelation function (ACF) and the partial autocorrelation (PACF) plots. Training data are transformed based on the stationarity check.
- SARIMAX models are trained and obtained to make estimation on fresh test data.
- To check the goodness of fit of the ARIMA model, MAPE values for both training dataset and testing data set are calculated.

After K-means clustering, the main steps followed for the XGBoost model fitting as follows:

- To obtain the best performance, the grid search algorithm is used to optimize the parameters: min_child_weight, gamma, subsample, colsample_bytree, max_depth and learning rate at a cluster level.
- Dataset is split into 70%, 20%, 10% training, test and validation sets, respectively.
- XBGoost models are trained and obtained to make estimation on fresh test data.
- To check the goodness of fit of the XGBoost model, MAPE values for both training dataset and testing data set are calculated.

After K-means clustering, the main steps followed for the LSTM model fitting as follows:

- Scale data using MinMaxScaler to speed up the learning process and help model.

- Hyperparameter such as number of layers, layer depths, activation functions, dropout coefficients are repeatedly tuned at a cluster. Manual tuning of LSTM hyperparameter is used instead of using automated tuning package such as SMAC3 to avoid common failure to find well-defined global minima with automated packages.
- Dataset is split into 70%, 20%, 10% training, test and validation sets, respectively.
- LSTM models are trained and obtained to make estimation on fresh test data. During the model training, the validation loss was monitored by early stopping call back function to halt the training if there is an increment observed in loss values. The number of epochs for the training was tested with different values.
- To check the goodness of fit of the LSTM model, MAPE values for both training dataset and testing data set are calculated.

4. Results and Discussion

In order to see the prediction effect of all three selected models, training, test and validation MAPEs are matched on 11,542 nodes for comparison. All three groups of MAPEs were classified into four groups: <= 5%, between 6 and 10%, between 11% and 15% and greater than 16%. Number of nodes were summarized for each MAPE group.

Tables 1 and 2 respectively present the comparison of the training and test prediction accuracy for three selected models.

Table 1 - Training MAPE by Models

Model	Training MAPE				Total
MAPE Range	<= 5%	>=6% & <= 10%	>=11% & <= 15%	>=16%	
SARIMA	1,360	2,828	2,059	5,295	11,542 nodes
XGBoost	11,333	55	20	134	11,542 nodes
LSTM	3,573	2,979	2,909	2,081	11,542 nodes

From Table 1, for the training set, the XGBoost model has the highest number of nodes with the MAPE value less than or equal to 10% than two other models.

Table 2 - Test MAPE by Models

Model	Test MAPE				Total
MAPE Range	<= 5%	>=6% & <= 10%	>=11% & <= 15%	>=16%	
SARIMA	982	746	1,175	8,639	11,542 nodes
XGBoost	2,244	2,228	2,870	4,200	11,542 nodes
LSTM	2,299	3,996	2,753	2,494	11,542 nodes

In Table 2, for the test set, the LSTM model has a higher number of nodes with the MAPE value less than or equal to 10% than two other models. Majority of the nodes with the LSTM showed the MAPE value less

than 15%, which shows the potential benefit of using the LSTM to predict network traffic. The ARIMA has the lowest number of nodes with a good fit, which might indicate the weakness of using conventional statistical models to predict for non-linear data.

To further evaluate model performance of all three selected models, we chose nodes with both training MAPE and test MAPE less than 10% to compare training MAPEs with test MAPEs. If the test MAPE is lower than training MAPE, we consider that node's forecast is a good fit. Vice versa, if the test MAPE is higher than training MAPE, we label that mode as overfitting training data but does not perform well on the evaluation data. Node counts for good fit and overfitting by different models are summarized in Table 3:

Table 3 - Node Counts with <= 10% MAPE by Models

Model	Node counts with good fit (Mape <= 10%)	Node counts with overfit (Mape <= 10%)
SARIMA	956	586
XGBoost	1,195	3,248
LSTM	2,129	2,278

As shown in Table 3, it is clearer that LSTM overall has less overfitting issues in network traffic forecasting than two other models. It further indicates that LSTM in this paper has higher fitness and predictive performance in network traffic prediction compared to ARIMA and XGBoost. XGBoost model has the most overfitting issues in this study.

To map model performance for the validation dataset, a comparison of 596 nodes validation MAPEs is shown in Table 4.

Table 4 - Node Counts MAPE by Models

Model	Node counts with Mape <= 10%)	Node counts with Mape >10%)
SARIMA	101	495
XGBoost	139	457
LSTM	346	250

Table 4 shows that LSTM has the most of number of nodes with validation MAPE less than 10%, followed by XGBoost and SARIMA.

According to the overall results, the performances of the ARIMA models were the lowest and the LSTM models performed the best. LSTM models produced the best results.

5. Conclusion

The author proposed to use K-means analysis to partition network traffic nodes to different cluster and apply seasonal ARIMA model (SARIMA), XGBoost model, and the Long Short-Term Memory (LSTM) model on clusters to reduce computational cost and predict effectively.

Three selected models are compared with respect to performance. According to the comparison, the performance of the LSTM network is better than ARIMA and XGB model. XBGoost model shows a reasonable performance but showed serious overfitting issues. Moreover, the classical data analysis model ARIMA has more obvious forecast error, which shows the disadvantage of classical parameterized approach faced with tremendous traffic data.

As for future work, the SARIMA model could be improved using one-step ahead forecast method. At the same time, LSTM models can also be improved by hyperparameter tuning such as the number of layers, learning rate, optimizers, etc. Overfitting with the XGBoost model could be avoided using the delta difference between time interval (i.e., lag term of the time series) as the input and let the input lag term predict the univariable time series. L1 and L2 regularisation can be introduced to the LSTM and XGBoost to address overfitting. Cross-validation can be tried with all three selected models to obtain more reliable forecasting models.

Abbreviations

ARIMA	Auto-Regressive Integrated Moving
SARIMA	Seasonal Auto-Regressive Integrated Moving
XGBoost	Extreme Gradient Boosting
LSTM	Long Short Term Memory
AR	Autoregressive Order
MV	Moving Average
RNN	Recurrent Neural Network
MAPE	Mean Absolute Percentage Error
AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
ACF	autocorrelation function
PACF	partial autocorrelation

Bibliography & References

Alim, Mirxat, Ye, Guo-Hua, Guan, Peng, et al. *Comparison of ARIMA Model and XGBoost Model for Prediction of Human Brucellosis in Mainland China: a Time-series Study*, England: BMJ Publishing Group LTD BMJ open, 2020-12-07, Vol.10 (12), p.e039676-e039676

Anderberg, M. R., 1973. Cluster Analysis for Application. Acdemic, New York

Babajide Mustapha I, Saeed F. *Bioactive Molecule Prediction Using Extreme Gradient Boosting Molecules* 2016;21. doi:10.3

Chen T, Guestrin C. *XGBoost: A Scalable Tree Boosting System* CoRR abs, 2016, pp. 1603-02754.

De'Ath G, Fabricius K E. *Classification and Regression Trees: a Powerful Yet Simple Technique for Ecological Data Analysis Ecology*, vol. 81, no. 11, 2000, pp. 3178-3192.

He, Wei; Zeng, QingTao. *Research on sales Forecast based on XGBoost-LSTM algorithm Model* Journal of Physics: Conference Series; Bristol Vol. 1754, Iss. 1, (Feb 2021). DOI:10.1088/1742-6596/1754/1/012191

Hochreiter, S., Schmidhuber, Jürgen (1997). *Long short-term memory* Neural Computation, 9(8), 1735–1780.

Kaufman, L., Rousseeuw, P. J. *Finding Groups in Data: An Introduction to Cluster Analysis* New York, NY: Wiley-Interscience, 1990

NCTA/covid-19-overview: https://www.ncta.com/whats-new/ncta-launches-covid-19-internet-dashboard

Pang L,Wang J, Zhao L, et al. *A novel protein subcellular localization method with CNN-XGBoost model for Alzheimer's disease*. Frontiers in Genetics (S1664-8021), 2019, 9:751.

Vujicic, B., Chen, H., Trajkovic, L. *Prediction of traffic in a public safety network* June 2006, SourceIEEE Xplore Conference: Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on Project: Collection, Characterization, and Modeling of Network Traffic

