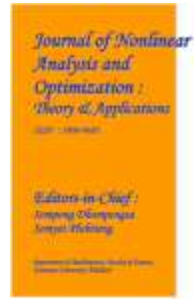
Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 2 : 2024
ISSN : **1906-9685**



Enhancing High-Speed Train Dispatching: Predicting Time Delays with Spatio-Temporal Graph Convolutional Networks

[1] MR.B.SURESH, [2]SHAIK PEDA KHASIM

#1 Assistant Professor, Department of Master of Computer Applications, QIS College of Engineering & Technology, Ongole, Andhra Pradesh, India

#2 PG Scholar, Department of Master of Computer Applications, QIS College of Engineering & Technology, Ongole, Andhra Pradesh, India

ABSTRACT

Train delay prediction can improve the quality of train dispatching, which helps the dispatcher to estimate the running state of the train more accurately and make reasonable dispatching decision. The delay of one train is affected by many factors, such as passenger flow, fault, extreme weather, dispatching strategy. The departure time of one train is generally determined by dispatchers, which is limited by their strategy and knowledge. The existing train delay prediction methods cannot comprehensively consider the temporal and spatial dependence between the multiple trains and routes. In this paper, we don't try to predict the specific delay time of one train, but predict the collective cumulative effect of train delay over a certain period, which is represented by the total number of arrival delays in one station. We propose a deep learning framework, train spatio-temporal

graph convolutional network (TSTGCN), to predict the collective cumulative effect of train
delay in one station for train dispatching and emergency plans. The proposed model is mainly composed of the recent, daily and weekly components. Each component contains two parts: spatio-temporal attention mechanism and spatio-temporal convolution,

which can effectively capture spatio-temporal characteristics. The weighted fusion of the three components produces the final prediction result. The experiments on the train operation data from China Railway Passenger Ticket System demonstrate that TSTGCN clearly outperforms the existing advanced baselines in train delay prediction.

Index: train, delay, dispatching, cnn, spatio temporal graph, tstgc

INTRODUCTION

BY JANUARY 2021, the total mileage of China's high-speed railway is 39,000 kilometers. Highspeed trains are favored by people for low price, high travel efficiency, safety and service quality. In recent years, with the continuous expansion of high-speed railway network and the continuous improvement of service quality, high-speed train has become one of the most important travel modes in China. Train delay is always one of the key research issues in train dispatching management and transportation organization. Unplanned interference may cause delay.

The train delay has propagation characteristics. Delayed trains not only affect their own operation, but also spread in one area, affecting the operation of other trains. Therefore, train delay prediction is one of the core tasks of train dispatching. Train delay prediction is of great significance to improving the quality of dispatching. Train delay prediction is mainly about to predict the influence degree of train operation interference and delay propagation, which is helpful to realizing real-time risk analysis and early warning of dispatching, as well as real-time adjustment of multi-mode transportation schedule in emergency [1]. It can assist dispatchers to analyze the operation status of trains, estimate delay risk, and serve as the basis for making reasonable traffic dispatching decisions [2]. Therefore, it is of significance to study the prediction model of train delay, which can provide support for the high-speed railway traffic command automation system.

A lot of work has been done to analyze and predict the train delay. Milinkovi*et al.* [3] proposed a fuzzy Petri net model to simulate the traffic process and train operation in the railway system; Tikhonov *et al.* [4] analyzed the relationship between the arrival delay of passenger trains and various features of the railway system, then applied SVM to the train delay analysis; Corman and Kecman [5] and Lessan*et al.* [6] built a train delay prediction model based on Bayesian network; Yaghini*et al.* [7] proposed a high-precision ANN model to predict the delay of Iranian railway passenger trains; Ping *et al.* [8] established a deep learning model for predicting the train delay time based on

RNN. Most of these researches focus on whether one train is delayed. Train delay is affected by many factors, such as route fault, train and communication network fault, extreme weather, passenger flow and on-site dispatch. The prediction accuracy will be reduced without considering these factors. Besides, they rarely consider both the temporal and spatial properties of trains and routes. In the train operation, the cumulative effect caused by delay is obvious, and different routes in some junction stations will cause different effects.

Different from the above researches, this paper does not predict the delay of one train, because if the delay of one train leads to the delay of other trains, the specific dispatching decision is made by the train dispatching department, which depends on the experience and knowledge of dispatchers. On the contrary, we predict the number of delayed trains in each period for each station, which is more valuable for train dispatching. The departure time of the delayed train is decided by the dispatcher on site. For example, in Beijingnan station (Beijing), there are four trains (the number is t1, t2, t3, t4) to Shanghai, Taiyuan and Wuhan respectively. Table I shows the departure information of these four trains. Due to the extreme weather, the trains are delayed. The station dispatcher may give priority to trains t1 and t4 to Shnghai based on the station environment (such as passenger flow).

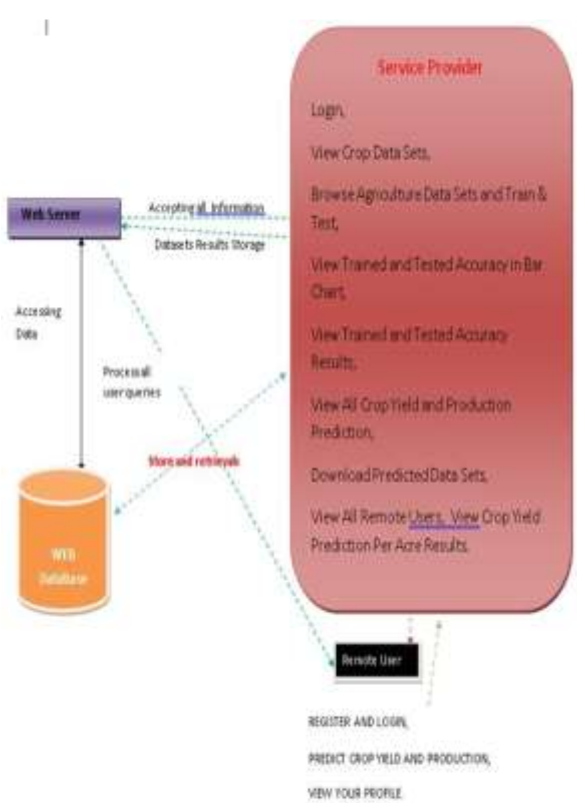
It can be seen from the above example that it is of little significance to predict the specific delay time of one train. Predicting the number of delayed train (collective cumulative effect) in a certain period is more valuable, which can guide the dispatcher's decision-making. In addition, collective cumulative effect will also consider the external factors like extreme weather that cause train delay, avoiding inaccurate prediction caused by incomplete considerations.

Based on the above analysis, this paper builds a TSTGCN model to predict the total number of the delayed train in each railway station. More precisely, we predict the number of arrival delays to provide reference for train dispatching and emergency plans.

Compared to the existing work, our contribution can be summarized as follows:

The collective cumulative effect prediction for train dispatching under the delay scenario is proposed for the first time to the best of our knowledge. A collective cumulative effect prediction of train delay model TSTGCN is constructed to predict the arrival delays in one station in a certain period. The proposed model fully considers the temporal and spatial dependence. A real graph of China's highspeed railway network is constructed, which includes not only all the stations, but also the mileage information of the routes. A 16 week actual operation data set of China's high-speed railway is also built by us, containing 1,954,176 delay records from October 8, 2019 to January 27, 2020, 727 stations, and all the routes between the stations,. ANN, SVR, RF, LSTM baselines are compared with our TSTGCN, and mean absolute error (MAE), root mean squared error (RMSE) and mean absolute percentage error(MAPE) are used to evaluate the performance in train delay prediction.

System Architecture



METHODOLOGY

ALGORITHMS

Decision tree classifiers

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decision making knowledge from the supplied data. Decision tree can be generated from training sets. The procedure for such generation based on the set of objects

(S), each belonging to one of the classes C1, C2, …, Ck is as follows:

Step 1. If all the objects in S belong to the same class, for example Ci, the decision tree for S consists of a leaf labeled with this class

Step 2. Otherwise, let T be some test with possible outcomes O1, O2,..,On. Each object in S has one outcome for T so the test partitions S into subsets S1, S2,..,Sn where each object in Si has outcome Oi for T. T becomes the root of the decision tree and for each outcome Oi we build a subsidiary decision tree by invoking the same procedure recursively on the set Si.

Gradient boosting

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.[1][2] When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest.A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

K-Nearest Neighbors (KNN)

Simple, but a very powerful classification algorithm. classifies based on a similarity measure. Nonparametric. Lazy learning. Does not "learn" until the test example is given. Whenever we have a new data to classify, we find its K-nearest neighbors from the training data

Example

Training dataset consists of k-closest examples in feature space Feature space means, space with categorization variables (non-metric variables). Learning based on instances, and thus also works lazily because instance close to the input vector for test or prediction may take time to occur in the training dataset.

Logistic regression Classifiers

Logistic regression analysis studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name logistic regression is used when the dependent variable has only two values, such as 0 and 1 or Yes and No. The name multinomial logistic regression is usually reserved for the case when the dependent variable has three or more unique values, such as Married, Single, Divorced, or Widowed. Although the type of data used for the dependent variable is different from that of multiple regression, the practical use of the procedure is similar. Logistic regression competes with discriminant analysis as a method for analyzing categorical response variables. Many statisticians feel that logistic regression is more versatile and better suited for modeling most situations than is discriminant analysis. This is because logistic regression does not assume that the independent variables are normally distributed, as discriminant analysis does. This program computes binary logistic regression and multinomial logistic regression on both numeric and categorical independent variables. It reports on the regression equation as well as the goodness of fit, odds ratios, confidence limits, likelihood, and deviance. It performs a comprehensive residual analysis including diagnostic residual reports and plots. It can perform an independent variable subset selection search, looking for the best regression model with the fewest independent variables. It provides confidence intervals on predicted values and provides ROC curves to help determine the best cutoff point for classification. It allows you to validate your results by automatically classifying rows that are not used during the analysis.

Naïve Bayes

The naive bayes approach is a supervised learning method which is based on a simplistic hypothesis: it assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature .

Yet, despite this, it appears robust and efficient. Its performance is comparable to other supervised learning techniques. Various reasons have been advanced in the literature. In this tutorial, we highlight an explanation based on the representation bias. The naive bayes classifier is a linear classifier, as well as linear discriminant analysis, logistic regression or linear SVM (support vector machine). The difference lies on the method of estimating the parameters of the classifier (the learning bias). While the Naive Bayes classifier is widely used in the research world, it is not widespread among practitioners which want to obtain usable results. On the one hand, the researchers found especially it is very easy to program and implement it, its parameters are easy to estimate, learning is very fast even on very large databases, its accuracy is reasonably good in comparison to the other approaches. On the other hand, the final users do not obtain a model easy to interpret and deploy, they does not understand the interest of such a technique.

Thus, we introduce in a new presentation of the results of the learning process. The classifier is easier to understand, and its deployment is also made easier. In the first part of this tutorial, we present some theoretical aspects of the naive bayes classifier. Then, we implement the approach on a dataset with Tanagra. We compare the obtained results (the parameters of the model) to those obtained with other linear approaches such as the logistic regression, the linear discriminant analysis and the linear SVM. We note that the results are highly consistent. This largely explains the good performance of the method in comparison to others. In the second part, we use various tools on the same dataset (Weka 3.6.0, R 2.9.2, Knime 2.1.1, Orange 2.0b and RapidMiner 4.6.0). We try above all to understand the obtained results.

7.6 Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

The first algorithm for random decision forests was created in 1995 by Tin Kam Ho[1] using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

An extension of the algorithm was developed by Leo Breiman and Adele Cutler, who registered "Random Forests" as a trademark in 2006 (as of 2019, owned by Minitab, Inc.).The extension combines Breiman's "bagging" idea and random selection of features, introduced first by Ho[1] and later independently by Amit and Geman[13] in order to construct a collection of decision trees with controlled variance.

Random forests are frequently used as "blackbox" models in businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration.

Algorithm 1: Pseudo code for the random forest algorithm

```
To generate c classifiers:
for i = 1 to c do
    Randomly sample the training data D with replacement to produce D_i
    Create a root node, N_i containing D_i
    Call BuildTree( N_i )
end for

BuildTree(N):
if N contains instances of only one class then
    return
else
    Randomly select x% of the possible splitting features in N
    Select the feature F with the highest information gain to split on
    Create f child nodes of N , N_1 ,..., N_f , where F has f possible values ( F_1 , ... , F_f )
    for i = 1 to f do
        Set the contents of N_i to D_i , where D_i is all instances in N that match
        F_i
        Call BuildTree( N_i )
    end for
end if
```

SVM

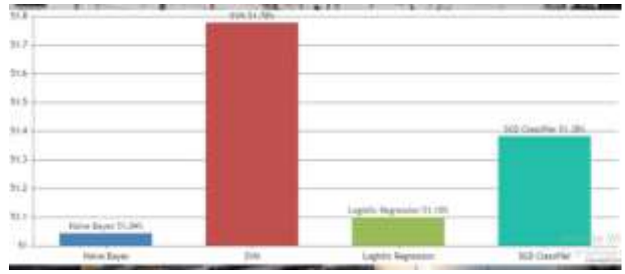
In classification tasks a discriminant machine learning technique aims at finding, based on an *independent and identically distributed* (iid) training dataset, a discriminant function that can correctly predict labels fornewly acquired instances. Unlike generative machine learning approaches, which require computations ofconditional probability distributions, a discriminant classification function takes a data point *x* and assignsit to one of the different classes that are a part of the classification task. Less powerful than generativeapproaches, which are mostly used when prediction involves outlier detection, discriminant approachesrequire fewer computational resources and less training data, especially for a multidimensional featurespace and when only posterior probabilities are needed. From a geometric perspective, learning a classifieris equivalent to finding the equation for a multidimensional surface that best separates the different classesin the feature space.

SVM is a discriminant technique, and, because it solves the convex optimization problem analytically, it always returns the same optimal hyperplane parameter—in contrast to *genetic algorithms* (*GAs*) or perceptrons, both of which are widely used for classification in machine learning. For perceptrons, solutions are highly dependent on the initialization and termination criteria. For a specific kernel that transforms the data from the input space to the feature space, training returns uniquely defined SVM model parameters for a given training set, whereas the perceptron and GA classifier models are different each time training is initialized. The aim of GAs and perceptrons is only to minimize error during training, which will translate into several hyperplanes' meeting this requirement

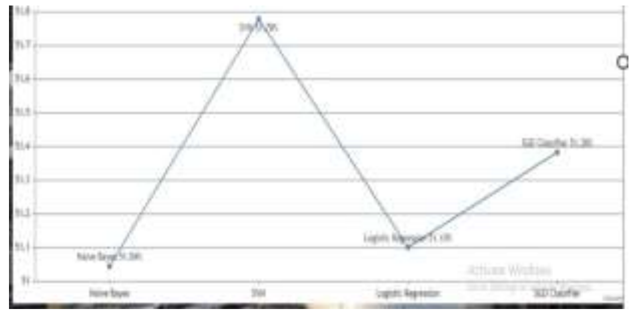
Algorithm 1: SVM

1. Set $Input = (x_i, y_i)$, where $i = 1, 2, \cdots, N, x_i = R^n$ and $y_i = \{+1, -1\}$.
2. Assign $f(X) = \omega^T x_i + b = \sum_{i=1}^{N} \omega^T x_i + b = 0$
3. Minimize the QP problem as, $min\ \varphi(\omega, \xi) = \frac{1}{2}\|\omega\|^2 + C.(\sum_{i=1}^{N} \xi_i)$.
4. Calculate the dual Lagrangian multipliers as $min\ L_p = \frac{1}{2}\|\omega\|^2 - \sum_{i=1}^{N} x_i y_i (\omega x_i + b) + \sum_{i=1}^{N} x_i$.
5. Calculate the dual quadratic optimization (QP) problem as $max\ L_D = \sum_{i=1}^{N} x_i - \frac{1}{2}\sum_{i,j=1}^{N} x_i x_j\ y_i\ y_j (x_i, x_j)$.
6. Solve dual optimization problem as $\sum_{i=1}^{N} y_i x_i = 0$.
7. Output the classifier as $f(X) = sgn(\sum_{i=1}^{N} x_i y_i (x \cdot x_i) +$ /.

RESULT ANALSIS



Prediction of Train Time Delay Bar Graph



Prediction of Train Time Delay Line Graph

Prediction of Train Time Delay Line Graph

CONCLUSION

According to the spatio-temporal characteristics anddynamic spatio-temporal correlation of highspeed train operationdata, this paper builds a TSTGCN model based onattention mechanism to predict the train arrival delay cumulative effect for railway dispatching. The model combinesspatio-temporal attention mechanism and spatio-temporal convolutionto capture the spatio-temporal characteristics of trainoperation data, so as to achieve more accurate prediction. In the experimental stage, we compare our TSTGCN withANN, SVR, RF and LSTM models, and use MAE, RMSEand MAPE to evaluate the prediction effect of these models. The experimental results show that TSTGCN is clearly betterfor the train delay cumulative effect prediction for traindispatching.

FUTURE ENHANCEMENT

Advanced Graph Convolutional Architectures

Dynamic Graph Structures: Implement techniques to dynamically adjust the graph structure based on evolving train network conditions, such as varying train speeds, maintenance schedules, and track conditions. Attention Mechanisms: Introduce attention mechanisms within the ST-GCN to prioritize relevant spatial and temporal features, enhancing the model's ability to focus on critical factors influencing delays. Incorporation of External Data Sources Weather and Traffic Data: Integrate real-time weather and traffic data into the model to account for external factors impacting train schedules, such as adverse weather conditions and traffic congestion near rail intersections.

Historical Data Analysis: Utilize historical train delay data to identify long-term patterns and trends, enabling the model to make more informed predictions over extended periods.

3. Multi-Modal Learning

Combination of Data Sources: Explore the integration of multiple data modalities, such as video feeds from train cameras, passenger load information, and maintenance logs, to provide a comprehensive view for delay prediction and dispatching decisions.

Fusion Techniques: Develop fusion techniques that combine information from different modalities effectively, leveraging the strengths of each data source to enhance prediction accuracy.

4. Real-Time Adaptation and Learning

Online Learning Strategies: Implement techniques for online learning and adaptation, allowing the model to continuously update and improve its predictions based on incoming real-time data streams and feedback.

Incremental Training: Develop methodologies for incremental training to efficiently incorporate new data while preserving the model's learned knowledge and avoiding catastrophic forgetting.

5. Interpretable Models and Decision Support

Model Explainability: Enhance the interpretability of the ST-GCN model outputs, providing insights into which factors most significantly contribute to predicted delays. This transparency can aid operators in understanding model decisions and adjusting dispatching strategies accordingly. Decision Support Systems: Integrate the delay prediction model into a broader decision support system that offers actionable recommendations to dispatchers, facilitating proactive management of train schedules and resources. 6. Scalability and Deployment

Edge Computing: Optimize the model for deployment on edge devices or within distributed computing environments near train stations or along tracks, reducing latency and enhancing scalability. Cloud Integration: Explore cloud-based solutions for scalability and resource management, enabling efficient handling of large-scale data processing and model training tasks.

7. Evaluation Metrics and Performance Monitoring

Comprehensive Metrics: Define and utilize comprehensive evaluation metrics beyond prediction accuracy, such as precision-recall curves, to assess model performance across different delay severity levels and operational scenarios.

Continuous Monitoring: Implement robust monitoring mechanisms to track model performance over time, detecting drifts in prediction accuracy and ensuring timely model retraining or recalibration as necessary.

8. Regulatory and Safety Considerations

Safety-Critical Applications: Ensure compliance with safety regulations and standards in railway operations, particularly for systems involved in critical decision-making processes such as train dispatching.

Ethical Considerations: Address ethical considerations related to data privacy, fairness, and transparency in deploying AI-driven systems in railway operations.

9. User Interface and Interaction Design

Intuitive Interfaces: Design user-friendly interfaces for operators and stakeholders to visualize model outputs, explore scenarios, and interact with the system effectively during decision-making processes. Feedback Mechanisms: Incorporate mechanisms for user feedback to improve system usability and address specific operational challenges encountered in real-world deployment.

REFERENCES

[1] S. Jing, "Research on delay prediction of high speed railway train based on data analysis," Ph.D.

dissertation, Southwest Jiaotong Univ., Chengdu, China, 2019.

[2] R. G. Xu, "Multiple traffic jams in full velocity difference model with reaction-time delay," *Int. J. Simul. Model.*, vol. 14, no. 2, pp. 325–334, Jun. 2015.

[3] S. Milinkovi´c, M. Markovi´c, S. Veskovi´c, M. Ivi´c, and N. Pavlovi´c, "A fuzzy Petri net model to estimate train

delays," *Simul. Model. Pract. Theory*, vol. 33, pp. 144–157, Apr. 2013.

[4] N. Markovi´c, S. Milinkovi´c, K. S. Tikhonov, and P. Schonfeld, "Analyzing passenger train arrival delays with support vector regression," *Transp. Res. C, Emerg. Technol.*, vol. 56, pp. 251–262, Jul.
2015.

[5] F. Corman and P. Kecman, "Stochastic prediction of train delays in realtime using Bayesian networks," *Transp. Res. C, Emerg. Technol.*, vol. 95, pp. 599–615, Oct. 2018.

[6] J. Lessan, L. Fu, and C. Wen, "A hybrid Bayesian network model for predicting delays in train operations," *Comput. Ind. Eng.*, vol. 127, pp. 1214–1222, Jan. 2019.

[7] M. Yaghini, M. M. Khoshraftar, and M. Seyedabadi, "Railway passenger train delay prediction via neural network model," *J. Adv. Transp.*, vol. 47, no. 3, pp. 355–368, Apr. 2013.

[8] H. Ping, W. Chao, L. Zhongcan, Y. Yuxiang, and P. Qiyuan, "A neural network model for realtime prediction of high-speed railway delays," *China Saf. Sci. J.*, vol. 29, no. S1, pp. 24–30, 2019.

[9] W. Chao, L. Zhongcan, H. Ping, T. Rui, M. Weiwei, and L. Li, "Progress and perspective of datadriven train delay propagation," *China Saf. Sci. J.*, vol. 29, no. S2, p. 1, 2019.

[10] Y. Zhaoxia and D. Zhongying, "Simulation system of train delay propagation," *J. China Railway Soc.*, vol. 17, no. 2, pp. 17–24, 1995.

[11] W. Xin, N. Lei, and L. Wen-Jun, "Study on robustness of high-speed train working diagram based on EMU utilization," *Railway Transp. Economy*, vol. 36, no. 11, pp. 50–55, 2014.

[12] P. Kecman and R. M. Goverde, "Online data-driven adaptive prediction of train event times," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 465–474, Feb. 2015.

[13] M. Carey and S. Carville, "Testing schedule performance and reliability for train stations," *J. Oper. Res. Soc.*, vol. 51, no. 6, p. 666, Jun. 2000.

[14] W. Chao, Y. Xiong, H. Ping, L. Zhongcan, and T. Youhua, "Review on conflict detection and resolution on railway train operation," *China Saf. Sci. J.*, vol. 28, no. S2, pp. 70–77, 2018.

[15] J. Yuan and I. A. Hansen, "Optimizing capacity utilization of stations by estimating knock-on train delays," *Transp. Res. B, Methodol.*, vol. 41, no. 2, pp. 202–217, Feb. 2007.

[16] J. Guo, L. Meng, P. Kecman, and F. Corman, "Modeling delay relations based

on mining historical train monitoring data: A Chinese railway case," in *Proc. 6th Int. Conf. Railway Oper. Modelling Anal.*, 2015, pp. 23–26.

[17] W.-H. Lee, L.-H. Yen, and C.-M. Chou, "A delay root cause discovery and timetable adjustment model for enhancing the punctuality of railway services," *Transp. Res. C, Emerg. Technol.*, vol. 73, pp. 49–64, Dec. 2016.

[18] Z. Yua n, Q. Zhi-Ming, H. Kang, and F. Shan-Shan, "Forecast method of train arrival time based on random forest algorithm," *Railway Transp. Economy*, vol. 38, no. 5, pp. 60–63 and 79, 2016.

[19] L. Oneto *et al.*, "Train delay prediction systems: A big data analytics perspective," *Big Data Res.*, vol. 11, pp. 54–64, Mar. 2018.

[20] Z. Pu, M. Lingyun, and L. Baoxu, "Prediction of high-speed railway train delay evolution based on machine learning," *Electr. Eng.*, vol. 20, no. z1, pp. 1–8, 2019.

[21] P. Huang, C. Wen, L. Fu, Q. Peng, and Y. Tang, "A deep learning approach for multi-attribute data: A study of train delay prediction in railway systems," *Inf. Sci.*, vol. 516, pp. 234–253, Apr. 2020.

[22] L. Oneto *et al.*, "Advanced analytics for train delay prediction systems by including exogenous weather data," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2016, pp. 458–467.

[23] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland:
Springer, 2018, pp. 362–373.

[24] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–9.

[25] F. Ning, G. Sheng-Nan, S. Chao, Z. Qi-Chao, and W. Huai-Yu, "Multicomponent spatial-temporal graph convolution networks for traffic flow forecasting," *J. Softw.*, vol. 30, no. 3, pp. 759–769, 2019.

AUTHORS PROFILE

Mr. B. Suresh, currently working as an Assistant Professor in the Department of Master of Computer Applications, QIS College of Engineering and Technology, Ongole, Andhra Pradesh. His area of interest is Machine Learning, Cloud Computing and Programming Languages

SHAIK.PEDA KHASIM currently pursuing MasterofComputer Applications at QIS College of engineering and Technology, Ongole, Andhra Pradesh. He Completed B.Sc. in Computer science Acharya Nagarjuna University Guntur, Andhra Pradesh. His areas of interest are Java & Cloud computing.