A Project Report
on

Train Delay Analysis and Prediction Methodology for Indian Railways

by

Ramesht Shukla & **Aditi Singhal**
Final Year, B.Tech. (Electronics)
Bharati Vidyapeeth (Deemed to be University),
College of Engineering, Pune

Under Kind Guidance
of

10.09.2020

Mr. V.P. Kaduskar
Bharati Vidyapeeth,
(Deemed to be University),
College of Engineering, Pune,

Mr. K.K. Chowdhury
Government of India, Ministry of railways,
Research Designs & Standards Organisation,
Manak Nagar, Lucknow.

June-2020

CONTENTS

ABSTRACT

Despite the fact that punctuality is an advantage of rail travel compared with other long-distance transport, train delays often occur. For this study, a dataset of weather, train delay and train schedule records were collected and analyzed in order to understand the patterns of train delays and to predict train delay time. We found that in severe weather train delays are determined mainly by the type of bad weather, while in ordinary weather the delays are determined mainly by the historical delay time and delay frequency of trains. Identifying the factors closely correlated with train delays, we developed a machine-learning model to predict the delay time of each train at each station. The proposed prediction model is useful not only for passengers wishing to plan their journeys more reliably, but also for railway operators developing more efficient train schedules and more reasonable pricing plans.

Abbreviations and Acronyms	
CAG	COMPTROLLER AND AUDITOR GENERAL
TDPS	TRAIN DELAY PREDICTION SYSTEM
ARIMA	AUTO REGRESSIVE INTEGRATED MOVING AVERAGE
KNN	K-NEAREST NEIGHNBOURS
ST-ARIMA	SPACE TIME ARIMA

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Background

Indian railways face large development in the recent years. Their performance is still confronted with disruptions, which impact the traffic and the passengers. Train delays are the major factors affecting for this disruption. In 2017-18, Indian Railways had the worst punctuality performance in three years. 30 per cent trains ran late in 2017-18, according to official data. From April 2017-March 2018, the punctuality of mail and express trains was 71.39 per cent, down from 76.69 per cent in 2016-17, which is a deterioration of 5.30 per cent. When Comptroller and Auditor General (CAG) of India looked into the reasons for this delay, it found some serious flaws in government's Rs 1 lakh crore station redevelopment plans. There isn't enough space for trains on stations which is the main cause of delays, and little is being done to change that. In a latest report tabled in Parliament, CAG said, "The station development/redevelopment plans mainly address on facilities for the passengers on the station premises and facade of stations only and not on removing constraints and bottlenecks for ensuring timely arrival and departure of trains to/from the stations, which should be one of the most important parameters of the quality of service being provided to the passengers.".

CAG further explains how the unavailability of path (platform/line) for accommodating trains and absence of enough platforms with sufficient length to handle trains with 24 or more coaches are main reasons for delay of trains. CAG selected 15 stations in 10 zonal railways falling on the routes with heavy passenger traffic for audit. It analyzed one-month data (March 2017) to study how deficient infrastructure leads to detention of trains at adjoining stations, en-route and on platforms. Only 100 platforms out of 164 have the capacity to handle trains with 24 or more coaches, it found.

Weather conditions are also the major cause of train delays. In hot summer days trains are delayed due to overheating or melting of parts and on the other hand in winters the train are delayed due to extreme cold conditions in some areas and due to the presence of fog. These factors that are discussed above, can be said as the key factors affecting the train schedule of trains.

1.2 Literature Review

Train delay prediction systems have been used since a long time. These systems are static and are operated with human assistance. Most of these systems do not consider historical train journey data collected from various information sources. As a result, relying on the current running train data is prone to inaccurate predictions. Most of us have already acknowledge this while travelling to different places by train. Moreover, certain setbacks have also been associated with the system, and many researchers and companies are working dynamically to make it error proof and precise.

The railway is a major means of long-distance transport in India. With more developed railway infrastructure, highspeed trains and railway management, passenger rail transport is faster and more convenient than it used to be. Although trains are still slower than airplanes, the punctuality, convenience and short waiting time of rail transport make it a competitive mode of transport for middle-distance travel. Despite the fact that rail transport has several advantages, occasional train delays still occur due to factors ranging from severe weather to equipment failure and poor management. Although train delays are usually caused by train speed restrictions imposed for safety reasons, the delays can disrupt connecting journeys (such as international flights) or important activities (such as important tasks or conferences). It is therefore important for passengers to be able to anticipate delays and adjust their travel arrangements in advance to avoid missing important connecting journeys and activities. In addition, understanding the mechanisms of train delays and being able to predict them assists railway operators in improving their train management plans. The analysis of delay times for different trains can also help railway operators to take the punctuality of each train into account and to develop more reasonable pricing plans.

The prediction of train delays can be divided into short-term and long-term prediction. Short-term prediction of train delays is usually estimated from real-time operating data and released on the official websites of railway. Long-term prediction of train delays, which is the focus of the present study, predicts delays a relatively long period (e.g. 3 days or 1 week) in advance. In comparison with short-term prediction, long-term prediction is more useful for passengers in planning and adjusting their trips, because a passenger is usually unable to change their travel plans by the time short-term delay information is released, and therefore has no choice but to wait for the delayed train. Moreover, long-term prediction of train delays can provide railway operators with more time to improve their train management plans. In the present study, we analyze the factors affecting train delays and propose a machine learning-based train delay model using three sets of big data. First, however, it is worth reviewing previous work in three related areas: factors affecting train delays; the spread of train delays; and train delay prediction.

Magadalega et al. **Error! Bookmark not defined.** and Olsson et al. **Error! Bookmark not defined.**discussed the main factors causing train delays. These factors include natural phenomena (such as changes of weather and natural disasters), human factors (such as improper operation) and systemic factors (including signal communication failure, cable failure and power outages). Wei et al. **Error! Bookmark not defined.**developed the switching max-plus system (SMPS) model and the all-paired critical-path (APCP) algorithm to study the impact of bad weather on train delays. Cerreto et al. [4] studied train delay data from the Danish railway network. The authors discovered that the train infrastructure layout, the performance and reliability of train vehicles and the train stopping time at stations will affect the train delay time, the minimum running time and the buffer time of trains. Lalinska et al. [5] and Jiang et al. [6] developed a subway train delay simulation model based on the stochastic characteristics of the delays of subway trains Ludvigsen et al. [8] studied the Finnish railway system. The authors analyzed the correlation between severe weather and equipment failure and proposed a train delay analysis model. Wang [9] proposed an intelligent fault analysis system for railway signal equipment based on computer monitoring. The author claimed that the system is effective at preventing train delays and accidents. Dingler et al. [10] used a railway traffic controller (RTC) to classify and quantify train delays and identify the factors causing train delays. The relationships among speed, siding dwell time and train delay were discussed. Methods of increasing train operation speed and the number of tracks were proposed to reduce train delay time. Weng et al. [11] analyzed equipment factors such as power failure, braking system failure and signal failure on the Hong Kong subway. The authors also set out a train delay prediction model.

Other researchers have investigated the spread of train delays. Berger et al. [12] proposed a stochastic model to simulate train delay propagation and predict the arrival and departure times of trains. The model was put into practice on the German rail network. Huisman et al. [13, 14] studied the delay caused by the differences in train speeds. The authors applied queuing theory to simulate the train delay propagation process and the train operation process. Rob et al. [15, 16] used the data mining tool TNV-Conflict (telecommunication network voltage-conflict) and the analysis tool TNV-Statistics to identify the path conflict chain and determine the signal arrangement according to the number of collisions, time loss and delay jump. A model and an algorithm for calculating the initial delay propagation on the periodic railway timetable were proposed. Schlake et al. [17] used scheduling simulation software to analyze delays on monorail and dual rail lines. The results showed that traffic volume and train condition have significant impacts on train delays, and delayed trains have different degrees of propagation impact on subsequent trains.

The long-term train delay prediction model is developed based on advanced weather-forecasting techniques, the close link between severe weather and train delays, and the relatively consistent rules of train operation (e.g. low-rank trains giving way to high-rank trains). The present study combines weather records, historical train delay records and train schedule data to determine the most important factors influencing train delays. We propose the new concepts of key train delay stations and the time interval threshold to determine whether a delay to one train results in a delay to the following train. Finally, we present a long-term train delay prediction model.

A number of studies have also investigated the field of train delay prediction. Oneto et al. [18] used machine learning algorithms and statistical tools to construct a train delay prediction system (TDPS) for large-scale railway networks. The system employed large-scale memory data-processing technology. The train delay prediction model was evaluated using the actual performance data of the Italian rail network. Hansen et al. [19] compared the actual blocking time with the predicted blocking time based on train delay data. The authors developed an online model to accurately predict the running time between two stations considering route conflict, vehicle type, weather conditions and other factors. Murali et al. [20] developed a complex network model to simulate the train delay process, which was also used to predict train delays. Jiang et al. [21] proposed a simulation model to study the relationship between train delays and passenger delays. The authors predicted the dynamic distribution of passengers in large rail networks. Using a Bayesian network (BN), Lessan et al. [22] proposed a train delay prediction model. The k-fold cross-validation method was used to test the train history data in three BN structures (heuristic, naive and hybrid). The authors found that the reconstructed hybrid heuristic BN structure was able to achieve higher prediction performance. Yaghini et al. [23] proposed a high-precision neural network model to predict the late arrival of Iranian railway passenger trains. The authors used decision trees and multiple logistic regression models to evaluate the quality of the results. Milinković et al. [24] proposed a fuzzy Petri net (FPN) model to predict train delays. Hallowell et al. [25] analyzed the railway line data and proposed a model employing dynamic prioritization to approximate the optimal planning process and analyze train delays. Zhuang et al. [26] used historical operation data and high accuracy Petri nets to develop a high-speed railway train operation map model. The authors also proposed a new high-speed train running conflict prediction method. Berger et al. [26] proposed a stochastic model to predict train delay propagation in large-scale traffic networks. Actual operation data from the German rail network was used to verify the prediction results. Oneto et al. [25] proposed a train delay prediction system that considered both historical trains running conditions and weather conditions. The authors pointed out that weather conditions may affect passenger flow, which in turn affects the stopping time of trains.

1.3 CHALLENGES OF METHODOLOGICAL DEVELOPMENT

1.3.1 Sharp nonlinearity

Traffic flows usually exhibit either a free flow state or congested state. The sharp nonlinearity of traffic variables comes from recurrent or non-recurrent congested states, due to bottlenecks, incidents, extreme weather condition or other events. Fig. 1 presents a set of Fundamental Diagrams that are constructed with observed hourly traffic volume, speed, and occupancy collected from a loop detector on the Ayalon Highway in Tel Aviv, Israel. A free flow state in the diagrams is color-coded as green, and a congested state as red. Correspondingly, Fig. 2 shows time series plots of hourly traffic volume, speed, and occupancy, using the same set of traffic data as Fig. 1. Red dots in both figures match one-on-one. Figs. 1 and 2 complement each other. The macroscopic fundamental diagram (MFD) is an important instrument for adaptive traffic control. The fundamental diagrams show traffic states and relationships between pairs of traffic variables, but not traffic dynamics over time, whereas, time series plots show traffic evolution over time, but not traffic states. The red dots on the time series plots represent those traffic events or measurement outliers that are usually difficult to capture by many traffic predictive models.

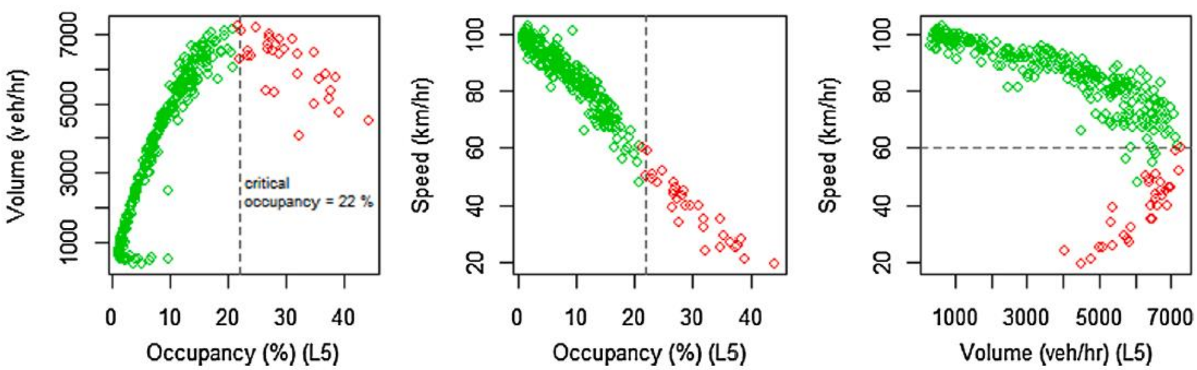


Fig. 1: Fundamental diagram created with traffic data

1.3.2 Network-scale spatial and temporal correlation

From a system perspective, the dynamics of traffic flows across urban road network exhibit network-wide temporal and spatial correlations, and co- movement patterns. As correlation matrix is symmetrical, Fig. 3(a) shows the upper triangular part of a correlogram of 10 traffic volume time series from the highway network. This correlogram combines the correlation coefficient and the pvalue of the correlation for all pairs of traffic time series. The p-values correspond to the significance levels of correlations. If the pvalue in the matrix is bigger than the specified significant level (e.g. $\alpha = 0.01$), then the corresponding correlation coefficient is regarded as insignificant and its graph is blank. Positive correlations are displayed in blue and negative correlations in red color. The intensity of color and the size of the circle are proportional to the correlation coefficients. The right-hand-side legend shows the scale of correlation coefficients and the corresponding colors. Fig. 3(b) shows the upper triangular part of a correlogram of 10 traffic speed time series from the highway network. There are different methods for correlation analysis between two random variables including, but not limited to, Pearson, Kendall, and Spearman (Bonett and Wright, 2000). Pearson's correlation is a parametric method that measures the linear dependence between two variables. Kendall and Spearman are non-parametric rank-based correlation analysis. In this study, Pearson's correlation coefficients between all pairs of traffic volume time series are computed to investigate the linear dependence between pairs of networkwide traffic flows. The numerical value of correlation coefficient, the color intensity, and the circle size in Fig. 3(a) and (b) together indicate that there exist strong spatial correlations and co-movement patterns among network-wide traffic flows. Fig. 4 shows a plot of cross-correlation between two traffic time series from location 5 and 6. The plot indicates a strong temporal correlation between two traffic time series. The cross-correlation function also indicates that repeating patterns or seasonality concurrently appear among two traffic flows.

1.4 SCOPE AND OBJECTIVES

- To develop a model that can predict train delays.
- To select an appropriate software for implementing the code.
- To monitor the areas where the train delays happen the most.

CHAPTER 2

METHADOLOGIES

2.1 Statistical methods

There are many classes of time series models that have been applied to traffic state prediction. A benchmark model starts from ARIMA (Autoregressive Integrated Moving Average). It had been extensively used for single-location traffic prediction and often chosen by researchers as a yardstick for the purpose of comparison with other models (Lee and Fambro, 1999; Smith et al., 2002). Williams and Hoel (2003) introduce a SARIMA (seasonal ARIMA) to model repeating patterns or seasonality of macroscopic traffic variables. Their results indicate that SARIMA outperforms ARIMA and provides significantly improved prediction accuracy. However, ARIMA and SARIMA models are mostly used for single-location traffic prediction, unable to reflect spatial correlation of topology structure of a transportation network. Kamarianakis and Prastacos (2005) introduce STARIMA (Space-Time ARIMA) to incorporate both spatial correlation between adjacent locations and temporal autocorrelation in one model, where a spatial-correlation matrix is introduced to a standard ARIMA as a coefficient matrix. The success of the model is conditional on the construction of a spatial weight matrix. This model was also investigated by Cheng et al. (2012), and Min et al. (2010). Cheng et al. (2012) argue that it is not always reliable to use a distance function to determine spatial order. An effective spatial order is dynamic, such that it becomes larger under a free flow traffic state and smaller under a congested traffic state. In order to consider multivariate traffic variables, Williams (2001) introduces ARIMAX model that uses transfer functions to include exogenous variables from upstream traffic. Min and Wynter (2011) introduce MSTARMA that is a multivariate spatial-temporal autoregressive moving average model for traffic volume and speed prediction. These are linear models that cannot deal with nonlinearity or structural changes in the traffic characteristics time series. To be able to consider nonlinearity, D'Angelo et al. (1999) and Liu et al. (2010) develop regime-switching SETAR (self-exciting threshold autoregressive) model for modelling and forecasting hourly traffic volume. Ma et al. (2015) develop nonlinear multivariate timespace threshold vector error correction model for short term traffic state prediction. The regime-switching models involve structural change detection and threshold identification mechanism. Similarly, Sun and Liu (2011) introduce a STAR (smooth transition autoregressive) model for traffic prediction. The model switches between two autoregressive (AR) parts via a smooth transition function. Kamarianakis et al. (2012) propose a temporal regime-switching model for traffic volume prediction. The time-based thresholds are predefined and fixed by partitioning 24 h into five temporal regimes. A linear autoregressive model is constructed for each time-based regime. Some statistical time series models are variants of simple models via combination. Zhang et al. (2014) propose a hybrid of ARIMA and GARCH model, where ARIMA is used to determine the mean and GARCH models the variance. The ARIMA-EM model is a combination of ARIMA and expectation maximization algorithm proposed by Cetin and Comert (2006). Other statistical methodologies are also proposed for traffic forecast. Castillo et al. (2008) develop a Bayesian Network for predicting traffic flow. Wang et al. (2014) propose a Bayesian combination method with three component predictors including ARIMA, Kalman Filter, and BPNN for traffic flow forecasting. Fusco et al. (2016) propose a hybrid modeling framework that joins a Bayesian network and a neural network for traffic speed predictions.

State-Space models are often classified as a standalone class. However, the essence of a state-space model is a statistical time series type of model that runs two time series in parallel, explicit and implicit. The term "state space" originated in the 1960s from the area of control engineering, and Kalman (1960) developed a Kalman algorithm to estimate state space models that are extensively used for traffic prediction (Antoniou et al., 2007; Okutani and Stephanedes, 1984; Whittaker et al., 1997). Stathopoulos and Karlaftis (2003) employ multivariate time-series state space models for traffic volume prediction for urban arterial streets and report that the state space model reduced into an ARIMA model in some cases. Guo et al. (2014) develop an adaptive Kalman filter approach to convert the SARIMA-GARCH structure into two state space representations for modeling and predicting traffic speed series. Dong et al. (2014) develop multivariate state space models based on macroscopic traffic flow models for network flow rate and time mean speed predictions taking into account congested and non-congested traffic state respectively. The Extended-Kalman-Filter (EKF) method was combined with second order macroscopic traffic flow models for traffic state prediction (Bellemans et al., 2006; Tampere and Immers, 2007; Wang and Papageorgiou, 2005). As both the Kalman recursion structure and the second order macroscopic traffic flow models are first order difference equations, it implies that the future traffic state is only correlated with the current measurement.

Other possible correlations between the future state and the past intervals and spatial correlations are not taken into account. Loosely speaking, Kalman recursion is a linear model. Its prediction performance is compromised for a nonlinear traffic state with significant sharp fluctuations. Qi and Ishak (2014) develop a Hidden Markov Model for short-term freeway traffic prediction during peak periods. A hidden Markov model is a specific type of state space model, if those states exhibit the Markov property. Both state space model and hidden Markov model are a kind of "hidden process model" based on probabilistic theory. Similarly, non-parametric K-Nearest Neighborhood (KNN) regression is also a type of statistical model, but often treated as a standalone class. Non-parametric models mainly refer to non-parametric KNN regression and its variants. The KNN regression is developed based on chaotic system theory rather than stochastic system theory that is the basis of time series models. Disbro and Frame (1989) and Wang et al. (2005) argue with some evidence that traffic flows exhibit chaotic behavior and properties. Shang et al. (2005) argue with phase space techniques that traffic time series have a strong chaotic signature, due to the positive largest Lyapunov Exponent and low correlation dimension. With the KNN method for traffic prediction, K observations are selected from a historical database based on their nearness to the current observation of traffic variable to form a nearest neighborhood. The efficiency of nonparametric KNN regression is directly dependent on the quality and size of the database. However, execution time searching for the nearest neighborhood will be compromised as the database expands. This is a significant issue for online applications of KNN to traffic flow prediction (Smith et al., 2002). Ryu et al. (2018) adopt KNN model with traffic state vectors for traffic flow prediction, where the traffic state vectors consider time delays and spatio-temporal correlations between the road sections in urban road network. Habtemichael and Cetin (2016) use an enhanced K Nearest Neighborhood algorithm for traffic volume forecast. Zheng and Su (2014) use a principle component technique to enhance the performance of the K-Nearest Neighborhood approach in forecasting traffic volume.

2.2 Machine learning

Machine learning techniques for traffic prediction primarily include Neural Network, Deep Learning, Support Vector Regression, and the recently proposed Spinning Network. Many types of Neural Networks have been developed for the purpose of traffic prediction, such as Back Propagation Neural Networks (BPNN) (Guo and Zhu, 2009; Liu et al., 2012; Zhu et al., 2010), Time Delay Neural Networks (TDNN) (Abdulhai et al., 2002), Elman Neural Network (ENN) (Gao et al., 2008) and Radial Basis Function (RBF) Network (Wang and Xiao, 2003; Yang et al., 2010). Qiao et al. (2001) propose a neural network-based system identification approach to establish an auto-adaptive model for simulating traffic flow dispersion and use it for online traffic flow forecasting. Zheng et al. (2006) develop a Bayesian combined Neural Network approach for freeway traffic flow prediction. Yin et al. (2002) develop a fuzzy-neural model for online rolling training and prediction. This fuzzy-neural model consists of two modules: a gate network is used to classify input data into clusters, whereas, an expert network is used to determine the input-output relationship. Vlahogianni et al. (2005) develop a genetic approach to optimize multilayer Neural Network structure for traffic flow prediction for an urban signalized arterial. Zhu et al. (2014) adopt RBF Neural
Networks for traffic volume forecast at a single location while considering traffic flows of the adjacent intersections. Dunne and Ghosh (2012) develop a Neural Network model with predetermined uncongested and congested regimes to perform traffic flow and speed prediction. Laña et al. (2019) propose evolving spiking neural networks for adaptive long-term traffic state estimation. Polson and Sokolov (2017) develop a deep learning model to predict traffic flows, where the first layer identifies spatio-temporal relations among predictors and other layers model nonlinear relations. It is reported that the deep learning architectures can capture the nonlinear spatial-temporal effects due to recurrent and non-recurrent traffic congestion patterns. Wu et al. (2018) propose a deep Neural Network-

based traffic flow prediction model that consists of an attention-based network, convolutional neural network, and recurrent neural network. Three networks are used to determine the importance of past traffic flow, mine the spatial features and temporal features of traffic flow respectively. Do et al. (2019), and Zhang et al. (2019) respectively propose a deep learning neural network for traffic flow prediction with attention mechanism to exploit spatial and temporal dependencies between road segments as well as time steps. Wu et al. (2004) propose SVR (support vector regression) for travel-time prediction using real highway traffic data. Wang and Shi (2013) propose a wavelet kernel function for SVR model to perform traffic speed forecasting, because the wavelet kernel function could capture both stationary and nonstationary, nonlinear characteristics of traffic speed data. In addition, Huang and Sadek (2009) develop a SPN (spinning network) model that is inspired by and mimics human memory mechanism for interstate highway traffic volume forecast. In addition to aforementioned two categories of methodologies, some fusion methods are proposed for traffic state prediction. Antoniou et al. (2013) propose a framework that includes a set of machine-learning approaches that perform three functions including clustering and classification, modeling the evolution of traffic states, and state-specific speed prediction. Guo et al. (2018) evaluate fusion-based frameworks using three different fusion strategies: averaged, weighed and k-Nearest Neighbor methods, applied to three different machine learning methods, Neural Networks, Support Vector Regression and Random Forests. It is worth mentioning that there is a class of models for estimating traffic state at any intermediate point from the boundary conditions of a segment of urban roadway or highway based on macroscopic traffic flow theory. Some representative works include, but are not limited to, the LWR partial differential equation (Lighthill and Whitham, 1955; Richards, 1956), the CTM (cell transmission model) (Daganzo, 1994; Munoz et al., 2003), second-order traffic flow model with Kalman filter (Nanthawichit et al., 2003), or with Lagrangian measurements (Herrera and Bayen, 2010), kinematic wave model (Daganzo, 2005; Newell, 1993), stochastic Newell's threedetector method (Deng et al., 2013; Laval et al., 2012). Overall, most of the aforementioned methodologies in the literature focus on local prediction. Moreover, the residual time series are usually not checked and verified by statistical tests to be a white noise. It is likely that traffic features still exist in the residuals. Hence, it is necessary to introduce a post-process mechanism for the residual time series of machine learning algorithms to make sure no traffic features remain in the residuals. To this end, we select the statistical ARIMA model as a post-processor to achieve this goal. The details of the working principles are described in Section 3.

CHAPTER 3

TERMINOLOGIES

3.1 Moving Averages

Moving average which can also be written as rolling average or running average is a calculation to analyze data points by creating a series of averages of different subsets of the full data set. It is also called a moving mean (MM) or rolling mean and is a type of finite impulse response filter. Variations include: simple, and cumulative, or weighted forms. Given a series of numbers and a fixed subset size, the first element of the moving average is obtained by taking the average of the initial fixed subset of the number series. Then the subset is modified by "shifting forward"; that is, excluding the first number of the series and including the next value in the subset.

A moving average is commonly used with time series data to smooth out short-term fluctuations and highlight longer-term trends or cycles. The threshold between short-term and long-term depends on the application, and the parameters of the moving average will be set accordingly. For example, it is often used in technical analysis of financial data, like stock prices, returns or trading volumes. It is also used in economics to examine gross domestic product, employment or other macroeconomic time series. Mathematically, a moving average is a type of convolution and so it can be viewed as an example of a low-pass filter used in signal processing. When used with non-time series data, a moving average filters higher frequency component without any specific connection to time, although typically some kind of ordering is implied. Viewed simplistically it can be regarded as smoothing the data.

Moving average (MA) is a process where the present value of series is defined as a linear combination of past errors. We assume the errors to be independently distributed with the normal distribution. The MA process of order q can be mathematically represented as,

$$y_t = c + \varepsilon_t + \emptyset_1 \varepsilon_{t_{t-1}} + \emptyset_2 \varepsilon_{t_{t-2}} + \cdots + \emptyset_p \varepsilon_{t_{t-p}} \quad (1)$$

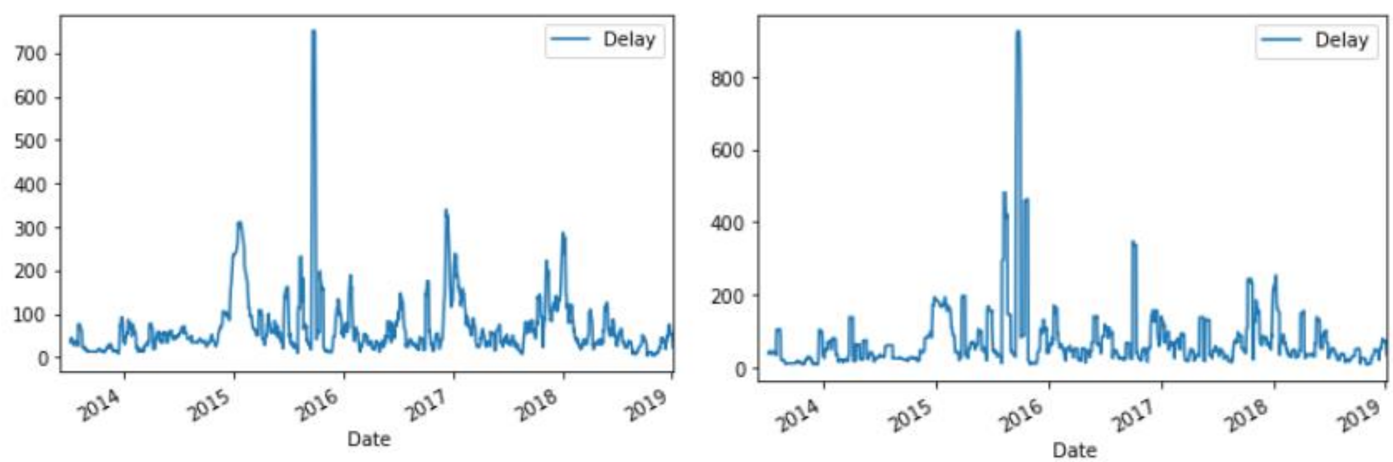
Here ϵt is a white noise. To get intuition of MA process let’s consider order 1 MA process which will look like,

$$y_t = c + \varepsilon_t + \emptyset_1 \varepsilon_{t_{t-1}} \quad (2)$$

let’s consider y'_t as the crude oil price and ϵ_t is the change in the oil price due to hurricane. Assume that c=10 (mean value of crude oil price when there is no hurricane) and $\theta_1=0.5$. Suppose, there is a hurricane today and it was not present yesterday, so y'_t will be 15 assuming the change in the oil price due to hurricane as $\epsilon_t=5$. Tomorrow there is no hurricane so y'_t will be 12.5 as $\epsilon_t=0$ and $\epsilon_{t-1}=5$. Suppose there is no hurricane day after tomorrow. In that case the oil price would be 10 which means it got stabilized back to mean after getting varied by hurricane. So the effect of hurricane only stays for one lagged value in our case. Hurricane in this case is an independent phenomenon.

Order q of the MA process is obtained from the ACF plot, this is the lag after which ACF crosses the upper confidence interval for the first time. As we know PACF captures correlations of residuals and the time series lags, we might get good correlations for nearest lags as well as for past lags. Why would that be? Since our series is linear combination of the residuals and none of time series own lag can directly explain its present (since its not an AR), which is the essence of PACF plot as it subtracts variations already explained by earlier lags, its kind of PACF losing its power here! On the other hand, being a MA process, it doesn’t have the seasonal or trend components so the ACF plot will capture the correlations due residual components only. You can also think of it as ACF which is a complete plot (capturing trend, seasonality, cyclic and residual correlations) acting as a partial plot since we don’t have trends, seasons, etc.

In the below code, I have defined a simple MA process and found its order using the ACF plot. We can expect the ACF plot to show good correlation with nearest lags and then sharp fall as its not an AR process to have good correlation with past lags. Also, we would expect the PACF plot to have gradual decrease as being an MA process, nearest lag values of time series cannot really predict its present value unlike AR process. So, we will get good correlations of residuals with further lags as well, hence the gradual decrease.



(a) (b)

Fig. 2: (a) Rolling mean (b) Rolling Standard Deviation

3.2 ACF and PACF Plots

ACF is an (complete) auto-correlation function which gives us values of auto-correlation of any series with its lagged values. We plot these values along with the confidence band and tada! We have an ACF plot. In simple terms, it describes how well the present value of the series is related with its past values. A time series can have components like trend, seasonality, cyclic and residual. ACF considers all these components while finding correlations hence it's a 'complete auto-correlation plot'.

PACF is a partial auto-correlation function. Basically instead of finding correlations of present with lags like ACF, it finds correlation of the residuals (which remains after removing the effects which are already explained by the earlier lag(s)) with the next lag value hence 'partial' and not 'complete' as we remove already found variations before we find the next correlation. So, if there is any hidden information in the residual which can be modeled by the next lag, we might get a good correlation and we will keep that next lag as a feature while modeling. Remember while modeling we don't want to keep too many features which are correlated as that can create multicollinearity issues. Hence, we need to retain only the relevant features.

Auto regressive (AR) process, a time series is said to be AR when present value of the time series can be obtained using previous values of the same time series i.e the present value is weighted average of its past values. Stock prices and global temperature rise can be thought of as an AR processes. Mathematically it can be represented as:

$$y_t = c + \emptyset_1 y_{t-1} + \emptyset_2 y_{t-2} + \cdots + \emptyset_p y_{t-p} + \varepsilon_t \quad (3)$$

Where ϵt is a white noise and $y't_{-1}$ and $y't_{-2}$ are the lags. Order p is the lag value after which PACF plot crosses the upper confidence interval for the first time. These p lags will act as our features while forecasting the AR time series. We cannot use the ACF plot here because it will show good correlations even for the lags which are far in the past. If we consider those many features, we will have multicollinearity issues. This is not a problem with PACF plot as it removes components already explained by earlier lags, so we only get the lags which have the correlation with the residual i.e. the component not explained by earlier lags.

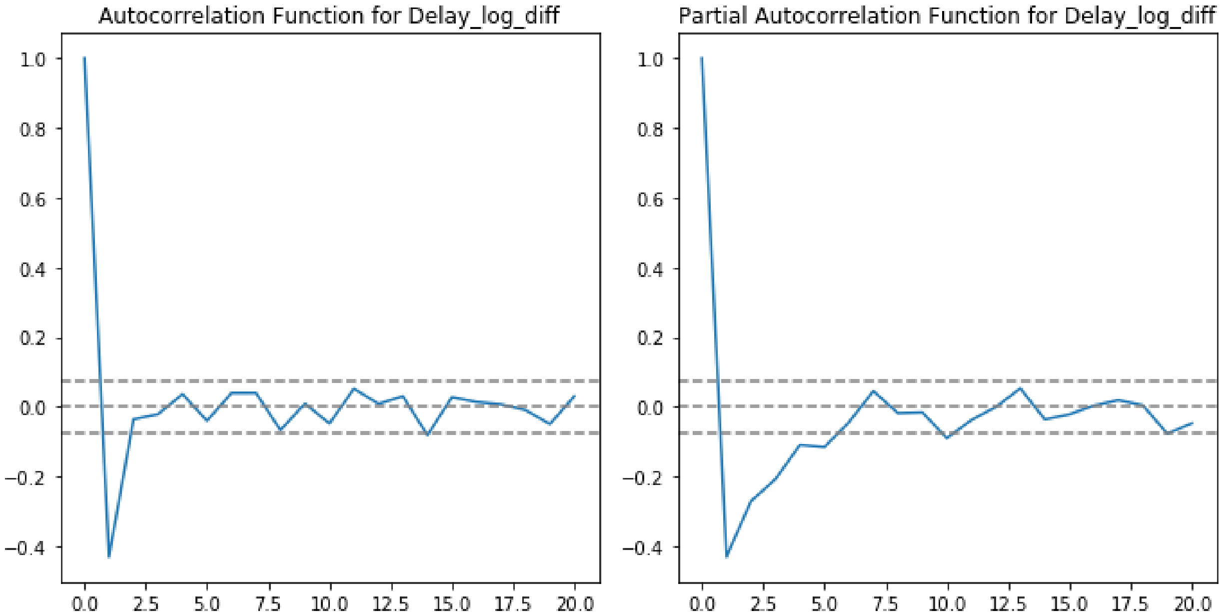


Fig. 3: ACF and PACF Plots

CHAPTER 4

SELECTION AND DESIGN OF MODEL

4.1 ARMIA

Auto Regressive Integrated Moving Average (ARMIA) is a univariate linear model which can be classified into two different types namely seasonal and non-seasonal ARMIA models which can be used for time series analysis. It has been widely applied to traffic flow forecasting for a single location in urban areas for a few decades, but it has limited accuracy due to the fact that it lacks the capacity to capture nonlinear properties of traffic time series. However, the advantages of ARIMA models are its simple mathematical form and capacity to explicitly identify the temporal correlation in the time series. The mathematical form of ARIMA (p, d, q) model can be written as Eq. (3) (Brockwell, 2016).

$$\emptyset(B)(1-B)^{d}X_{t} = \theta(B)e_{t} \quad (4)$$

where X_t is time series variable, $\{e_t\} \sim WN(0,\sigma^2)$, $\emptyset(B) = 1 - \emptyset_1 B - \cdots - \emptyset_p B^P$ and $\theta(B) = 1 + \emptyset_1 B + \cdots + \emptyset_q B^q$ are pth and qth degree polynomials, $\emptyset_1, \ldots, \emptyset_p$ and $\theta_1, \ldots, \theta_q$ are coefficients, d denotes the order of difference, B is the backward shift operator, p and q represent the autoregressive and moving average order, respectively.

In statistics and econometrics, and in particular in time series analysis, an autoregressive integrated moving average (ARIMA) model is a generalization of an autoregressive moving average (ARMA) model. Both of these models are fitted to time series data either to better understand the data or to predict future points in the series (forecasting). ARIMA models are applied in some cases where data show evidence of non-stationarity, where an initial differencing step (corresponding to the "integrated" part of the model) can be applied one or more times to eliminate the non-stationarity.

The AR part of ARIMA indicates that the evolving variable of interest is regressed on its own lagged (i.e., prior) values. The MA part indicates that the regression error is actually a linear combination of error terms whose values occurred contemporaneously and at various times in the past. The I (for "integrated") indicates that the data values have been replaced with the difference between their values and the previous values (and this differencing process may have been performed more than once). The purpose of each of these features is to make the model fit the data as well as possible.

Non-seasonal ARIMA models are generally denoted ARIMA(p,d,q) where parameters p, d, and q are non-negative integers, p is the order (number of time lags) of the autoregressive model, d is the degree of differencing (the number of times the data have had past values subtracted), and q is the order of the moving-average model. Seasonal ARIMA models are usually denoted ARIMA(p,d,q)(P,D,Q)m, where m refers to the number of periods in each season, and the uppercase P,D,Q refer to the autoregressive, differencing, and moving average terms for the seasonal part of the ARIMA model.

4.2 Prophet

Prophet is designed to have intuitive parameters that can be adjusted without knowing the details of the underlying model. This is necessary for the analyst to effectively tune the model. The implementation of this model can be easily be done in python environment. Decomposable time series model (Harvey & Peters 1990) with three main model components: trend, seasonality, and holidays. They are combined in the following equation:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \quad (5)$$

Here g(t) is the trend function which models non-periodic changes in the value of the time series, s(t) represents periodic changes (e.g., weekly and yearly seasonality), and h(t) represents the effects of holidays which occur on potentially irregular schedules over one or more days. The error term t represents any idiosyncratic changes which are not accommodated by the model; later we will make the parametric assumption that ϵ_t represents any idiosyncratic changes which are not accommodated by the model; later we will make the parametric assumption that ϵ_t is normally distributed.

This specification is similar to a generalized additive model (GAM) (Hastie & Tibshirani 1987), a class of regression models with potentially non-linear smoothers applied to the regressors. Here we use only time as a regressor but possibly several linear and non-linear functions of time as components. Modeling seasonality as an additive component is the same approach taken by exponential smoothing (Gardner 1985). Multiplicative seasonality, where the seasonal effect is a factor that multiplies g(t), can be accomplished through a log transform.

The GAM formulation has the advantage that it decomposes easily and accommodates new components as necessary, for instance when a new source of seasonality is identified. GAMs also fit very quickly, either using backfitting or L-BFGS (Byrd et al. 1995) (we prefer the latter) so that the user can interactively change the model parameters. We are, in effect, framing the forecasting problem as a curve-fitting exercise, which is inherently different from time series models that explicitly account for the temporal dependence structure in the data. While we give up some important inferential advantages of using a generative model such as an ARIMA, this formulation provides a number of practical advantages:

– Flexibility: We can easily accommodate seasonality with multiple periods and let the analyst make different assumptions about trends.

– Unlike with ARIMA models, the measurements do not need to be regularly spaced, and we do not need to interpolate missing values e.g. from removing outliers.

– Fitting is very fast, allowing the analyst to interactively explore many model specifications, for example in a Shiny application (Chang et al. 2015).

– The forecasting model has easily interpretable parameters that can be changed by the analyst to impose assumptions on the forecast. Moreover, analysts typically do have experience with regression and are easily able to extend the model to include new components.

Automatic forecasting has a long history, with many methods tailored to specific types of time series (Tashman & Leach 1991, De Gooijer & Hyndman 2006). Our approach is driven by both the nature of the time series we forecast at Facebook (piecewise trends, multiple seasonality, floating holidays) as well as the challenges involved in forecasting at scale.

4.2.1 The Trend Model

For this experiment we have implemented two trend model which cover a large range of applications and are classified into saturating growth model and a piecewise linear model.

4.2.1.1 Nonlinear, Saturating Growth

For growth forecasting, the core component of the data generating process is a model for how the population has grown and how it is expected to continue growing. Modeling growth at Facebook is often similar to population growth in natural ecosystems (e.g., Hutchinson 1978), where there is nonlinear growth that saturates at a carrying capacity. For example, the carrying capacity for the number of Facebook users in a particular area might be the number of people that have access to the Internet. This sort of growth is typically modeled using the logistic growth model, which in its most basic form is

$$g(t) = \frac{C}{1 + \exp(-k(t-m))} \quad (6)$$

with C the carrying capacity, k the growth rate, and m an offset parameter. There are two important aspects of growth at Facebook that are not captured in (2). First, the carrying capacity is not constant – as the number of people in the world who have access to the Internet increases, so does the growth ceiling We thus replace the fixed capacity C with a time-varying capacity $C(t)$. Second, the growth rate is not constant. New products can profoundly alter the rate of growth in a region, so the model must be able to incorporate a varying rate in order to fit historical data. We incorporate trend changes in the growth model by explicitly defining changepoints where the growth rate is allowed to change. Suppose there are S changepoints at times S_j. $j = 1 \ldots, S$. We define a vector of rate adjustments $\delta \in \mathrm{R}^s$, where δ_j is the change in rate that occurs at time S_j. The rate at any time t is then the base rate k, plus all of the adjustments up to that point: $k + \sum_{j:t>s_j} s_j$. This is represented more cleanly by defining a vector $\boldsymbol{a}(t) \epsilon \{0,1\}^s$ such that

$$a_j(t) = \begin{cases} 1, & if\ t \geq s_j \\ 0, & otherwise \end{cases} \quad (7)$$

The rate at time t is then $k + \boldsymbol{a}(t)^T \delta$. When the rate k is adjusted, the offset parameter m must also be adjusted to connect the endpoints of the segments. The correct adjustment at changepoint j is easily computed as

$$\gamma_j = \left(s_j - m - \sum_{l<j} \gamma l\right)\left(1 - \frac{k+\sum_{l<j} \gamma l}{k+\sum_{l\leq j} \gamma l}\right) \quad (7)$$

The piecewise logistic growth model is then

$$g(t) = \frac{C(t)}{1 + \exp\left(-k + \boldsymbol{a(t)^T}\delta\right)(t - (m + a(t)^T\gamma))} \quad (8)$$

An important set of parameters in our model is $C(t)$, or the expected capacities of the system at any point in time. Analysts often have insight into market sizes and can set these accordingly. There may also be external data sources that can provide carrying capacities, such as population forecasts from the World Bank.

The logistic growth model presented here is a special case of generalized logistic growth curves, which is only a single type of sigmoid curve. Extensions of this trend model to other families of curves is straightforward.

4.2.1.2 Linear Trend with Changepoints

For forecasting problems that do not exhibit saturating growth, a piece-wise constant rate of growth provides a parsimonious and often useful model. Here the trend model is

$$g(t) = (k + \boldsymbol{a}(t)^T \boldsymbol{\delta})t + (m + \boldsymbol{a}(t)^T \boldsymbol{\gamma}) \quad (9)$$

whereas before k is the growth rate, $\boldsymbol{\delta}$ has the rate adjustments, m is the offset parameter.

4.2.1.3 Automatic Changepoint Selection

The changepoints s_j could be specified by the analyst using known dates of product launches and other growth-altering events, or may be automatically selected given a set of candidates. Automatic selection can be done quite naturally with the formulation in (3) and (4) by putting a sparse prior on δ.

We often specify a large number of changepoints (e.g., one per month for a several year history) and use the prior $\delta_j \sim Laplace(0, \tau)$. The parameter τ directly controls the flexibility of the model in altering its rate. Importantly, a sparse prior on the adjustments δ has no impact on the primary growth rate k, so as τ goes to 0 the fit reduces to standard (not-piecewise) logistic or linear growth.

CHAPTER 5

MODEL FITTING

When the seasonality and holiday features for each observation are combined into a matrix X and the changepoint indicators $a(t)$ in a matrix A, the entire model in (1) can be expressed in a few lines of Stan code for model fitting we use Stan's L-BFGS to find a maximum a posteriori estimate, but also can do

```
model {
      // Priors
      k ~ normal (0, 5);
      m ~ normal (0, 5);
      epsilon ~ normal (0, 0.5);
      delta ~ double_exponential (0, tau);
      beta ~ normal (0, sigma);
      // Logistic likelihood
      y ~ normal (C./ (1 + exp (- (k + A * delta). * (t - (m + A *
gamma)))) + X * beta, epsilon);
      // Linear likelihood
      y ~ normal ((k + A * delta). * t + (m + A * gamma) + X * beta,
sigma);
}
```

full posterior inference to include model parameter uncertainty in the forecast uncertainty. The Prophet forecast is able to predict both the weekly and yearly seasonalities, and unlike the baselines, does not overreact to the holiday dip in the first year. In the first forecast, the Prophet model has slightly overfit the yearly seasonality given only one year of data.

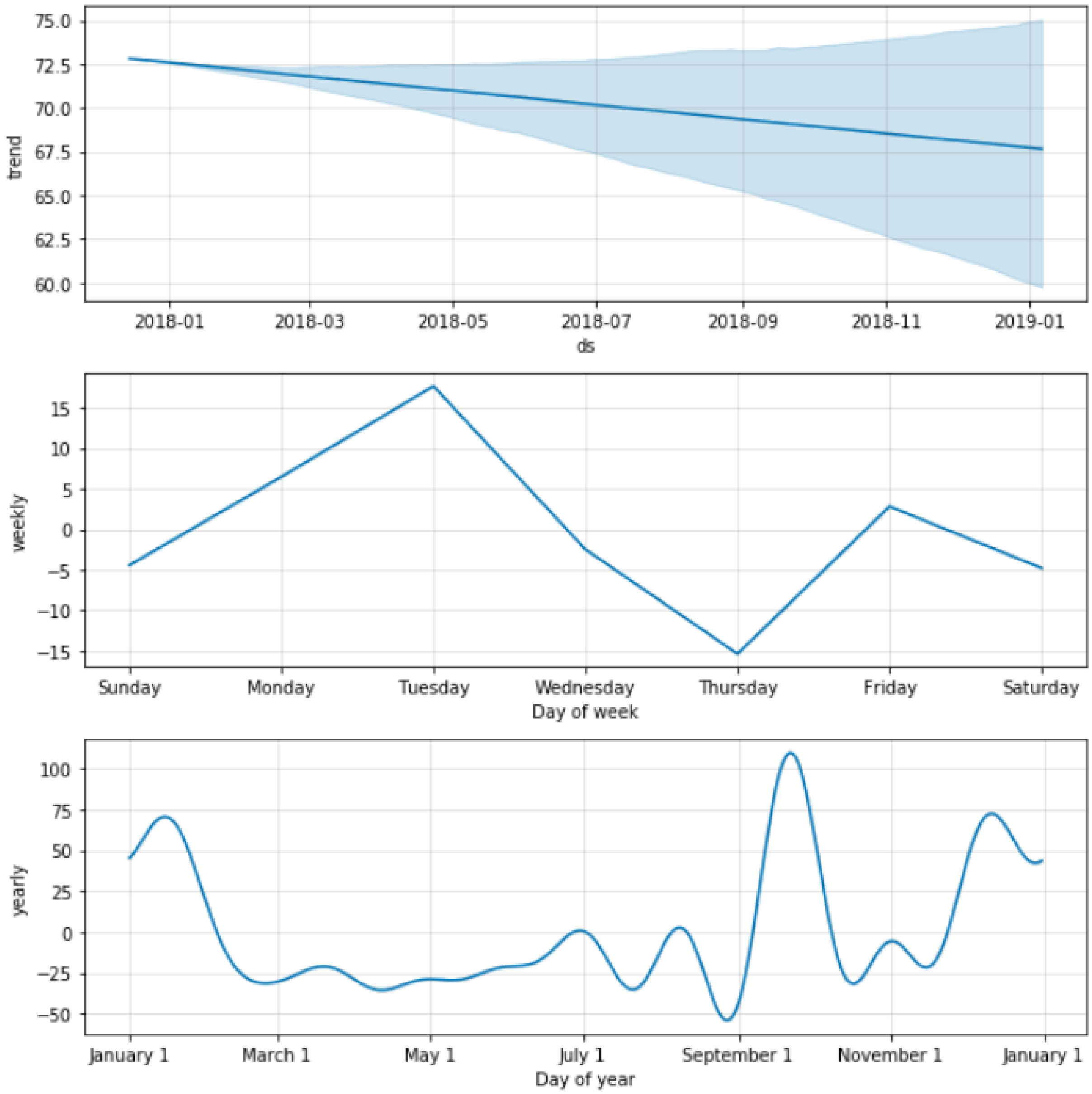


Fig. 4: Components of the Prophet forecast

The parameters tau and sigma in Listing 1 are controls for the amount of regularization on the model changepoints and seasonality respectively. Regularization is important for both of these to avoid overfitting, however there likely is not enough historical data to select the best regularization parameters via cross-validation. We set default values that are appropriate for most forecasting problems, and when these parameters need to be optimized it happens with the analyst in the loop.

Analysts making forecasts often have extensive domain knowledge about the quantity they are forecasting, but limited statistical knowledge. In the Prophet model specification, there are several places where analysts can alter the model to apply their expertise and external knowledge without requiring any understanding of the underlying statistics.

Capacities: Analysts may have external data for the total market size and can apply that knowledge directly by specifying capacities.

Changepoints: Known dates of changepoints, such as dates of product changes, can be directly specified.

Holidays and seasonality: Analysts that we work with have experience with which holidays impact growth in which regions, and they can directly input the relevant holiday dates and the applicable time scales of seasonality.

Smoothing parameters: By adjusting τ an analyst can select from within a range of more global or locally smooth models. The seasonality and holiday smoothing parameters (σ, ν) allow the analyst to tell the model how much of the historical seasonal variation is expected in the future.

With good visualization tools, analysts can use these parameters to improve the model fit. When the model fit is plotted over historical data, it is quickly apparent if there were changepoints that were missed by the automatic changepoint selection. The τ parameter is a single knob that can be turned to increase or decrease the trend flexibility, and σ is a knob to increase or decrease the strength of the seasonality component. Visualization provides many other opportunities for fruitful human intervention: linear trend or logistic growth, identifying time scales of seasonality, and identifying outlying time periods that should be removed from fitting are a few. All of these interventions can be made without statistical expertise, and are important ways for analysts to apply their insights or domain knowledge.

The forecasting literature often makes the distinction between statistical forecasts, which are based on models fit to historical data, and judgmental forecasts (also called managerial forecasts), which human experts produce using whatever process they have learned tends to work well for a specific time series. Each of these approaches has their advantages. Statistical forecasts require less domain knowledge and effort from human forecasters, and they can scale to many forecasts very easily. Judgmental forecasts can include more information and be more responsive to changing conditions, but can require intensive work by analysts (Sanders 2005).

Our analyst-in-the-loop modeling approach is an alternative approach that attempts to blend the advantages of statistical and judgmental forecasts by focusing analyst effort on improving the model when necessary rather that directly producing forecasts through some unstated procedure. We find that our approach closely resembles the “transform-visualize model” loop proposed by Wickham & Grolemund (2016), where the human domain knowledge is codified in an improved model after some iteration.

Typical scaling of forecasting would rely on fully automated procedures, but judgmental forecasts have been shown to be highly accurate in many applications (Lawrence et al. 2006). Our proposed approach lets analysts apply judgment to forecasts through a small set of intuitive model parameters and options, while retaining the ability to fall back on fully automated statistical forecasting when necessary. As of this writing we have only anecdotal empirical evidence for possible improvements to accuracy, but we look forward to future research which can evaluate the improvements analysts can have in a model assisted setting.

The ability to have an analyst-in-the-loop at scale relies critically on automatic evaluation of forecast quality and good visualization tools. We now describe how forecast evaluation can be automated to identify the most relevant forecasts for analyst input.

CHAPTER 6

FLOWCHART FOR THE PROPOSED METHADOLOGY OF TIMES SERIES ANALYSIS

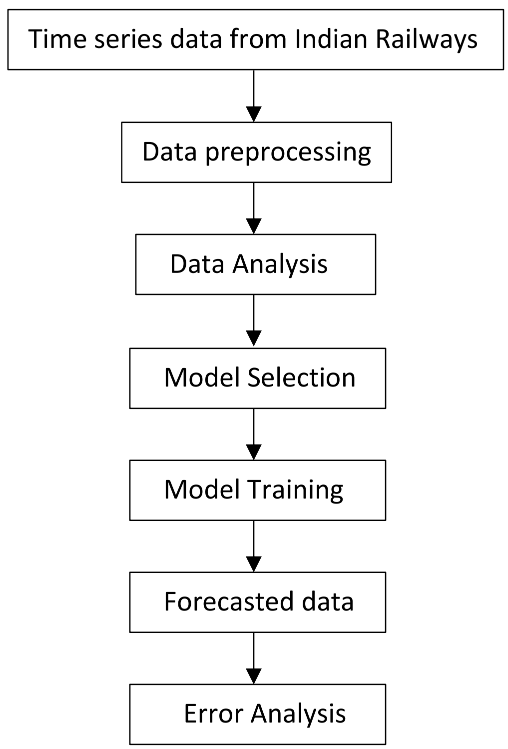


Fig 5: Flowchart of the performed study

CHAPTER 7

RESULTS AND DISCUSSION

The forecast is performed for macroscopic traffic variables. For the current study the dataset we spit the dataset into ratio of 8:2 which can be seen clearly in Fig. 6. Two different ways are considered to visualize the result of prediction. Fig.8 shows the scatter plots of forecasted volume vs. predicted volume. In each plot, if a forecast value exactly matches the observed value, the dot should be exactly on the 45-degree diagonal line. If a forecast value is greater than the observed value, the dot is above the diagonal line, and vice versa. The further the dots deviate from the diagonal line, the less accurate the forecast value is, and the bigger the mean squared error of the prediction is. If the majority of the dots lie on one side of the line, it means that the model produces forecast with bias. From the plots, it can be seen that the predictions of the prophet model are evenly distributed on both sides of the diagonal line and concentrate on the line with very narrow deviation. The predictions by prophet model not only have more points on one side of the diagonal line, but also have bigger deviation. Fig. 7 shows time series plot of predicted traffic delay volume. The lines of predictions show how close each prediction to the observed value and how the predictions evolve over time. It can be seen that the prediction by prophet model in the red dot line matches the line of observations.

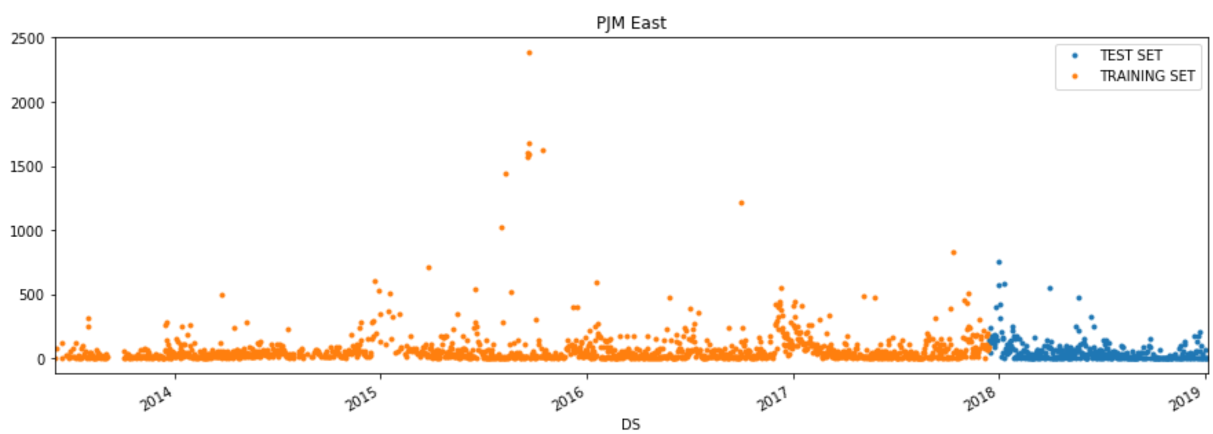


Fig 6: Train test split of the dataset

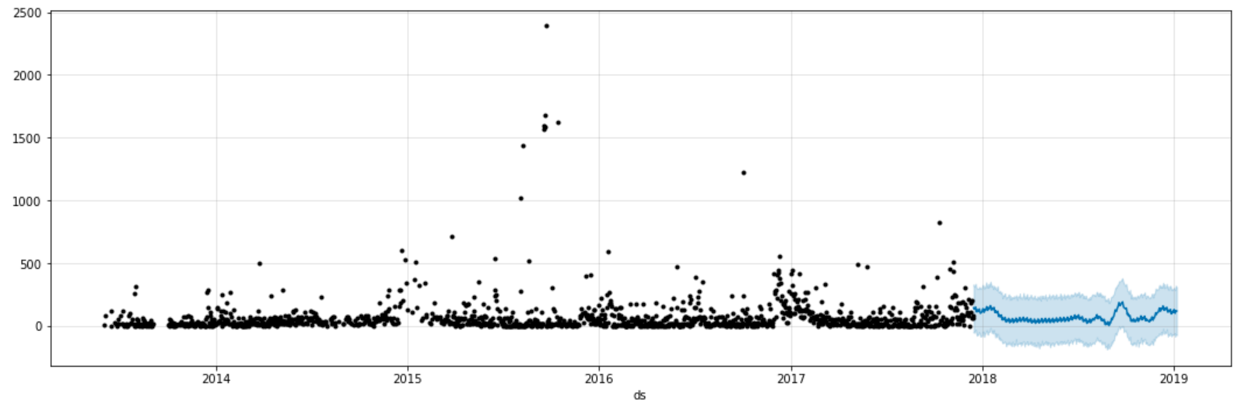


Fig 7: Forecasted data points

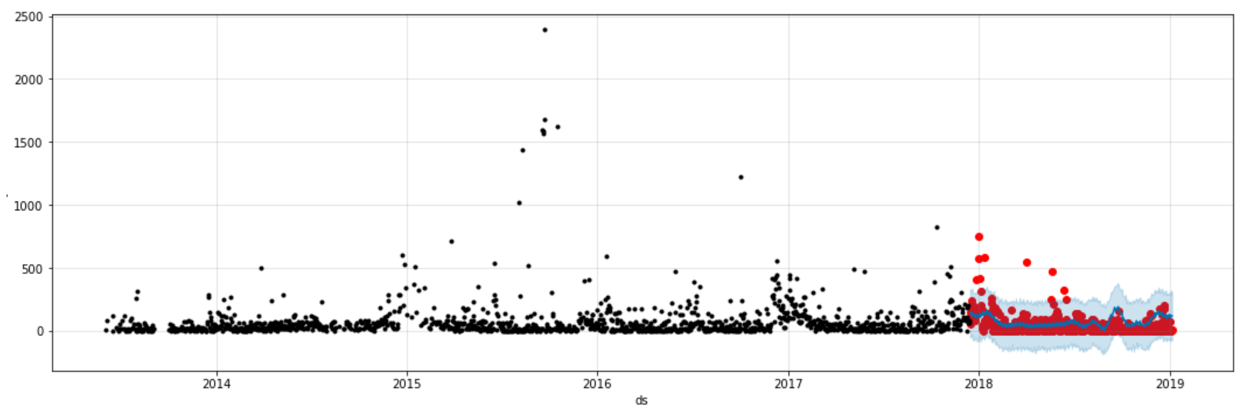


Fig 8: Forecasted data samples overlapped over original data samples

The total amount of data that is collected is of seven years. This data is focused on Jhelum express (Train number - 11077) which has running duration of 40 hrs. 40 min. It commences its journey station from Pune and ends at Jammu which is approximately 2000 km away. Fig. 9 Shows the route taken by the train. The major stations that come in route are namely Bhopal, Jhansi, New Delhi and Ludhiana. We monitored the route closely and created a database which is further used in this study. Table 1 shows a snippet which is prepared for this study.

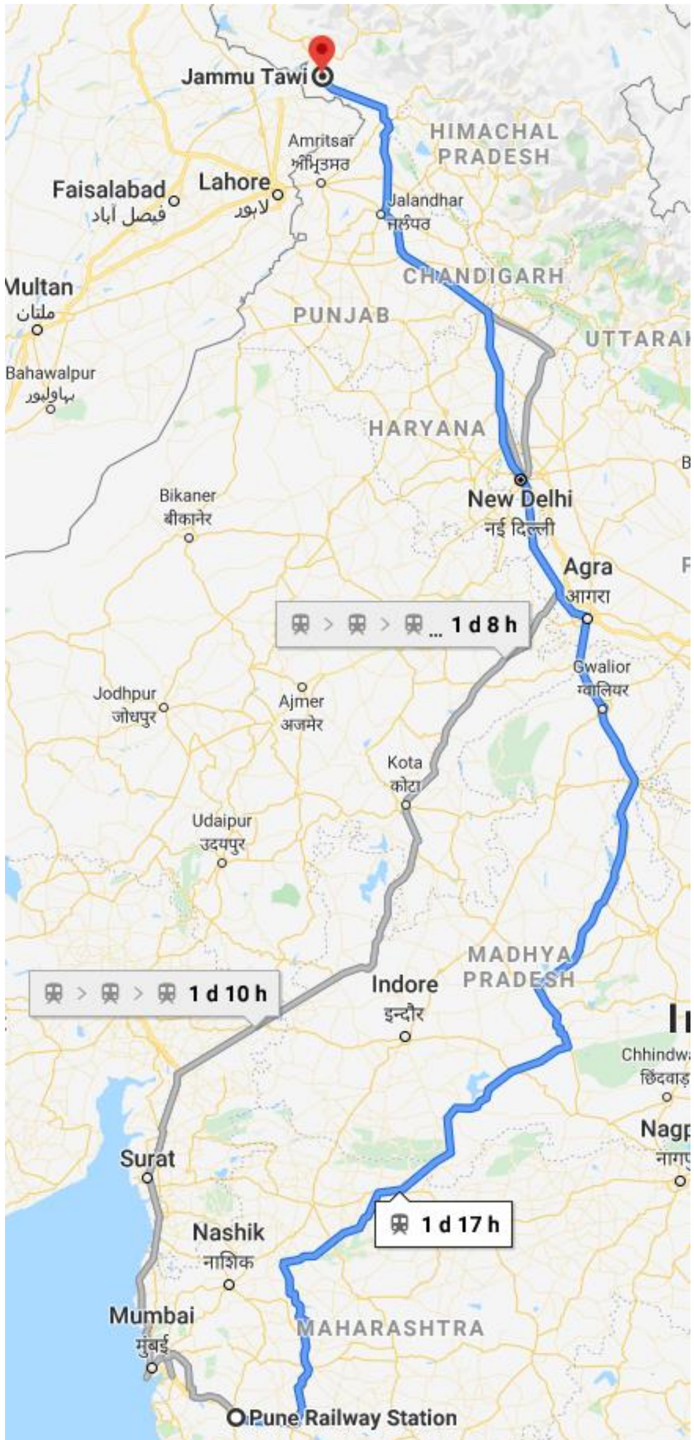


Fig 9: Route taken by the Jhelum express

Station	Sch Arrival	Act Arrival	Delay Time	Date	Train Number
Pune	Source	Source	15M	1st jaunary 2018	11077
Uruli	5:46 AM	6:05 AM	19M	1st jaunary 2018	11077
Daund	6:35 PM	7:00 PM	25M	1st jaunary 2018	11077
Ahmadnagar	8:22 PM	9:01 PM	39M	1st jaunary 2018	11077
Belapur	9:24 PM	9:44 PM	20M	1st jaunary 2018	11077
Kopargaon	10:14 PM	10:24 PM	10M Late	1st jaunary 2018	11077
Manmad	11:20 PM	11:20 PM	No Delay	1st jaunary 2018	11077

Table 1: Snippet of the dataset considered for this study.

CONCLUSION

In the study, we proposed a novel approach that sequentially concatenates a machine learning algorithm with a statistical model, i.e. PROPHET. This approach is able to consider both network-scale spatial-temporal correlations among traffic flows and location specific traffic characteristics. Its postprocessing by the ARIMA analysis can extract traffic features from the residuals of Neural Network, hence, significantly improve the accuracy of prediction. In addition, this approach is also able to capture the sharp nonlinearity of traffic flows. In the context of using seven years long railway traffic time series data set, the case study provides numerical evidence that the predictive capacity of the proposed prophet model. It is an effective and efficient tool for delay forecasting prediction. It is noteworthy that the MSE is significantly reduced in the comparison of ARIMA. It clearly reflects the important role of network-wide spatial correlations in traffic prediction. It is beneficial if the network traffic flows are treated as a whole in the prediction. The postprocess significantly improves the accuracy of delay prediction. In the future research, the proposed approach may be applied to traffic prediction for a large dataset over a wide range of trains if data is available. It may also be compared with those preprocess machine learning algorithms.

ACKNOWLEDGEMENT

The authors are sincerely thankful to the Government of India, Ministry of Railways, Research Design and Standards Organisation (RDSO), Lucknow, for facilitating this research project.

REFERENCES

[1]. Magadagela K, Nel H, Marnewick A. Identification of delay factors that affect high dwell times of freight trains. In: IEEE Technology & Engineering Management Conference (TEMSCON), Santa Clara, CA, USA, 2017, 179–84.

[2]. Olsson NOE, Haugland H. Influencing factors on train punctuality—results from some Norwegian studies. Transp Policy 2004;11:387–397.

[3]. Wei D, Liu H, Qin Y. Modeling cascade dynamics of railway networks under inclement weather. Transp Res Part E Log Transp Rev 2015;80:95–122.

[4]. Cerreto F, Nielsen OA, Harrod S, et al. Causal analysis of railway running delays. In: 11th World Congress on Railway Research (WCRR) , Milan, Italy, 2016.

[5]. Lalinska J, Gašparik J, Šipuš D, et al. Factors affecting the delay of passenger trains. Logi Sci J Transp Log 2017;8:74–81.

[6]. Jiang Z, Xu R, Xie C. Train delay propagation simulation in rail transit system. In: International Conference on Transportation Engineering, Chengdu, China, 2007, 789–94.

[7]. Schlake BW, Barkan CPL, Edwards JR. Train delay and economic impact of in-service failures of railroad rolling stock. Transp Res Rec 2011;2261:124–133.

[8]. Ludvigsen J, Klæboe R. Extreme weather impacts on freight railways in Europe. Nat Hazards 2014;70:767–787.

[9]. Wang J. Research on intelligent analysis system of railway signal equipment fault based on computer monitoring. Sci-tech Innov Product 2013;12:101–103.

[10]. Dingler M, Koenig A, Sogin S, et al. Determining the causes of train delay . In: AREMA Annual Conference Proceedings. Orlando, FL, USA, 2010. https://www.arema.org//files/library/2010_Conference_Proceedings/Determining_The_Cause_of_Train_Delay.pdf.

[11]. Weng J, Zheng Y, Yan X, et al. Development of a subway operation incident delay model using accelerated failure time approaches. Accid Anal Prev 2014;73:12–19.

[12]. Berger A, Gebhardt A, Müller-Hannemann M, et al. . Stochastic delay prediction in large train networks. In: 11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems, Saarbrücken, Germany, 2011, 100–11.

[13]. Huisman T, Boucherie RJ. Running times on railway sections with heterogeneous train traffic. Transp Res Part B Meth 2001;35:271–292.

[14]. Huisman T, Boucherie RJ, van Dijk NM. A solvable queueing network model for railway networks and its validation and applications for the Netherlands. Eur J Oper Res 2002;142:30–51.

[15]. Goverde RMP, Meng L. Advanced monitoring and management information of railway operations. J Rail Transp Plan Manag 2011;1:69–79.

[16]. Goverde RMP. A delay propagation algorithm for large-scale railway traffic networks. Transp Res Part C Emerg Technol 2010;18:269–287.

[17]. Oneto L, Fumeo E, Clerico G, et al. Train delay prediction systems: a big data analytics perspective . Big Data Res 2018;11:54–64.

[18]. Hansen IA, Goverde RMP, van der Meer DJ. Online train delay recognition and running time prediction. In: 13th International IEEE Conference on Intelligent Transportation Systems, Funchal, Portugal, 2010, 1783–8.

[19]. Murali P, Dessouky M, Ordóñez F, et al. A delay estimation technique for single and double-track railroads. Transp Res Part E Log Transp Rev 2010;46:483–495.

[20]. Jiang Z, Li F, Xu R, et al. A simulation model for estimating train and passenger delays in large-scale rail transit networks. J Cent South Univ 2012;19:3603–13.

[21]. Lessan J, Fu L, Wen C. (12 March 2018) A hybrid bayesian network model for predicting delays in train operations. Comput Ind Eng , 10.1016/j.cie.2018.03.017.

[22]. Yaghini M, Khoshraftar MM, Seyedabadi M. Railway passenger train delay prediction via neural network model. J Adv Transp 2013;47:355–368.

[23]. Hallowell S. F, Harker PT. Predicting on-time performance in scheduled railroad operations: methodology and application to train scheduling. Transp Res Part A Policy Pract 1998;32:279–295.

[24]. Zhuang H, Feng L, Wen C, et al. Corrigendum to “high-speed railway train timetable conflict prediction based on fuzzy temporal knowledge reasoning” [Engineering (2016) 366−373]. Engineering 2017;3:150.

[25]. Berger A, Gebhardt A, Müller-Hannemann M, et al. . Stochastic delay prediction in large train networks. In: 11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems, Saarbrücken, Germany, 2011, 100–11.

[26]. Oneto L, Fumeo E, Clerico G, et al. Advanced analytics for train delay prediction systems by including exogenous weather data. In: 3rd IEEE International Conference on Data Science and Advanced Analytics (DSAA), Montreal, Canada, 2016, 458–67.