2019 IEEE Intelligent Transportation Systems Conference (ITSC)
Auckland, NZ, October 27-30, 2019

Towards a General Prediction System for the Primary Delay in Urban Railways

Jianqing Wu[1], Luping Zhou[1,2], Chen Cai[3], Fang Dong[4], Jun Shen[1,5], Geng Sun[1]
[1] School of Computing and Information Technology
University of Wollongong, Wollongong, Australia
[2]School of Electrical and Information Engineering
University of Sydney, Sydney, Australia
[3]Advanced Data Analytics in Transport
Data 61, CSIRO, Sydney, Australia
[4]School of Computer Science and Engineering
Southeast University, Nanjing, Jiangsu, China
[5]Research Lab of Electronics, Department of EE and CS,
Massachusetts Institute of Technology, Cambridge, MA, 02139, USA
Email: jw937@uowmail.edu.au, luping.zhou@sydney.edu.au, chen.cai@data61.csiro.au,
fdong@seu.edu.cn, {jshen, gsun}@uow.edu.au

***Abstract*—**Nowadays a large amount of data is collected from sensor devices across the cyber-physical networks. Accurate and reliable primary delay predictions are essential for rail operations management and planning. However, very few existing 'big data' methods meet the specific needs in railways. We propose a comprehensive and general data-driven Primary Delay Prediction System (PDPS) framework, which combines General Transit Feed Specification (GTFS), Critical Point Search (CPS), and deep learning models to leverage the data fusion. Based on this framework, we have also developed an open source data collection and processing tool that reduces the barrier to the use of the different open data sources. Finally, we demonstrate an advanced deep learning model, the novel ConvLSTM Encoder-Decoder model with CPS for better primary delay predictions.

Keywords- Prediction, Primary Delay, Railways, GTFS, Long Short-Term Memory

I. INTRODUCTION

With the rapid development of Internet of Things (IoT), cloud and edge computing, Big Data analytics (BDA), and artificial intelligence (AI) technologies, an increasing number of AI-based systems have been implemented to solve practical problems in various fields, such as business, healthcare, biology, education, and transportation. A large volume of data is generated through the IoT devices, and they are stored on the pervasive cloud platforms. However, such volume, velocity and variety of data cannot be processed by conventional data processing algorithms and tools. The applications of BDA and AI play vital roles in handling the IoT based sensor data to provide better services for human production activities and daily life needs.

Currently, BDA and AI have increasingly attracted the attention of practitioners and researchers in aspects of rail transportation engineering [1]. For instance, studying and analysing delay propagation behaviour is essential for developing such practical applications. In rail networks, delay propagation refers to that once a delay occurs at one station or one line, it often causes consequent delays in multiple stations or multiple lines, and even leads to the interruption of the entire railway network. If we predict the single primary delay, we can prevent delays in advance. For example, let us assume that a train departs from Station A and passes through Stations B and C. When Station A has a 120-second delay, followed by Station B with a 130-second delay, Station C has a 140-second delay. It is worth noting that if the Station A's delay is alleviated or avoided, the Station B and C may produce a delay of less than 30 seconds due to the nature of train delay propagation. Under such a circumstance, it is said that the train passes through the stations A, B, and C on time, since the 30-second delay is allowed for on-time performance.

Current train delay prediction systems still use static rules, which are built and operated by domain experts based on classical statistics. Establishing a practical and accurate delay prediction system could provide useful information to significantly improve traffic management and dispatching processes underlying passenger information systems, freight tracking systems, nominal timetable planning, delay management [2]. However, most of the delay and prediction information obtained from the data could be useless for adjusting time tables to schedule real-time trains. This is because, if a train arrives at or departs from a station more than 30 seconds or 60 seconds later than the scheduled time, it is considered as a delay. The often-occurred small delays need to be studied by data analysts again, which is very time-consuming.

Additionally, there is a lack of traceback for the causality of predicted data. Thus, to establish an automated train delay prediction system, it should contain two major components: an AI-based component to deal with big data, and an expert system-based component to emulate the ability of human experts to reason the data causality.

As railway IoT systems generate a large amount of data every day, it is feasible to apply the concepts of machine learning and deep learning to establish data-driven models of train delay prediction. Yaghini et al. developed an artificial neural network (ANN) model to estimate train delay based on historical data [3]. Pongnumkul et al. proposed two algorithms to predict train arrival times at three train stations. The experiment was based on a moving average of historical travel times and the travel times of k-nearest neighbours (k-NN) of the last known arrival time [4]. Oneto et al. implemented shallow and deep Extreme Learning Machines (ELM) for forecasting train delays of a large scale network with weather information on the Apache Spark [5]. In the follow-up work, Oneto et al.evaluated the system on six months of train movement data from the entire Italian railway network [2].

Train delay prediction has been explored more and more along with open data becoming increasingly available. Transit agencies have published open datasets to remove barriers for information-sharing among developers, researchers and data analytic organisations. For example, General Transit Feed Specification (GTFS) provided detailed schedules and associated geographic information in an open data format[6]. Even though the initial aim of GTFS is to offer a unified data format for developing user-focused route and schedule planning software, it has also become a critical data source for researches on intelligent railway systems[7]. However, there are still many issues with the direct use of these data for the prediction of a train delay, such as a large amount of data duplication, inconsistent information, missing data, and lack of practical information integration.

In this paper, we target at bridging the aforementioned research gaps and propose a data-driven Primary Delay Prediction System (PDPS) framework to predict primary delays using GTFS static and real-time data.

The remainder of the paper is organised as follows. Section 2 introduces the description of primary delay prediction problem. Section 3 describes the proposed train primary delay prediction system framework. Section 4 presents how to use the GTFS static and the GTFS real-time data to build and test the proposed models. Section 5 summarizes our experimental results, and finally, Section 6 concludes briefly.

II. Primary Train Delay Prediction Problem

The train delays are divided into two categories: primary delays and flow-on delays. The flow-on delay, which also is referred to as the secondary delay, is caused by the primary delay, [8] [9]. From a system perspective, there are two approaches to prevent delays from spreading out, by either making a more robust timetable or avoiding the occurrence of primary delays[8]. During the peak hours in urban railways, trains are operated quite densely. Once a delay occurs, it could be easily propagated to the succeeding trains. Thus, if we can predict and reduce the primary delays, the propagated delays can be reduced or avoided accordingly. This leads to great alleviation of humans' effects on the traffic management system.

According to [2] and [10], a railway network is considered as a graph where nodes indicate a series of checkpoints $\mathrm{C} = \{\mathrm{C1}, \mathrm{C2}, \dots, \mathrm{Cn}\}$ successively connected. For any checkpoint C, a train arrives at the time $s_{t_A}^{C}$ and departs at a time $s_{t_D}^{C}$ in the scheduled timetable, where t denotes a timestamp. The actual arrival and departure times of a train are denoted as $a_{t_A}^{C}$ and $a_{t_D}^{C}$. The differences of $(a_{t_A}^{C} - s_{t_A}^{C})$ and $(a_{t_D}^{C} - s_{t_D}^{C})$ are defined as the arrival and departure delays respectively. A train is delayed if its delay is greater than the 30s (or 1 min), generally. Additionally, a dwell time is obtained by calculating the difference between the arrival and departure time ($a_{t_D}^{C} - a_{t_A}^{C}$), while a running time is gained by calculating the difference between the departure time of the current checkpoint and the arrival time of the next checkpoint $(a_{t_A}^{C+1} - a_{t_D}^{C})$.

The primary delay detection problem is to predict the checkpoints that will have the first delay, which will cause delays in succeeding checkpoints. If the delay occurs, we can quickly predict which stations will also have a primary delay in the future. Traffic operators based on the information reschedule the train network in a timely and accurate manner, thereby reducing the number of stations that are delayed, or even avoiding the train network failure. Therefore, the primary delay prediction is a crucial task in the field of the railway management system.

III. Primary Delay Prediction Framework

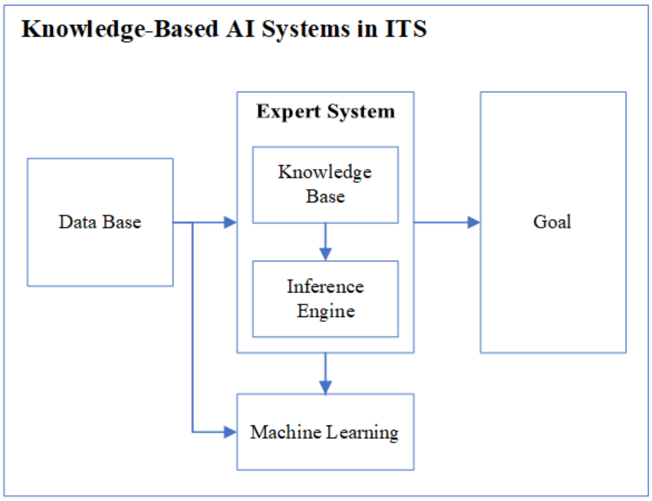

Figure 1. Knowledge-Based Artificial Intelligence System.

Typically, an expert system includes knowledge bases, inference engine, and user interfaces. An Inference Engine mainly contains two types of algorithms: Forward Chaining Algorithm (FCA) and Backward Chaining Algorithms (BCA) [11]. Inspired by Spring's work [11], a knowledge-based AI system in intelligent transportation systems (ITS) is derived as shown in Figure 1. An expert system is to mimic the intelligence and function of domain experts.

On the other hand, machine learning methods aim to apply the complex mathematical calculation-based algorithms to explore the relationships among large-scale data. To build a practical PDPS, the advantages of the expert system and machine learning should be integrated to achieve successful and scientifically useful predictions. The entire PDPS framework is roughly divided into four main modules: database, knowledge base, inference engine and machine learning component.

As depicted in Figure 2, each component is composed of multiple corresponding subcomponents. Firstly, we develop a data collector to collect real-time train data to establish a database. Secondly, for having a knowledge base, we implement a data preprocessing tool to fuse the data from two data sources, namely train schedules and associated geographic information. As structured information is created efficiently, we deploy the knowledge base on the cloud server for long-term data storage. Additionally, our model only uses data from the knowledge base to predict train delays, therefore, the overall calculation time of the entire system is greatly reduced. Thirdly, we propose a critical point search algorithm to integrate domain knowledge as an inference engine to categorize the data and find the primary delays. Finally, deep learning models are applied to achieve accurate predictions. As a result, the system extracts the valuable information, which are directly visualized to the system users for the planning and control rail services at the operational level. To the best of our knowledge, this is the first work that provides a comprehensive and conceptual framework for the design on the combination of expert systems in PDPS and deep learning. The system that performs causal reasoning in the delay prediction task, is illustrated in Figure 2.

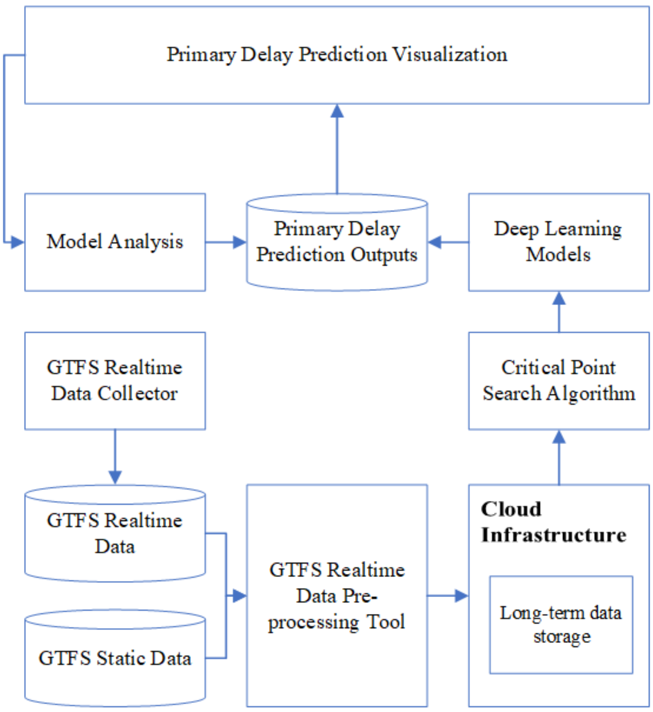

Figure 2. Framework of PDPS

A. *General Transit Feed Specification*

GTFS is developed for transit agencies to publish detailed transit schedules in an open data format; GTFS and GTFS-real-time specification enable transit agencies and operators to exchange both static and real-time public transit information [6]. Using GTFS data can conduct accessibility analysis, discover schedule padding, perform single or multiple transit system analysis, and investigate social equity in transportation planning [12] [13] [14] [15]. However, GTFS-realtime needs to be collected through the Application Programming Interface (API). The downloaded raw data needs to be preprocessed by data duplication, sorting and so on. It is merged with the information of GTFS-Static. Additionally, the new datasets are back up and stored on the cloud as research resources.

B. *Critical Point Search Algorithm*

The motivation to use the critical point search algorithm is to identify the primary delay and the flow-on delays[3] [4]. Our proposed algorithm is employed to the time series forecasting models to improve the prediction of the primary train delays, which are the causes of a lot of flow-on delays due to tracing causality. Since no existing studies, to the best of our knowledge, analyse the primary delay scenario with machine learning approaches and conduct delay classifications, this is a novel design in the train delay prediction field.

In order to predict the primary and flow-on delays, firstly we need to calculate the difference among the actual departure time $a_{t_D}^C$, the actual arrival time $a_{t_A}^C$, the scheduled departure time $s_{t_D}^C$ and the scheduled arrival time $s_{t_A}^C$. Subsequently, a few difference values are calculated from the following equations.

1) The difference D_1 between the acutal arrival time and the scheduled arrival time:

$$D_1 = a_{t_A}^C - s_{t_A}^C \quad (1)$$

2) The difference D_2 between the acutal arrival time at a timestep t and the acutal departure time at a timestep $t-1$:

$$D_2 = a_{t_A}^C - a_{t_D}^C \quad (2)$$

3) The difference D_3 between the scheduled arrival time at a timestep t and the scheduled departure time at a timestep $t-1$:

$$D_3 = s_{t_A}^C - s_{t_D}^C \quad (3)$$

4) The difference D_4 between D_2 and D_3:

$$D_4 = D_2 - D_3 \quad (4)$$

To find the primary points, an inference engine using forward chaining searches the critical points until it finds the points where $D_1 \geq$ *the first threshold value* V_1 and $D_4 \geq$ *the second threshold value* V_2. For an initial checkpoint, only D_1 is used to find the primary points. The Critical Point Search Algorithm is summarized as follows. By calculating differences of respective train arrival or departure times, the different category of delay or on-time points are added to the corresponding lists, W_1, W_2, W_3. The output W_1 denotes a list of primary delay points, W_2 a list of secondary delay points, and W_3 a list of running on-time points.

Algorithm Critical Point Search Algorithm

Require: Input all train data $\boldsymbol{R_t} = (\boldsymbol{R_1^t}, \boldsymbol{R_2^t} \dots \boldsymbol{R_N^t})$, and pre-defined thresholds $\boldsymbol{V_1}$ and $\boldsymbol{V_2}$
Output: $\boldsymbol{W_1}, \boldsymbol{W_2}, \boldsymbol{W_3}$
for each train $\boldsymbol{R} = (a_{t_A}^C,\ a_{t_D}^C,\ s_{t_A}^C,\ s_{t_D}^C)$ **do**
  $D_1 = a_{t_A}^C - s_{t_A}^C$
  if $D_1 >= V_1$:
    $D_4 = D_2 - D_3$
    if $D_4 >= V_2$:
      W_1
    else:
      W_2
  else:
    W_3
end for

C. *LSTM Neural Networks for Multi-Step Time Series Forecasting*

Recurrent Neural Network (RNN) is a commonly used and effective tool for sequence prediction problems. RNN includes many variants such as Long Short-Term Memory (LSTM) [16] and Gated Recurrent Unit (GRU) [17]. LSTM networks are capable of solving many tasks of the time series by using fixed-length time windows [18]. They have stacked to accurately model complex patterns of multivariate sequences [19]. Shi et al. introduced a convolutional LSTM (ConvLSTM) architecture, which is a combination of convolutional and LSTM layers [20]. Based on the state-of-the-art encoder and decoder design, Gehring et al. proposed a fully convolutional model structure for the sequence-to-sequence learning, which achieved superior performance over the strong recurrent models on machine translation tasks[21]. According to Shi et al.'s work [20], the ConvLSTM included the memory cell C_t, input gate i_t, forget gate f_t and output gate o_t as well as the output hidden state H_t, where $\circ$ indicates the Hadamard product. It used convolution structures directly in both the input-to-state and state-to-state transitions. Thus, the model is suitable to encode information for spatiotemporal data.

$$i_t = \sigma(W_{xi} X_t + W_{hi}H_{t-1} + W_{ci} \circ C_{t-1} + b_i)$$

$$f_t = \sigma\left(W_{xf} X_t + W_{hf}H_{t-1} + W_{cf} \circ C_{t-1} + b_f\right)$$

$$C_t = f_t \circ C_{t-1} + i_t \circ tanh(W_{XC}X_t + W_{hc}H_{t-1} + b_c)$$

$$o_t = \sigma(W_{xo} X_t + W_{ho}H_{t-1} + W_{co} \circ C_t + b_o)$$

$$H_t = o_t \circ tanh(C_t)$$

The input-to-state filters determine the output (W_{xi}, W_{xf}, W_{XC}, W_{xo}) and state-to-state filters (W_{hi}, W_{hf}, W_{hc}, W_{ho}). The input X at the time step t is the historical arrival or departure delay time. The final output is the predicted arrival or departure delay time, respectively. ConvLSTM does not have negative number predictions by using nonlinear activation function at each of ConvLSTM layers and Rectified Linear Unit (ReLU) activation function at the fully connected (FC) layer. It is vital for delay forecast models to predict positive times from the positive times of historical data.

D. Data Preparation

For evaluation, the proposed models are applied to a Sydney Train GTFS dataset from the NSW open data hub, which unlocks its data to share with developers, researchers, and data analytic organisation, and offers exciting opportunities for them to create an innovative solution for diverse stakeholders [22]. The raw data with a frequency range of 10 to 30 sec is extracted from the real-time GTFS that has a large amount of data every day. For example, collecting GTFS trip updates of Sydney Trains with a 10-sec frequency generate a dataset between 2 and 4 GB per day, which is preprocessed into a data set between approximately 3 and 6 MB dataset. Such open source data have great potential to be preprocessed to carry out a longitudinal study in rail transportation.

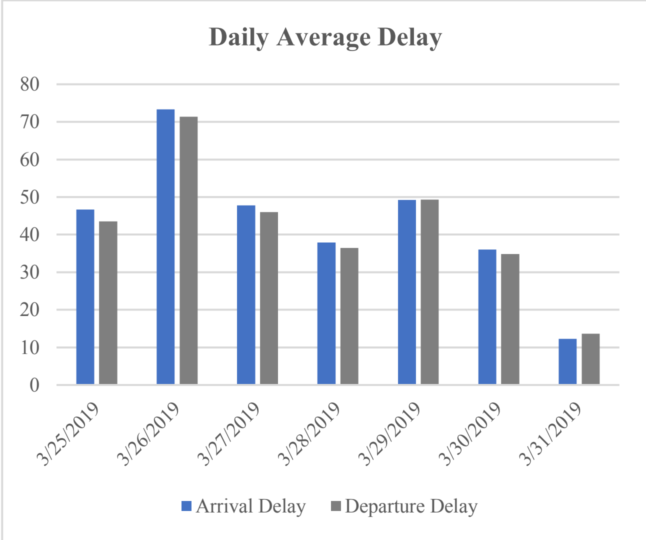

Figure 3: Daily average delay (sec)

After removing duplicated data, we pre-processed the dataset as follows. As shown in Figure 3, the means of the daily arrival delay and departure delay data for the entire railway network can be calculated. Specifically, according to Figure 3, we sorted the March 26 data and found many delays longer than 30 minutes. Moreover, ConvLSTM's future state of a cell in the grid is determined by the input and past state of its local neighbours [20]. Based on our experiments, the prediction error could increase significantly when an outlier is used as an input at a timestep close to the timestep of the output. Hence, such data cannot be harnessed to predict the next day's delay times. We proposed critical point search rules that can classify data efficiently and reasonably. The algorithm limits the upper and lower bounds of the data through a set of rules to split the dataset into three lists (W_1, W_2, W_3). It can be easily modified and extended to generate more categories of the list for the actual prediction, for instance, a list for special events. Furthermore, the entire train network consists of 8 lines, namely T1, T2…T8; where each line has multiple routes, and each route has multiple trips per day; and also the total number of nodes (stations) are different among the trips. Since each trip has a unique reference number, and there are no obvious systematic time-dependent patterns among the difference trips, if we simply split the dataset into the training and the test sets and then apply the deep learning model to predict the delays by using the datasets, the predicted results could be erroneous and not convincing. Besides, to utilize LSTM models for supervised learning in sequence data we need to predefine the number of subsequences and the length of subsequences, in order to determine the number of nodes we expected to forecast. Thus, for train delay prediction, the model only predicts a trip of one checkpoint at a time. If we would like to quickly complete calculations for all checkpoints, parallel computing can be used so that all calculations are carried out simultaneously.

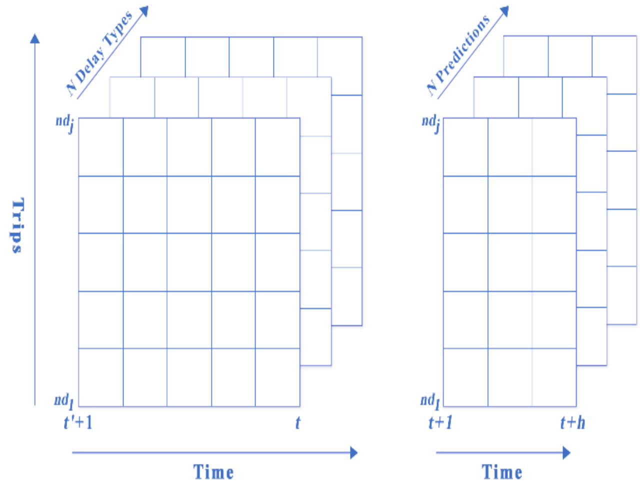

Figure 4. Input and Output Shapes

It is worth mentioning that the model we proposed is a generic model, which does not depend on a specific data set. The ConvLSTM model can learn long term correlation in a sequence and capture the spatiotemporal patterns by using good quality input data. Figure 4 shows that *N* delay categories of samples, where t' is an initial time, and each trip with a window of the historical time steps from $t' + 1$ to t. nd_1 to nd_j indicate the numbers of trips. The outputs include *n* delay predictions at *h* time-steps ahead, $t + 1$, …, $t + h$. For multiple trips, the input samples are sampled at non-fixed time resolutions to predict the outputs. Therefore, the data is transformed into a two-dimensional format oriented to supervised learning (train and test data in a tabular form). Specifically, our design is to split the trips into three tabular forms by using CPS. For the further study, when a checkpoint occurs a primary delay, we use Bayesian Learning to calculate the probability of a trip at subsequent checkpoints, at which events occur (the primary delay, secondary delay or on time running). The framework from this paper can be applied to generate a delay prediction model to estimate the arrival delay time or departure delay time for each type of events.

IV. Accuracy Analysis and Model Comparison

Root mean square error (RMSE) and mean absolute error (MAE) are applicable measures to evaluate the efficiency of the proposed prediction models. They have been defined as indicated in Eq. 5 and Eq. 6, where y_t is the actual times for sample *t* and $\hat{y}_t$ is the predicted times. As the multi-time-step model predicts train delays for all *r* trips for the next *n* time-steps, both y_t and $\hat{y}_t$ have the dimensionality $h \times r \times 1$.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_t - \hat{y}_t)^2} \quad (5)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_t - \hat{y}_t| \quad (6)$$

To obtain better predictions, we repeat the evaluation of the same model configuration on the same GTFS dataset and then estimate the average performance of the prediction models. For this experiment, we explored the patterns of train delays on weekdays. Table 1 shows the results of the trip number “146U” at “Seven Hills Station Platform 2” delay forecast, using GTFS data between January 29, 2019, and April 2, 2019, and GTFS-Static data. The advantage of integrating GTFS data is that we have more information about each station, such as station name, coordinates, node number, route name and so on. As evidenced by the results, except for the slight difference in the performance of CNN, the performance of three types of LSTM does not have much different. Our results are consistent with Greff et al.’s findings as well [23].

Table 1: Results of the models without CPS

Model	MAE (sec)		RMSE (sec)	
	Mean	SD	Mean	SD
CNN	84.51	4.52	136.13	3.83
Pure LSTM	80.61	1.25	133.17	1.26
CNN-LSTM	79.64	1.90	134.08	1.84
ConvLSTM	79.49	0.83	133.61	0.60

After applying CPS to find higher than 40-sec primary delays for the trip number “146U” at an initial station, “Emu Plains Station Platform 2”, the results of the proposed models indicate the different results. The main reason for using the different station is that CPS can remove the outliers at the same station, which means that the non-primary delayed data is not considered for primary delay prediction. Hence, the forecast result is improved.

Although Pure LSTM performs well on the given dataset, we found CNN, Pure LSTM, and CNN-LSTM perform negative values for delay predictions, which are abnormal values. Additionally, ConvLSTM's mean and standard deviation (SD) is higher than Pure LSTM’s in Table 2, whereas it has the smallest SD in Table 1. To sum up, ConvLSTM is more stable than other models to make predictions based on data with large residuals. Notably, it also performs accurate forecasts that are closer to the ground truth.

Table 2: Results of the proposed models

Model	MAE (sec)		RMSE (sec)	
	Mean	SD	Mean	SD
CNN	48.63	14.08	53.65	14.75
Pure LSTM	16.82	1.19	18.66	1.18
CNN-LSTM	34.97	4.53	37.62	5.29
ConvLSTM	37.56	3.48	42.63	3.89

In predicted primary train delay results, the algorithm assumes that all the predicted train running is the same as the actual train running time and this assumption is unrealistic. The recommended algorithm would be sensitive to the values of V_1 and V_2. Normally, V_1 or V_2 should be greater than 30 or 60 seconds. Our proposed model could capture long term correlation in sequence learning. Inspired by Yamamura’s work [8], as the prediction error exists, to accurately find the primary delay, the value of an offset needs to be calculated and be involved with the predicted output data. Therefore, to develop a primary delay prediction system, V_1, V_2, offset should be suggested by the domain experts, who can estimate the values based on reality.

V. Conclusion

This paper has proposed a PDPS system framework for train delay prediction and identified the feasibility of using GTFS data for such studies. The system framework includes the GTFS data pre-processing tool, the critical point search algorithm, and deep learning models. The combination is not only to deal with big data in railways but also to achieve causality for delay event classifications.

Our experiments classify the data of a single train line and forecast the corresponding stops of a single line. In the future, we will extend and apply the CPS to implement the data classification of the entire train network. Meanwhile, Bayesian Learning will be incorporated in the next stage for developing an online primary delay prediction system.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China under Grants, No. 61872079. We gratefully acknowledge the support of Sydney Trains and Data61, CSIRO. Wu also gratefully acknowledge financial support from the China Scholarship Council (201608320168).

REFERENCES

[1] F. Ghofrani, Q. He, R. M. Goverde, and X. Liu, "Recent applications of big data analytics in railway transportation systems: A survey," *Transportation Research Part C: Emerging Technologies,* vol. 90, pp. 226-246, 2018.

[2] L. Oneto *et al.*, "Train delay prediction systems: a big data analytics perspective," *Big Data Research,* vol. 11, pp. 54-64, 2018.

[3] M. Yaghini, M. M. Khoshraftar, and M. Seyedabadi, "Railway passenger train delay prediction via neural network model," *Journal of Advanced Transportation,* vol. 47, no. 3, pp. 355-368, 2013.

[4] S. Pongnumkul, T. Pechprasarn, N. Kunaseth, and K. Chaipah, "Improving arrival time prediction of Thailand's passenger trains using historical travel times," *2014 11th International Joint Conference on Computer Science and Software Engineering (JCSSE),* pp. 307-312, 2014.

[5] L. Oneto *et al.*, "Dynamic delay predictions for large-scale railway networks: Deep and shallow extreme learning machines tuned via thresholdout," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 47, no. 10, pp. 2754-2767, 2017.

[6] Google, "Google transit APIs," 2006.[Online]. Available: https://developers.google.com/transit/

[7] A. Owen and D. M. Levinson, "Developing a comprehensive US transit accessibility database," in *Seeing Cities Through Big Data*, Cham, Switzerland: Springer, 2017, pp. 279-290.

[8] A. Yamamura, M. Koresawa, S. Adachi, and N. Tomii, "Identification of causes of delays in urban railways," *Computers in Railways,* vol. 13, pp. 403-414, 2013.

[9] H. Huang, K. Li, and P. Schonfeld, "Real-time energy-saving metro train rescheduling with primary delay identification," *PloS one,* vol. 13, no. 2, p. e0192792, 2018.

[10] L. Oneto *et al.*, "Delay prediction system for large-scale railway networks based on big data analytics," in *2nd INNS Conference on Big Data: Advances in Big Data*, Cham, Switzerland: Springer, 2017, pp. 139-150.

[11] G. Spring, "Artificial Intelligence in Transportation: Information for Application," 2007.[Online]. Available: http://onlinepubs.trb.org/onlinepubs/circulars/ec113.pdf

[12] N. Wessel and M. J. Widener, "Discovering the space–time dimensions of schedule padding and delay from GTFS and real-time transit data," *Journal of Geographical Systems,* vol. 19, no. 1, pp. 93-107, 2017.

[13] J. Wong, "Leveraging the general transit feed specification for efficient transit analysis," *Transportation Research Record: Journal of the Transportation Research Board,* no. 2338, pp. 11-19, 2013.

[14] A. Owen and D. M. Levinson, "Modeling the commute mode share of transit using continuous accessibility to jobs," *Transportation Research Part A: Policy and Practice,* vol. 74, pp. 110-122, 2015.

[15] A. Guthrie, Y. Fan, and K. V. Das, "Accessibility scenario analysis of a hypothetical future transit network: social equity implications of a general transit feed specification–based sketch planning tool," *Transportation Research Record: Journal of the Transportation Research Board,* no. 2671, pp. 1-9, 2017.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[17] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Conference on Empirical Methods in Natural Language Processing*, pp. 1724-1734, 2014.

[18] F. A. Gers, D. Eck, and J. Schmidhuber, "Applying LSTM to time series predictable through time-window approaches," in *International Conference on Artificial Neural Networks: Artificial Neural Networks,* Berlin: Springer, pp. 193-200, 2002.

[19] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *European Symposium on Artificial Neural Network*, pp. 89-94, 2015.

[20] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, pp. 802-810, 2015.

[21] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 1243-1252, 2017.

[22] Transport for NSW, "General Transit Feed Specification (GTFS) and GTFS-Realtime (GTFS-R)," 2016.[Online]. Available: https://opendata.transport.nsw.gov.au/documentation

[23] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber, "LSTM: A search space odyssey," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222-2232, Oct. 2017.