Demystifying the Relationship Between Network Latency and Mobility on High-Speed Rails: Measurement and Prediction

Xiangxiang Wang*, Jiangchuan Liu*, Fangxin Wang†, Ke Xu‡
*School of Computing Science, Simon Fraser University, Canada
†Future Network of Intelligence Institute (FNii) and School of Science and Engineering
The Chinese University of Hong Kong, Shenzhen, China
‡Department of Computer Science, Tsinghua University, China
Email: xwa203@sfu.ca; jcliu@cs.sfu.ca; wangfangxin@cuhk.edu.cn; xuke@tsinghua.edu.cn

2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS) | 978-1-6654-1494-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/IWQOS52092.2021.9521359

***Abstract*—Recent years have seen increasing attention on building High-Speed Railways (HSR) in many countries. Trains running on the railways have a top velocity of up to over 300 km/hour. This makes it become a scenario with unstable connection qualities. In this paper, we propose a novel model that can accurately estimate the mobility status on HSR based on the changing patterns of network latency. Though various impact factors make the prediction complex, we however argue that the recent advance of deep learning applies well in our context, and further we design a neural network model that can estimate the moving velocity based on monitoring network latency's changing patterns in a short period. In this model, we use a new variable called Round Difference Time (RDT) to describe latency's changing patterns. We also use the Fourier Transform to extract the hidden time-frequency and use the generated spectrum for estimation. Our data-driven evaluations show that with suitable parameters, this model can get an accuracy of up to 94% on all three lines.**

***Index Terms*—velocity estimation, high-speed railway, Recurrent Neural Network**

I. Introduction

Recent years have seen increasing attention on building High-Speed Railways (HSR) in a wide range of countries to accommodate large-scale population migration given its comfortable and convenient traveling service. While with the rapid development of smart devices [1] and Internet of Thing (IoT) technology [2], demands of mobile communication with high quality are also growing. Therefore, many researchers focus on the HSR scenarios and propose many creative schemes [3]–[5] to improve the servicing quality of network connections.

Toward this goal, there however exists a first and foremost step, i.e., content providers need to distinguish the moving states of clients before providing services, which are naturally ignored by existing approaches. To solve this problem, servers can simply ask clients to provide their locational information, which further raises another two problems. On one side, this solution requires these clients to have the capacity of acquiring current locational information. It is easy for a portion of smart devices with GPS support (like smartphones) to report their locational information. Yet it is difficult for those IoT devices without GPS support due to the consideration of energy-saving and cost. On the other side, even if servers can obtain clients' current locational information, such schemes highly rely on cross-layer technology and only work in limited scenarios. Therefore, we need some new technology to help to determine current moving states without specific hardware support.

Fortunately, network connections between devices and remote servers provide rich information that can be used for mobility reasoning. Many researchers have realized that there has some associations between mobility and some metric indicators, like delivery rate, on HSR scenarios [6]–[14]. Yet because there are a lot of non-network factors that also affect these indicators, it is hard to decide how these relationships work. Different types of applications also have different expressions on the same indicator. Therefore, there still exists a huge gap between mobility understanding and rich network connections, with few research efforts focused on it.

In this paper, based on previous measurement works, we investigate several indicators that may be helpful for exploring the relationship. Some of them, like delivery rate, have strong relevance to current application types, and the others, like network latency, are closely related to the transmitting paths. From all these indicators, we find out that the change patterns of network latency in a short period is a suitable choice. On one side, network latency includes network propagation delay and queueing delay. For connections with a low delivery rate, it is almost impossible to introduce new congestion into the network. Network propagation delay has little relation with current application types. Therefore, for these mice streams, network latency is irrelevant to the application types and its standard deviation in a time window also has nothing to do with service types. On the other side, although network latency has a strong dependency on the transmitting path, we can view the latency spending on the wired paths in several seconds is constant. Hence in a short period, the change patterns of network latency only rely on the wireless paths. Such aforementioned characteristics make it a suitable indicator to explore the implicit relationship.

To explore the potential relationship and analyze how it

works, we design a measurement experiment on HSR scenarios. We choose the Beijing-Tianjin inter-city railway (BTR) routes to analyze network latency with traveling. We conduct this by sending packets to remote servers. The payload of each packet is low so that it will not introduce any new network congestion. The transmitting latency of these packets is captured as samples to represent the network latency. In our experiment, we derive some key observations based on the analysis of the HSR scenarios. In the 300 kilometer journeys, smart devices experience handovers for every 30 seconds. The network latency also oscillates sharply, with a maximum value of up to 10 seconds, and it seems to have no patterns. But because bullet trains always run on the known railway routes under the same running standard, their moving velocity is strongly related to their location. Meanwhile, relative to the railways, geographical distributions of basestations in a certain location are also fixed. When trains run in the same areas, smart devices have a higher probability to experience similar cellular networks, which makes network latency have similar changing patterns. If network latency always has similar patterns, we can use these to infer its moving velocity based on a similar location.

We also design a new model to verify this relationship, which can be used to estimate mobility based on the change patterns of network latency. In this model, we use a new variable called Round Difference Time (RDT) to describe the changing patterns. Because many other factors(like terrain information) also have an impact on the patterns, we need to extract useful information according to these factors. Fourier transform is a classic and widely used method that can be applied to extract frequency features from sequential data. We use the Short-Time Fourier Transform (STFT) method to do this and generate a spectrum with a given set of samples in a short time window. Recurrent Neural Network (RNN) is used to estimate the mobility with a given spectrum. Our trace-driven evaluations show that this model has an accuracy of at least 94% on each route. In summary, the contributions of this paper can be summarized as follows.

- We define a new metric, called Round-Difference Time (RDT). RDT has a strong relationship with wireless paths, and we use RDT to extract information related to high-speed scenarios.
- We conduct extensive field measurement on high-speed rails. Our measurement results show that handovers have a major impact on latency. The handovers happen often with the increasing N/P ratio. We also find out that with higher traveling speed, handover happens more frequently, in which burst increment and continuous decrement often happen together. This can be used to determine the current running status.
- We use Short Time Fourier Transform (STFT) to extract hidden information from the RDT series and an LSTM-based model is used for estimating. Our evaluations show that this model can achieve at least 94% accuracy. We evaluate performance with different configuration combinations and the number of hidden neurons and the selection of time windows has an outstanding impact on the final accuracy.

The rest of this paper is organized as follows. We discuss how we select the indicator in Section II. In Section III we will analyze the potential relationship between network latency and mobility. Then we propose the new model in Section IV to verify this relationship and evaluate its performance in Section V. Next, we discuss how it works with the 5G network in Section VI. Lastly we introduce related work in Section VII and conclude the whole paper in Section VIII.

II. Motivation and Metric

The rapid development of constructing high-speed rails in many countries brings a growing requirement for mobile communications. Due to the complexity of the rails' context and potential advancement of serving quality, many researchers start to focus on this scene and propose various schemes on improving the performance of mobile communications. Some of them propose new creative schemes to improve the performance of various services. However, although there are some achievements in this area, few of them have been used in practice. Because of lacking locational information, it is hard for the ISPs to determine whether the current served client is on a moving train or not. This makes the ISPs preferring to use traditional schemes, which are designed for static or low-speed scenes. They have good performance in these scenes, while they meet performance issues in HSR scenarios.

If ISPs can acquire locational information directly from the served customers, they could use this information to select the best scheme for each customer to get better performance. The simplest way is asking customers to send their positional information to the ISPs so that they can make a decision on a suitable scheme. However, more and more people try to protect their privacy, by hiding their location or providing fake information. This makes this method does not work. In addition, the locational information is often acquired in the physical layer, while most optimization schemes are realized in-network or transmitting layers. Acquiring and utilizing the movement information needs cross-layer technology support, which further limits the deployment of this method.

Using the normal delivery traffic is another way to estimate the speed. A previous study shows that the delivery latency has some relation with the trains' speed. But they do not point out how the speed affects the delivery. Many delivery metrics are affected by the train's running like the delivery rate is more unstable, and more packets meet out-of-time in a short period. There are many metric indices that can be used here, like delivery rate, network latency, packet loss rate, and so on. Many researchers have shown that these metrics behave differently from those in static or low-speed scenarios. However, these indices cannot be directly used, for some of them are related to other factors that could also affect them. To choose a suitable metric index, several conditions should be met.

1) Relevent to the delivery path: The selected metric should be related to the delivery path. It also should have a high correlation with the wireless paths, while has less relationship with the wired ones.
2) Irrelavent to other services: Other service applications that share the same wireless paths should not have a prominent impact on the selected metric.

The whole connection can be divided into two parts, wired and wireless paths. Previous research shows that the network metrics are mainly affected by the wireless paths. Thus, the first condition guarantees the selected metric contains some information about the users' location, while it is not affected by the wired path.

Because most users share the same frequency band to use cellular networks under the coverage of the same base station, other users' network behaviors could also affect some metrics. The second condition makes sure that the selected metric is not affected by other users.

The delivery rate cannot be used here, because they rely not only on the wireless paths but also on the wired paths. The packet loss rate is a good metric to evaluate the status of wireless paths, while most connections meet few loss events due to the settings of the huge buffer. We start from the network latency because it only relies on the delivery paths, and other users have less impact on it. The common latency (T) usually contains three parts, the latency on wired (T_d) and wireless paths (T_l), and the latency on processing packets (T_p).

$$T(t) = T_\mathrm{d}(t) + T_\mathrm{l}(t) + T_\mathrm{p}(t) \tag{1}$$

To exclude the affection caused by the wired path, we observe the network latency in a short time slot. The time for servers to process each packet is similar, and the wired paths are the same, which makes the t_{wired} also be similar. Thus, we define a new metric, called Round-Difference Time (RDT), as the metric. RDT can be defined by the latency difference in a short time slot (τ) by

$$RDT(t) = T(t) - T(t-\tau) \tag{2}$$

Although there are a bunch of articles talking about estimating the One-Way Delay (OWD), we here still use the Round-Trip Time (RTT) as the latency metric. RTT is an accurate metric, and it is easy to get from daily use.

III. HSR Measurement

We conduct a measurement experiment on HSR routes. Based on the observation of the network transmissions, there are no clear mathematical connections between the network latency and the mobility status, which renders the traditional model-based methods ineffective. The unique characteristics of bullet trains running on HSR routes provide an alternative perspective to understand such correlations. Specifically, various impact factors, such as cellular network coverage and handovers between two basestations, reveal rich information that can be utilized to infer the correlations between network latency and mobility.

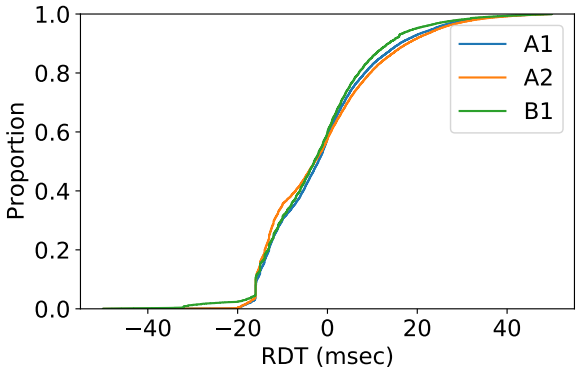


Fig. 1: Distributions of the RDT over three routes. We only show the distributions of RDT samples in the range of (-50, 50).

TABLE I: High-Speed Rails routes in the measurement expreiment

Route No.	End Stations	Middle Station	Running Time
A1	BJ-TJ	-	30 minutes
A2	BJ-TJN	-	30 minutes
B1	BJ-TJ	WQ	42 minutes

A. Measurement Deployment

We select the Beijing-Tianjin Inter-City Railway (BTR) to do the measurement. BTR is a famous route in China, as it connects two well-developed cities. As of 2019, it has provided rapid and comfortable inter-city traveling service for over 280 million people. One of the main reasons we choose BTR routes is that it has good cellular network coverage, which is an essential element in our experiment. Illustration of BTR routes is shown in Table I. BTR includes three routes, one is between Beijing South Railway Station and Tianjin Railway Station, and another is the same route with a docking station called Wuqing Station. The last one is between Beijing South Railway Station and Tianjin West Railway Station. To simplify the description, we call routes between Beijing South Railway Station and Tianjin Railway Station with and without docking station as Route A1 and A2 and call the other one as Route B1. Trains running on all three routes have a top velocity of about 350 km/hour (about 97 m/sec) and they need about 30 minutes running on route A1 and B1, and additional 12 minutes on route A2.

A laptop is used to send packets to public servers and record their transmitting latency. Instead of using TCP, we select `ping` command to do this. The main reason of using ICMP packets is that TCP transmissions will introduce new interference caused by other factors, like congestion control algorithms [15] and acknowledgment aggregation problem [16]. Each ICMP packet has a size of 64 bytes and this constructs a byte stream with a delivery rate of about 6 KB/sec, which we believe cannot introduce new network congestion. This satisfies previous requirements.

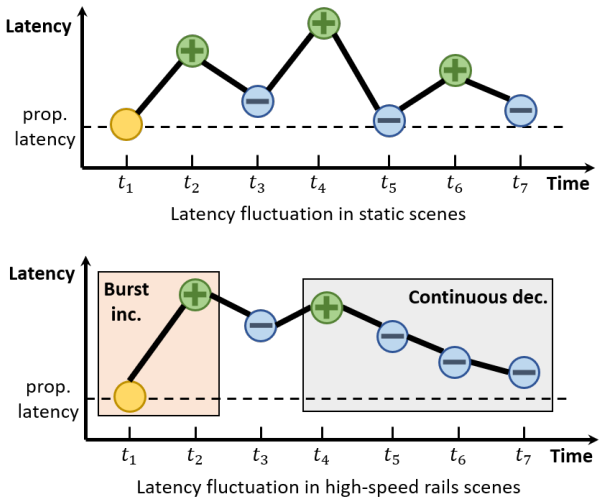


Fig. 2: Illustration of the proportions of positive and negative samples in static and high-speed rails scenes. The yellow circle represents the initial latency sample. The green and blue circles are the positive and negative samples, respectively.

We use a smartphone to share the cellular network with the laptop by the USB sharing function. Here, the reason that we select USB sharing instead of WiFi sharing is that there is also WiFi service supplying by the trains. This will cause WiFi contentions. Furthermore, we also use the smartphone to collect both mobile information like moving velocity and physical information like handover information.

B. Relationship Between RDT and Handover

When we observe the collected latency samples, we find out that each route has a serious heavy-tail problem, in which the maximum latency is over 5 seconds for one packet transmitting. But if we only look at the RDT samples falling into the range between -50 msec to +50 msec, which contains over 98% samples in each route, we find out that their distributions are similar, as shown in Figure 1.

Then we count the proportions of the positive and negative samples in each route. In the static scenes, delivery latency should fluctuate around its propagation latency, which is its minimum value [17]. Thus, the number of positive and negative RDT samples should be similar. However, in our measurement, we find out that in each route, there are about 40% positive samples and about 56% negative samples. This shows two patterns of delivery latency in high-speed scenes. One is burst increment, in which delivery latency has a burst increment. The other is continuous decrement, where the latency of continuous packets is in descending order. Figure 2 shows the differences between these two kinds of latency fluctuation. The yellow circles represent the delivery latency of the initial packet. If the latency of the received packet is larger than that of the previous one, then we use a green positive mark to represent it, while a blue negative mark on the other side.

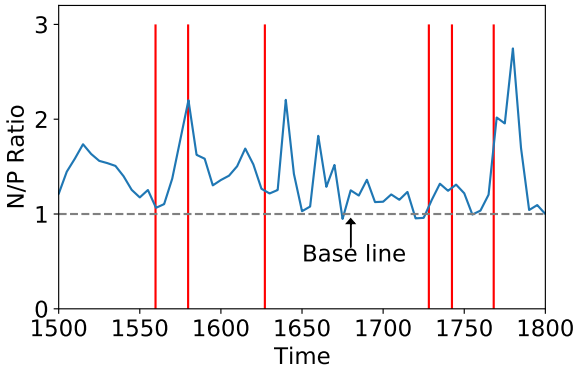


Fig. 3: Illustration of the relationship between N/P ratio and handovers. The grey baseline means the numbers of positive and negative samples are equal. Each red bar represent a basestation handover.

A reasonable explanation of this phenomenon is the base station handovers. In order to make it clear, we define the ratio between the numbers of the negative and positive samples as the **N/P Ratio**, and a larger ratio means more negative samples. When smartphone experiences handover, packets are temporary stored in the cache of the base station. After the handover finishes, the cached packets are sent to the devices. The prior packets experience a higher queueing latency, which results in a burst increment. Subsequent packets spend less time in queueing, which shows a continuous decrement. An example is shown in Figure 3, the blue line shows the trend of N/P ratio with the train's traveling, and each red bar represents a base station handover event. In most cases, handover always happens with the increment of the N/P ratio.

To simplify the description of the next, we make definitions of these two patterns. In a period d, a burst increment happens if the delivery latency of the current packet is α ($\alpha > 1$) times the minimal latency. If the delivery latency of β ($\beta \geq 4$) continuous packets is in descending order, then we say a continuous decrement happens. α and β are called the *degree* of the patterns.

C. Relationship Between RDT and Traveling Speed

While we have found some relationship between RDT and base station handovers, the connection between RDT and train's traveling speed has not been built. Previous study [6] shows that smartphones experience base station handovers more frequently at high speed, and this can help us to construct a connection. Based on the definition of the Physical Cell Information (PCI), we can estimate handover according to the PCI records. PCI is an encoding method to give each base station a unique identifier in a given area. The PCI can be computed by

$$\text{PCI} = 3 \times \text{CellGroupID} + \text{CellNumber} \tag{3}$$

and we can use PCI to classify inner-handover, in which handover happens between two cells belonging to the same

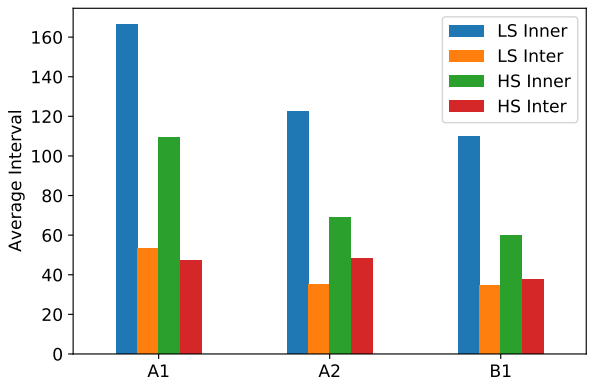


Fig. 4: Interval between two handovers. **LS** and **HS** represent low-speed and high-speed, respectively.

base station, and inter-handover, in which handover happens between two different base stations. We count the number of each kind of handover and calculate the average interval between two handovers. The result is shown in Figure 4. Either in low-speed or in high-speed scenarios, the intervals between two inter-handovers are similar, but inner-handover happens more frequently in high-speed scenarios.

Next, we consider the burst increment pattern. Suppose there are two burst increments with degree $\alpha_1 = 2$ and $\alpha_2 = 5$. Although either in low-speed or in high-speed, burst increment often happens with the base station handover, we find out that the pattern of degree α_2 happens more in high-speed. In other words, in high-speed scenarios, burst increment behaves more seriously than that in low-speed scenarios. The continuous decrement pattern has a similar result, a pattern with a higher degree is more likely to happen in high-speed scenarios.

However, even though we find some weak connections between RDT and train movement, we still have several challenges to solve. The first one is how to determine the value of α and β. These two values can reveal the base station handover, and this can be used for determining current movement status. An overlarge value makes it hard to find, while a tiny value brings noise that can influence the estimation accuracy. The other challenge is how to extract other hidden information from the RDT series. More information can help us to get a better estimation.

IV. Estimation Model

Our measurement results show that the trains' running speed can be estimated based on the two patterns of the RDT series. However, there are still some challenges. One is how to extract other information from the RDT series to make a better estimation. The other one is how to determine the value of the hyperparameters α and β. To solves these two problems, we use Fourier Transform to turn the series of RDT from the time domain into the frequency domain. Besides, instead of deciding certain values of these two parameters, we use the LSTM model for estimation.

A. RDT Spectrum

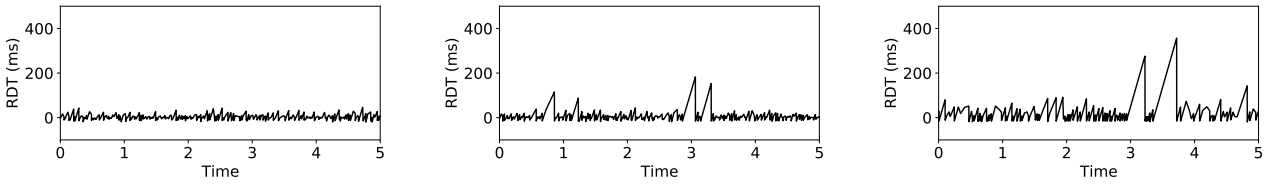
In our measurement study, we find out that there are possible more burst increment and continuous decrement patterns when trains run at a higher speed, especially when handover happens. But only relying on these two patterns cannot help us to directly estimate current speed. To get a better estimation, we need more information. A common way to do this is using Short-Time Fourier Transform (STFT) or Wavelet methods. These methods transfer time series data into the frequency domain to get more information. Here, we use the STFT method to process the RDT series, and their results are called RDT spectrums. Burst increment patterns contain more information in the high-frequency area, making a hot spot shown in the high-frequency area on the spectrum. A higher intension means a higher increment of the delivery latency. On the other side, a continuous decrement contains more information on the low-frequency area, due to it has a longer duration. Besides, because continuous decrement patterns usually happen after burst increment patterns, it shows high intension on the low-frequency area.

[6] shows that delivery latency behaves differently within different trains' running speeds, thus we also analyze the delivery latency with different running speeds. In Figure 5 and 6, we show three examples of RDT series and their corresponding RDT spectrum in static, low-speed, and high-speed scenes. The data in the static scenes are collected when trains stop in the stations. Besides, the data with a train's running speed of 150 km/hour is used as the example in the low-speed scene, and the one collected with a speed of 300 km/hour is used as the example in the high-speed scene.

In Figure 5, the delivery latency fluctuates in a larger range with a higher running speed. Meanwhile, with a higher running speed, RDT series have more burst increment and continuous decrement, and there are more patterns with higher degrees. In Figure 6, we show the RDT spectrum corresponding to the RDT series in different scenes, respectively. In the static scene, because the delivery latency fluctuates in a normal range, and it has few burst increments and continuous decrement patterns, the hot areas mainly concentrate on the low-frequency areas with low intention. But when the trains run, because there are the delivery latency is much more unstable, spectrums show hot spots in high-frequency areas, which corresponds to the burst increment patterns. In low-frequency areas, there are also some hot spot areas, which is resulted from the continuous decrement pattern. The degrees of patterns also are shown in these spectrums. For the burst increment patterns, both the intension and the corresponding frequency bands show their degree, and for the continuous decrement patterns, a high intension means that there are a long series of packets having the delivery latency in descending order.

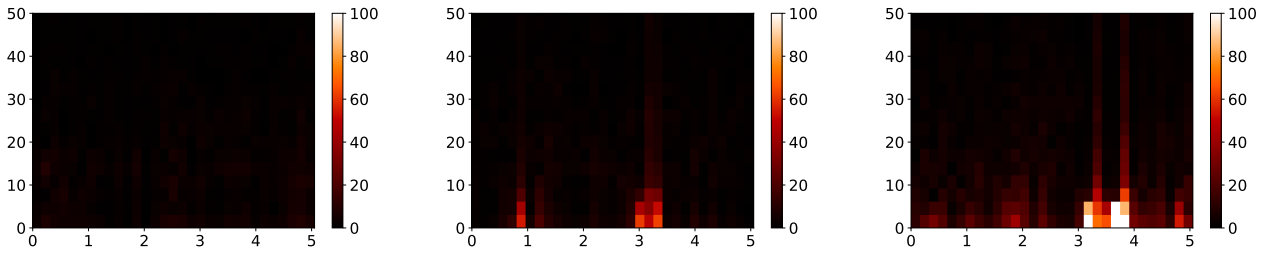
B. LSTM Model

Although we use the RDT spectrum to extract the hidden information from the original latency series, there is still a challenge of the selection on the hyperparameters α and β. Selecting suitable values for these parameters is important,



(a) Static (b) Low-speed (c) High-speed

Fig. 5: Examples of the RDT series captured in various scenes. The delivery latency fluctuates with a low range, while with a larger range in movement scenes. In high-speed scenes, there are much more burst increment and continuous decrement patterns, making the latency be more unstable.



(a) Static (b) Low-speed (c) High-speed

Fig. 6: Examples of RDT spectrum captured in static, low-speed, and high-speed scenarios. Because the delivery latency fluctuates in a larger range, there are more hot spots in the RDT spectrums.

because the real high-speed rails scenarios are complicated, and it is difficult to determine these values. Undersized values selection makes false recognition to the normal latency fluctuation as the two patterns, while it is difficult to capture these patterns when choosing oversized values. Besides, it is difficult to estimate the trains' running speed when several patterns are captured.

Instead of choosing certain values for the hyperparameters, we use the time-related information for estimation. Each row of the spectrum can be treated as a feature vector. The fluctuation of the delivery latency creates hot spots on the spectrums, and if the continuous areas of hot spots are observed, then we can estimate that the train is running at a high speed.

Based on this estimation method, Recurrent Neural Network (RNN) is a good selection, for it utilizes the statistical information to make an accurate estimation. Because the traditional RNN faces the gradient vanishing problem, we here use the Long-Short Term Memory (LSTM) as the model for estimation. LSTM shows good performance on estimation for a given time-series data, and our evaluations show that this model works well on estimation.

C. Speed Level

In our estimation model, we classify the speeds into 10 levels, and the speed range in each level is shown in Table II. The main reason we use speed level instead of directly estimating the speed is that an accurate estimate of current speed is not necessary. Previous study shows that the trains' speed does have an impact on the delivery latency, while a little change in the trains' speed does not have a great difference in the delivery latency. Thus only estimate the current speed level is enough for the servers to make a decision on the served clients. In our measurement experiments, we find out that using a range of 10 m/sec is good enough to satisfy the requirement.

TABLE II: Speed level and its corresponding speed range.

Level	Speed Range (m/sec)	Level	Speed Range (m/sec)
0	$[0, 10)$	5	$[50, 60)$
1	$[10, 20)$	6	$[60, 70)$
2	$[20, 30)$	7	$[70, 80)$
3	$[30, 40)$	8	$[80, 90)$
4	$[40, 50)$	9	$[90, 100)$

TABLE III: Structure of the LSTM model.

Layer	Name	Input / Output Dimension
1	LSTM	65 / 128
2	ReLU	128 / 128
3	BatchNorm2d	128 / 128
4	LSTM	128 / 128
5	ReLU	128 / 128
6	BatchNorm2d	128 / 128
7	FC	128 / 128
8	BatchNorm1d	128 / 128
9	FC	128 / 10

D. Estimation Model

The whole process is shown in Figure 7. Captured series of delivery latency is used as the raw input. It firstly processes

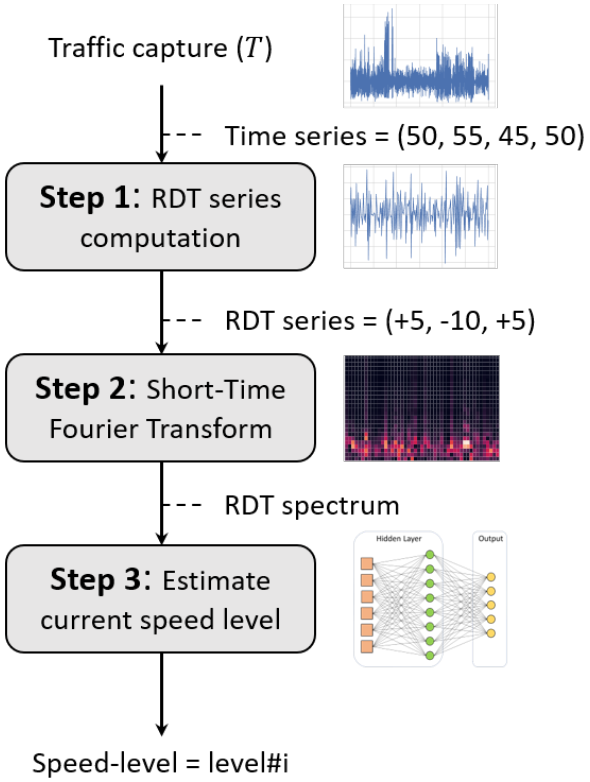


Fig. 7: The whole estimating process to estimate the train's speed level with a given series of delivery latency.

the series of latency into series of RDT, and it calculates the corresponding RDT spectrums. The spectrums are then passed into the LSTM model, and the model estimates the matching speed level. The structure of the LSTM model is shown in Table III. Here, we use a two-layer LSTM model to extract information. We use ReLU as the activation function for each layer, and we also add normalization operations after activating. Two full-connection layers are used for final estimation. The model outputs the estimation for the speed level.

V. Evaluation

In this section, we evaluate the performance of the estimation model. Our model is developed by the Pytorch [18]. We use a normal computer, with one CPU of Intel Core i7-3770 and a video card of GeForce GTX 1080 to accelerate the training process. This video card has 2560 CUDA cores for accelerating.

To have a better result from Fourier Transform, we choose a step size of one microsecond in the interpolation process. For the transform, we choose a window size of 128 samples with an overlap of 50% samples. Therefore, in the training process, each turn has an input vector with a length of 65 values. We train models for each route. For data collected on each route, we use 99% of them to train the model and the rest for testing. Finally, we use the estimating accuracy to evaluate the model's performance.

A. Number of hidden neurons

We choose GRU as the default network and compare the model's performance with a different number of hidden neurons. Results of each route are shown in Figure 8a. We test the model's accuracy with 32, 64, and 128 hidden neurons on the testing dataset. The lowest accuracy is over 80% and the highest accuracy of about 98% is achieved on route A2 with 128 hidden neurons. This result shows that the designed model does have the ability to estimate current movement velocity. Even with only 32 hidden neurons, these models can also have an accuracy of over 80%. With more hidden neurons, these models can gain better performance.

Besides, in the training process, we find out that for each 2000 training samples, the training time has an increment of about 12.5% with the number of hidden neurons increasing from 32 to 64, and about 11.1% with the increasing from 64 to 128. Therefore, increasing the number of hidden neurons does not make the training time increase too much but can effectively increase the accuracy of the trained models.

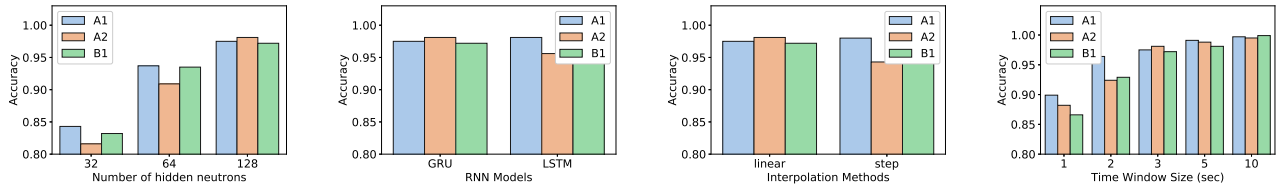
B. RNN model selection

Then we evaluate the accuracy of the designed model with different RNN model selections. We mainly evaluate the performance of two famous and traditional RNN models: GRU and LSTM models. The configurations for these models have 128 hidden neurons, the linear interpolation method, and a time window of 3 seconds. The result is shown in Figure 8b. Both RNN models have similar estimating results of over 95% accuracy, but GRU performs better than the LSTM model on routes A2 and B1. However, comparing with their running time, we find out that the LSTM method needs another about 7.4% training time. We believe this is caused by the number of parameters in a single RNN neuron. A traditional LSTM neuron needs three different gates to select useful information to remember while a GRU neuron only needs two gates. Therefore, GRU neurons consume less training time than that using LSTM neurons. Theoretically, LSTM neurons spend more time to gain better results. But in our case, its advantage is not obvious, while on some routes, it even has worse performance.

C. Interpolation methods

Next, we evaluate the impact of the interpolation method selection on the final accuracy. We choose linear and step functions as the interpolation functions to do this. The step function is widely used to illustrate network latency, but it loses gradient information. This introduces some noise into the spectrum generated by the STFT method. On the other side, the linear function can reserve most gradient information and reduce noise.

The result is shown in Figure 8c. We use GRU as default with 128 hidden neurons, a time window of 3 seconds. We find out that using the linear method for interpolation is better than that using the step method. Among them, the model on route A1 has little accuracy advantage on using the step function, while on the other two lines, the step function has about a



(a) (b) (c) (d)

Fig. 8: Accuracy of models on each route with different configurations.

2% accuracy gap with those using the linear method. We think this is that the linear method can reverse some gradient information, which can help the model to improve the final accuracy. Using the step method will generate horizontal lines with a burst when latency changes. But for Fourier transform, horizontal lines are hard to decompose, which makes the generated spectrum be useless for further training.

D. Time window

Lastly, we evaluate the results with different time window selections. We test the final accuracy with five different time window sizes on all three lines, including 1, 2, 3, 5, and 10 seconds. The result is shown in Figure 8d. With a longer time window size, we can have a better accuracy result. When we select a time window of one second, the best final result is achieved on route A1, with a value of lower than 90%. However, when we choose a longer time window, the final accuracy also keeps growing. When we select a time window of 10 seconds, the model can have an accuracy of up to 99.9%.

At the same time, adding the time window size also increases the time for processing in both Fourier Transform and testing process. When we use a time window of one second, the running time is about 40 seconds for each 2000 training sample. However, when we turn the time window into ten seconds, the running time increases to up to 300 seconds for each 2000 training sample.

VI. Further Discussion

Nowadays, the 5th generation (5G) cellular network has been started to be built in some countries. Therefore, a new question that how this model deal with the 5G network. We think the 5G network will not highly impact this model for two reasons. On one side, some countries just start to build the 5G network. Because the 5G network has a much smaller coverage range than that of the LTE network, deployment of the 5G network is mainly done in the central urban areas. But in suburban areas, due to the limitation of the basestations cost, it needs a long period for the carriers to deploy the 5G network, which results in a long period of using LTE network in these areas.

On the other side, also because of the consideration of the deployment cost, carriers prefer to use both Standalone (SA) and Non-StandAlone (NSA) schemes for the deployment of the 5G network. In areas covered by the 5G network with using NSA schemes, handovers of smart devices suffered on the traveling are much like those with LTE network. Therefore, even though in the 5G network, if it is using the NSA scheme to update the LTE network into the 5G network, the relationship between velocity and network latency still remains similar. Therefore, our model can still be used in the future.

VII. Related Work

There is some measurement research on HSR routes. [8] introduce the using LTE in HSR scenarios and challenges it faces [6]–[10], [19]. [6] measures the performance of using Multi-Path TCP [20], which is used for solving poor efficiency of using wireless transmitting bandwidth. [7] measures the performance of traditional TCP transmitting and tries to find out the relationship between handovers and TCP transmission's performance. [9] tries to model the delivery rate of TCP transmitting using the Reno congestion control algorithm on HSR routes. However, these measurements are mainly focused on TCP transmitting. Because there are plenty of factors that may affect its performance, including in both wireless and wired transmitting, it is hard to understand the reason how trains running on HSR affect the network latency.

There are also some works focusing on improving the performance of network transmitting. [12] proposes a new architecture to improve the performance of TCP transmitting. It uses multiple devices to relay transmitting packets and thus increases the robustness of transmissions. [13] proposes a new MPTCP schedule to improve performance with a high packet loss rate. [14] also designs a new MPTCP scheduler to be aware of lost packets. These schemes are helpful for improving transmitting performance on HSR routes. These works mostly focus on improving network transmission service, while our work well complements them in bridging the gap between mobility prediction and services.

With the rapid development of machine learning, especially deep learning methods, more and more researchers use machine learning methods to solve problems faced in daily life. [21] uses WiFi signals to recognize keystroke. [22] designs a method to extract 3D information using data from public resources. [23] proposes a method to capture hands' moving traces only based on WiFi signals. [24] designs a new method to estimate human behaviors based on WiFi signals. It also analyzes the relationship between signals and behaviors and proposes a formula to explain their relationship. However, because these methods highly rely on Doppler Effect, they

are mainly used in direct link scenarios like WiFi connections and are not suitable in our context.

There is also some research trying to construct connections between transmitting signals and moving states. [25] only uses the information of network transmissions to estimate the current location. The only thing it relies on is the delivery rate when sending songs from servers to smart devices. A limitation of this work is that it can only be used on fixed moving routes. However, in HSR scenarios, this is not a problem because trains always run on the same routes.

VIII. Conclusion

In this paper, we design a series of measurement experiments on HSR scenarios to explore the relationship between the network latency and the bullet train's velocity. Our measurement results show us that network latency does have some relation with current velocity by the features of the bullet train's running. Based on this result, we design a new model that can be used to estimate current velocity only based on a measurement of network latency in a short time window. Our model uses Fourier transform to get useful information and use the recurrent neural network to get the final estimation. Our evaluations show that the selection of parameters does have some impact on the final result, but this model can gain up to at least 94% accuracy on all three routes.

Acknowledgement

This work was supported by the Canada Natural Sciences and Engineering Research Council (NSERC) Discovery Grant. Fangxin's work was supported in part by the the Key Area R&D Program of Guangdong Province with grant No. 2018B030338001. Ke's work was supported by the China National Funds for Distinguished Young Scientists with No. 61825204 and Beijing Outstanding Young Scientist Program with No. BJJWZYJH01201910003011.

References

[1] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen, "Smart Devices Are Different: Assessing and MitigatingMobile Sensing Heterogeneities for Activity Recognition," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 127–140.

[2] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[3] B. Han, F. Qian, S. Hao, and L. Ji, "An Anatomy of Mobile Web Performance over Multipath TCP," in *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '15. New York, NY, USA: Association for Computing Machinery, 2015.

[4] Y.-s. Lim, Y.-C. Chen, E. M. Nahum, D. Towsley, R. J. Gibbens, and E. Cecchet, "Design, Implementation, and Evaluation of Energy-Aware Multi-Path TCP," in *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT 2015. New York, NY, USA: Association for Computing Machinery, 2015.

[5] A. Croitoru, D. Niculescu, and C. Raiciu, "Towards Wifi Mobility without Fast Handover," in *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*. Oakland, CA: USENIX Association, May 2015, pp. 219–234.

[6] L. Li, K. Xu, T. Li, K. Zheng, C. Peng, D. Wang, X. Wang, M. Shen, and R. Mijumbi, "A Measurement Study on Multi-Path TCP with Multiple Cellular Carriers on High Speed Rails," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, ser. SIGCOMM 2018. New York, NY, USA: Association for Computing Machinery, 2018, pp. 161–175.

[7] J. Wang, Y. Zheng, Y. Ni, C. Xu, F. Qian, W. Li, W. Jiang, Y. Cheng, Z. Cheng, Y. Li, and et al., "An Active-Passive Measurement Study of TCP Performance over LTE on High-Speed Rails," in *The 25th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom 2019. New York, NY, USA: Association for Computing Machinery, 2019.

[8] J. Calle-Sánchez, M. Molina-García, J. I. Alonso, and A. Fernández-Durán, "Long term evolution in high speed railway environments: Feasibility and challenges," *Bell Labs Technical Journal*, vol. 18, no. 2, pp. 237–253, Sep. 2013.

[9] Q. Liu, K. Xu, H. Wang, M. Shen, L. Li, and Q. Xiao, "Measurement, Modeling, and Analysis of TCP in High-Speed Mobility Scenarios," in *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, June 2016, pp. 629–638.

[10] L. Li, K. Xu, D. Wang, C. Peng, K. Zheng, R. Mijumbi, and Q. Xiao, "A Longitudinal Measurement Study of TCP Performance and Behavior in 3G/4G Networks Over High Speed Rails," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2195–2208, Aug 2017.

[11] L. Li, K. Xu, D. Wang, C. Peng, Q. Xiao, and R. Mijumbi, "A measurement study on TCP behaviors in HSPA+ networks on high-speed rails," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, April 2015, pp. 2731–2739.

[12] Y. Ni and C. Xu, "A Multipath Transport Multihoming Mobile Relay Architecture for High-Speed Rails Networking," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys 2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 515.

[13] E. Dong, M. Xu, X. Fu, and Y. Cao, "LAMPS: A Loss Aware Scheduler for Multipath TCP over Highly Lossy Networks," in *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*, Oct 2017, pp. 1–9.

[14] E. Dong, M. Xu, X. Fu, and Y. Cao, "A loss aware MPTCP scheduler for highly lossy networks," *Computer Networks*, vol. 157, pp. 146–158, 2019.

[15] S. Ha, I. Rhee, and L. Xu, "CUBIC: A New TCP-Friendly High-Speed TCP Variant," *SIGOPS Oper. Syst. Rev.*, vol. 42, no. 5, pp. 64–74, Jul. 2008.

[16] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, I. Swett, J. Iyengar, V. Vasiliev, P. Jha, Y. Seung, K. Yang *et al.*, "BBR Congestion Control Work at Google: IETF 101 Update," in *Proc. ICCRG at IETF 102th Meeting*, 2018.

[17] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-Based Congestion Control," *Commun. ACM*, vol. 60, no. 2, p. 58–66, Jan. 2017. [Online]. Available: https://doi.org/10.1145/3009824

[18] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[19] L. Li, K. Xu, D. Wang, C. Peng, K. Zheng, H. Wang, R. Mijumbi, and Xiangxiang Wang, "A measurement study on Skype voice and video calls in LTE networks on high speed rails," in *2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS)*, June 2017, pp. 1–10.

[20] R. Khalili, N. Gast, M. Popovic, and J. Le Boudec, "MPTCP Is Not Pareto-Optimal: Performance Issues and a Possible Solution," *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1651–1665, Oct 2013.

[21] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke recognition using WiFi signals," *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, vol. 2015-Septe, pp. 90–102, 2015.

[22] J. a. G. P. Rodrigues and A. Aguiar, "Extracting 3D Maps from Crowdsourced GNSS Skyview Data," in *The 25th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '19. New York, NY, USA: Association for Computing Machinery, 2019.

[23] L. Sun, S. Sen, D. Koutsonikolas, and K.-H. Kim, "Widraw: Enabling hands-free drawing in the air on commodity wifi devices," in *Proceedings of the 21st Annual International Conference on Mobile Computing*

and Networking, ser. MobiCom '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 77–89.

[24] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of wifi signal based human activity recognition," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 65–76.

[25] K. Sung, J. Biswas, E. Learned-Miller, B. N. Levine, and M. Liberatore, "Server-Side Traffic Analysis Reveals Mobile Location Information over the Internet," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1407–1418, June 2019.