

# A Deep Learning-Based Approach for Train Arrival Time Prediction

**Citation for published version (APA):**

Buijse, B. J., Reshadat, V., & Enzing, O. (2021). A Deep Learning-Based Approach for Train Arrival Time Prediction. In D. Camacho, P. Tino, R. Allmendinger, H. Yin, A. J. Tallón-Ballesteros, K. Tang, S-B. Cho, P. Novais, & S. Nascimento (Eds.), *Intelligent Data Engineering and Automated Learning - 22nd International Conference, IDEAL 2021, Proceedings: 22nd International Conference, IDEAL 2021, Manchester, UK, November 25–27, 2021, Proceedings* (pp. 213-222). (Lecture Notes in Computer Science (LNCS); Vol. 13113), (Information Systems and Applications, incl. Internet/Web, and HCI (LNISA); Vol. 13113). Springer. [https://doi.org/10.1007/978-3-030-91608-4\\_22](https://doi.org/10.1007/978-3-030-91608-4_22)

**DOI:**

[10.1007/978-3-030-91608-4\\_22](https://doi.org/10.1007/978-3-030-91608-4_22)

**Document status and date:**

Published: 01/01/2021

**Document Version:**

Accepted manuscript including changes made at the peer-review stage

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.



# A Deep Learning-Based Approach for Train Arrival Time Prediction

Bas Jacob Buijse<sup>1</sup>, Vahideh Reshadat<sup>2(✉)</sup>, and Oscar Willem Enzing<sup>3</sup>

<sup>1</sup> Department of Cognitive Science and Artificial Intelligence or Data, Jheronimus Academy of Data Science, 5211 DA's-Hertogenbosch, The Netherlands

`bas.buijse@walnutdata.nl`

<sup>2</sup> Department of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology, P.O. Box 513, 5600 Eindhoven, MB, The Netherlands

`v.reshadat@tue.nl`

<sup>3</sup> ProRail, 3511 Utrecht, EP, The Netherlands

`oscar.enzing@prorail.nl`

**Abstract.** Level crossings have a function to let the traffic cross the railroad from one side to the other. In the Netherlands, 2300 level crossings are spread out over the country, playing a significant role in daily traffic. Currently, there isn't an accurate estimation of the arrival time of trains at level crossings while it plays an important role in traffic flow management in intelligent transport systems. This paper presents a state-of-the-art deep learning model for predicting the arrival time of trains at level crossings using spatial and temporal aspects, external attributes, and multi-task learning. The spatial and temporal aspects incorporate geographical and historical travel data and the attributes provide specific information about a train route. Using multi-task learning all the information is combined and an arrival time prediction is made both for the entire route as for sub-parts of that route. Experimental results show that on average, the error is only 281 s with an average trip time of one hour. The model is able to accurately predict the arrival time at level crossings for various time steps in advance. The source code is available at <https://github.com/basbuijse/train-arrival-time-estimator>.

**Keywords:** Train arrival time prediction · Deep learning · Spatial-temporal neural networks · Multi-task learning

## 1 Introduction

Every metropolis with an extensive transportation network commonly experiences problems with traffic flow management. These problems may lead to the delay of public transport, poor emergency services, increased fuel consumption, environmental pollution, etc. [2]. Providing accurate and timely traffic information such as arriving time of vehicles plays an important role in intelligent transport systems. In this term, level crossings also play a big role as they can hinder traffic for a specific time period when a train passes.

© Springer Nature Switzerland AG 2021

H. Yin et al. (Eds.): IDEAL 2021, LNCS 13113, pp. 213–222, 2021.

[https://doi.org/10.1007/978-3-030-91608-4\\_22](https://doi.org/10.1007/978-3-030-91608-4_22)

A railroad is divided into sections separated by electrical separation welds used to track whether a particular train is on the railroad. Whenever a train crosses an electrical separation weld, an electrical circuit is closed, and the electrical signal is converted to a digital signal. The signal is received, and the train position is updated by a train traffic controller in a control room. This helps the train traffic controller to guide the train traffic across the railroad safely. Being informed of the train cross passing time at level crossings in advance can have several advantages for multiple stakeholders. Those benefits are direct or indirect and include but are not limited to enhanced traffic flow, improved decision making for emergency services, and additional safety enhancements for (non-secured) level crossings.

Forecasting methods have large influence on the development of different artificial intelligent branches consists of Fuzzy Systems [10], Natural Language Processing [12–15], Expert Systems [19] etc. Numerous methods for predicting the arrival times of vehicles have been proposed in recent years. These methods support a variety of different approaches from traditional machine learning and statistical based models (e.g. support vector machines [7] and Kalman filter [1, 6]) to neural network based architectures (e.g. long short term memory (LSTM) [3, 11, 17, 20]). Predicting the arrival times with the help of deep learning models has been done in some recent works [9, 22]. However, most of these works aim to predict the arrival time for road vehicles [5, 16], and the railroad industry lagged behind that in terms of development. This paper presents a deep neural network-based architecture for predicting the train arrival time at level crossings. It is inspired by the Deep Travel Time Estimator model of Wang et al. [18]. The main contributions of this paper are as follows:

- In this paper, a state-of-the-art deep arrival time estimator is presented that predicts the arrival time of trains at level crossings. The model uses the spatial-temporal features, external attributes, and a special Convolutional Neural Network (CNN) layer called Geo-Conv layer.
- To the best of our knowledge, there is no previous study in the domain of arrival time predictions for trains that have utilized a deep learning model as an arrival time estimator to predict the arrival time of trains at level crossings.
- The experiments show that the proposed model can accurately predict the arrival time for various timesteps in advance.

The rest of this paper is organized as follows: In Sect. 2, an overview is given of the datasets that are used. Section 3 presents our proposed model in detail; Sect. 4 is dedicated to the details of the experiments and results; and finally, Sect. 5 concludes the paper.

## 2 Datasets

The dataset for training and testing of the proposed model is described in more detail in this section. In Listing 1.1 the required input format of the model is shown.

**Listing 1.1.** The data in JSON-format

```

{
  ....
}
{
  "Sections":["AH$2001T", ..., "HDR$1119BT"],
  "time_gap":[0.0, ..., 8507.0],
  "dist_gap":[0.0, ..., 185.56992499610186],
  "lats":[51.98704004887645, ..., 52.94988504678495],
  "lngs":[5.873883222270905, ..., 4.762389563964237],
  "TrainID":3066,
  "timeID":1142,
  "weekID":5,
  "dateID":1,
  "time":8507.0,
  "dist":185.56992499610186,
  "trainTypeID":3
}
{
  ....
}

```

In the listing, the *Sections* key is a list with all the sections that have been traversed for a particular train route. The *time\_gap* and *dist\_gap* keys respectively hold a list with the cumulative time and distance since the start of the route. Every new value in these lists correspond to the entry of a new section. The *timeID* key/value pair represents the start time of a trip in minutes from 12:00 PM and thus is a number between 0 and 1440 ( $1440/60 = 24$  h). The *weekID* is a number between 0 (Monday) and 6 (Sunday) that represents the day in the week. The *dateID* is the day of the month and last of all the *time* and *dist* keys hold the values for the total time and total distance traveled. Last of all, the *trainTypeID* key holds the type of the train and the *TrainID* key holds the unique identifier for that train.

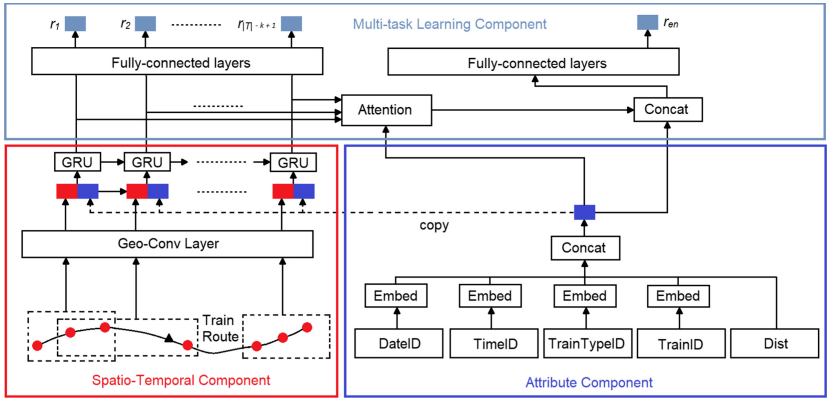
In order to predict the arrival time for trains at level crossings, an additional dataset is used that contains the coordinates of all the level crossings in the Netherlands. The coordinates of a level crossings often interfere with the coordinates of the end of a section. Thus, whenever the coordinates of a level crossing are known the travel time prediction can simply be made up until that point.

### 3 Proposed Model

We propose a new model for predicting the arrival time of trains at level crossings using a deep neural network architect. The model uses the spatial-temporal features, external attributes, and a special CNN layer namely, the Geo-Conv layer. These three main components are described in more detail in this section. An overview of the model is shown in Fig. 1.

**The attribute component** of the model processes the external factors and includes the start time of the trip (TimeID), day of the week (DateID), the train type (TrainTypeID), and a specific train identifier (TrainID). An embedding method is used to transform the categorical attributes into a lower dimensional vector. This means that each value of a categorical attribute  $v \in [V]$  is mapped to an embedding vector  $\mathbb{R}^{Ex1}$ . The influence of the above four attributes on the arrival time have been investigated while training the model.

Besides, the travel distance (Dist) is also incorporated as an attribute. The output of this component is a concatenation of all the attributes.



**Fig. 1.** The architecture of the proposed model

**The spatio-temporal component** in the model consists of two parts. The first part consists of a so-called GEO-convolutional network (GEO-Conv) that converts the coordinates of the sections into feature maps. The second part consists of gated recurrent units (GRUs) that learn the temporal aspects from the feature maps created by the GEO-convolutional neural network. GRUs use less training parameters and therefore use less memory. Moreover, in comparison with LSTMs they are executed faster and trained faster; so they are computationally efficient [21].

The GEO-Conv is similar to a standard CNN. Since a CNN usually needs a grid of equal partitioned cells in order to convolve over this grid, a modified GEO-Conv is applied. For instance, a CNN is applied to an image of  $N \times M$  pixels. This is in contrast to GPS points that could take every position on a map and thus are not structured at all. In order to grasp this fine granularity, the GEO-Conv is introduced. This layer uses a convolutional filter with kernel size  $k$  and a 1D-window that is applied over the sequence to generate a convoluted location sequence. The kernel size is a hyperparameter that can be adjusted by the user. A bigger kernel size means that more subsequent segments are pooled together into a single embedding. For example in Fig. 1 the kernel size is illustrated to be equal to three. This means three subsequent segments are

pooled together. Because it is difficult for the 1D-window to extract the distance directly from the raw coordinates, the final output of the GEO-convolution layer is concatenated with the distance of each segment.

**The multi-task learning component** is the final part of the model. This component combines all the previous output data and predicts a travel time for both the entire route and every segment or section.

To estimate the travel time for the segments, the spatio-temporal features of the  $i$ 'th subsequence of the route are used. Then, two fully connected layers are used to map every feature vector to a variable called  $r_i$  which represents the travel time for the  $i$ 'th segment. Figure 1 shows this fact. The prediction of the entire path is done based on the same feature sequence. For this, an attention pooling method is used that combines the spatial information of a subsequence with the external factors such as the start time of the route. Then, the attention vector is fed to a fully connected layer connected to the residual connections that enable the network to skip layers. This robust technique allows the training of very deep neural networks with multiple layers [8]. Finally, the network outputs predictions for the entire path as well as the segments. For the entire path, a single neuron is used to output the final prediction denoted as  $r_n$ .

## 4 Experiments

In this section, the performance of the proposed model is evaluated using the datasets described in Sect. 2. The dataset used for training consists of approximately 350.000 train routes that 70% is used as training data and 15% as validation- and test data. Applying grid search, the combination of hyperparameters that achieves the best performance is gained. The setting of the model hyperparameters used in the experiments is shown in Table 1. Cross-validation is not used in this research as the data is collected over time and thus it contradicts the fundamental assumptions of cross-validation that the data is independent and identically distributed [4].

**Table 1.** The best model hyperparameters after grid search

Hyperparameter	Description	Value
Batch size	The size of the batch to train	8
Epochs	The amount of training iterations	10
Kernel size	The kernel size of the Geo-Conv layer	2
Alpha	The weight of combination in multi-task learning	0.8

Table 2 provides the results of the best performing model. Both the error for the collective estimation (prediction for the entire routes) as well as the error for the individual estimation (prediction per section) are reported in Table 2. The individual estimations are used to provide the arrival time prediction for the

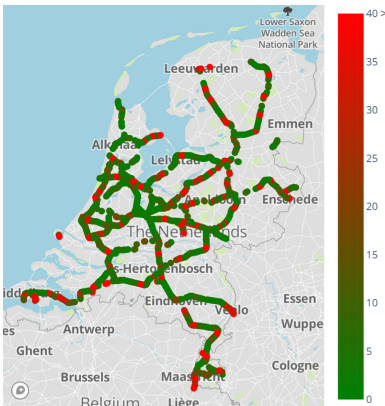
various timesteps before the level crossing. Considering the fact that the average trip time is approximately one hour, an MAE (Mean Absolute Error) of 281s for all the train trips in the test set reflects very accurate predictions.

**Table 2.** The results of the best model after grid search

Results best model after grid search		
	MAE (sec)	RMSE (sec)
Collective estimation	281.67	583.88
Individual estimation	7.18	34.48

Figure 2 and Fig. 3 give a visual insight on how the model performs on the test data. The MAE for the travel time estimation per sections is shown in Fig. 2. The smaller the MAE is, the greener a section.

Figure 3 shows the number of times that a section appears in the test data. The more a section appears in the dataset, the more its color turns to the blue spectrum. The color turns grey if a section only appears a few times in the dataset. Comparing Fig. 2 and Fig. 3, it can be concluded that fewer section data is available in the parts where the model has a high MAE. Due to the intrinsic of the model, this is expected. Therefore, the model most likely doesn't perform well for some parts in the Netherlands with less data available.



**Fig. 2.** The MAE per section (Color figure online)



**Fig. 3.** Number of routes traversing a section (Color figure online)

As an instance of how the model performs in the prediction of train arrival times at level crossing, two sample level crossings are investigated.

The first sample belongs to a level crossing at the Jachtlaan in the city of Apeldoorn of the Netherlands. Since it is situated close to the station of

Apeldoorn, trains are approaching this level crossing at different speeds (e.g., freight trains do not stop at the Apeldoorn station and cross the level crossing with a much higher speed than intercities and sprinters that stop at the station).

The second level crossing is a crossing at Jonkheer Ramweg near the Schalkwijk village. This level crossing is not positioned close to a station. However, because it is positioned in a more rural area, fewer train routes in the dataset crossed this level crossing. The arrival time prediction (Prediction Period) for both level crossings is made up to 20 min before the trains arrive at the level crossings. In this case, a farmer crossing with his herd or emergency services looking for the fastest route will have enough time to make a plan.

**Table 3.** Accuracy of the predictions for a level crossing in Apeldoorn

Predictions for the LC at the Jachtlaan, Apeldoorn			
Prediction Period	Accuracy of the prediction (in seconds)		
	98%	90%	Average
19–21 min	[−50 ~ −20]	[−45 ~ −25]	−35
9–11 min	[−7.6 ~ 7.8]	[−5.3 ~ 5.6]	0.1
0–1 min	[−2.9 ~ −1.9]	[−2.8 ~ −2.1]	−2.4

Table 3 shows the arrival time predictions for the level crossing at Jachtlaan. The table reports three prediction periods and the confidence intervals of 90% and 98%. The average prediction per period is displayed in the last column of the table. The results show that the prediction accuracy increases whenever the train is closer to the level crossing. This is mainly because a shorter route consists of less sections and therefore the accumulation of the error is less. Considering the 19–21 min prediction period, it can be seen that the 98% prediction interval is still only 50 s off. This means that even 20 min before the train arrives at the level crossing, the model is able to make an accurate prediction which is at most one minute off in 98% of the cases.

**Table 4.** Accuracy of the predictions for a LC in Schalkwijk

Predictions for the LC at the Jonkheer Ramweg, Schalkwijk			
Prediction Period	Accuracy of the prediction (in seconds)		
	98%	90%	Average
19–21 min	[−108 ~ −91]	[−106 ~ −94]	−100
9–11 min	[−27 ~ −20]	[−27 ~ −21]	−24
0–1 min	[0 ~ 0.9]	[0.2 ~ 0.8]	0.5



The predictions for the level crossing at the Jonkheer Ramweg in Schalkwijk can be seen in Table 4. It can be noted that the predictions are less accurate as compared to the predictions made for the level crossing at the Jachtlaan. This can be explained due to the fact that less data is available for the railroad traversing this level crossing. However, the error is still manageable.

## 5 Conclusion

The paper presents a deep learning-based architecture for predicting train arrival times at level crossings that consists of three main components, namely Spatio-Temporal, Multi-task learning, and Attribute. The Spatio-Temporal component consists of a Geo-Conv layer which extracts the spatial information from a train route and a GRU which extracts the temporal information. The Attribute component extracts information from the external attributes. Several useful attributes are embedded in the Attribute component, such as a unique train identifier (trainID) and the type of the train (trainTypeID). Using a multi-task learning component, prediction for the entire route and for the sections of that route is possible. This enables us to provide the most actual arrival time predictions possible at various points in time. Finally, the Multi-task learning component applies attention and predicts both the travel time per segment and the travel time for the entire route. The results of the experiments show that the model is capable of predicting the arrival time with great precision for various timesteps. This model can function as a basis for other applications built upon the predictions of this model in order to predict the arrival time for the trains accurately.

For future work, it would be interesting to investigate if more attributes could be included in the attribute component. Think for example of including the weather as a categorical variable in the attribute component. Moreover, it could be interesting to test whether using an LSTM instead of the GRU would obtain better results. Besides, it is interesting to test the model performance for different level crossings. Finally, the running time of a train arriving at a level crossing and leaving it again is not considered in this research and thus could be embedded into future work.

## References

1. Achar, A., Bharathi, D., Kumar, B.A., Vanajakshi, L.: Bus arrival time prediction: a spatial Kalman filter approach. *IEEE Trans. Intell. Transp. Syst.* **21**(3), 1298–1307 (2019)
2. Agafonov, A., Yumaganov, A.: Spatial-temporal  $K$  nearest neighbors model on mapreduce for traffic flow prediction. In: Yin, H., Camacho, D., Novais, P., Tallón-Ballesteros, A.J. (eds.) *IDEAL 2018*. LNCS, vol. 11314, pp. 253–260. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-03493-1\\_27](https://doi.org/10.1007/978-3-030-03493-1_27)

3. Agafonov, A., Yumaganov, A.: Bus arrival time prediction with LSTM neural network. In: Lu, H., Tang, H., Wang, Z. (eds.) ISNN 2019. LNCS, vol. 11554, pp. 11–18. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-22796-8\\_2](https://doi.org/10.1007/978-3-030-22796-8_2)
4. Bergmeir, C., Benítez, J.M.: On the use of cross-validation for time series predictor evaluation. *Inf. Sci.* **191**, 192–213 (2012)
5. Chen, C.H.: An arrival time prediction method for bus system. *IEEE Internet Things J.* **5**(5), 4231–4232 (2018)
6. Emami, A., Sarvi, M., Bagloee, S.A.: Using Kalman filter algorithm for short-term traffic flow prediction in a connected vehicle environment. *J. Modern Transp.* **27**(3), 222–232 (2019)
7. Hashi, A.O., Hashim, S.Z.M., Anwar, T., Ahmed, A.: A robust hybrid model based on Kalman-SVM for bus arrival time prediction. In: Saeed, F., Mohammed, F., Gazem, N. (eds.) IRICT 2019. AISC, vol. 1073, pp. 511–519. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-33582-3\\_48](https://doi.org/10.1007/978-3-030-33582-3_48)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
9. Jabamony, J., Shanmugavel, G.R., et al.: IoT based bus arrival time prediction using artificial neural network (ANN) for smart public transport system (SPTS). *Int. J. Intell. Eng. Syst.* **13**(1), 312–323 (2020)
10. Khetarpaul, S., Gupta, S.K., Malhotra, S., Subramaniam, L.V.: Bus arrival time prediction using a modified amalgamation of fuzzy clustering and neural network on spatio-temporal data. In: Sharaf, M.A., Cheema, M.A., Qi, J. (eds.) ADC 2015. LNCS, vol. 9093, pp. 142–154. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-19548-3\\_12](https://doi.org/10.1007/978-3-319-19548-3_12)
11. Petersen, N.C., Rodrigues, F., Pereira, F.C.: Multi-output bus travel time prediction with convolutional LSTM neural network. *Expert Syst. Appl.* **120**, 426–435 (2019)
12. Reshadat, V., Faili, H.: A new open information extraction system using sentence difficulty estimation. *Comput. Inf.* **38**(4), 986–1008 (2019)
13. Reshadat, V., Feizi-Derakhshi, M.R.: Studying of semantic similarity methods in ontology. *Res. J. Appl. Sci. Eng. Technol.* **4**(12), 1815–1821 (2012)
14. Reshadat, V., Hoorali, M., Faili, H.: A hybrid method for open information extraction based on shallow and deep linguistic analysis. *Interdiscip. Inf. Sci.* **22**(1), 87–100 (2016)
15. Reshadat, V., Hourali, M., Faili, H.: Confidence measure estimation for open information extraction. *Inf. Syst. Telecommun.* **1** (2018)
16. van der Spoel, S., Amrit, C., van Hilleegersberg, J.: Predictive analytics for truck arrival time estimation: a field study at a European distribution centre. *Int. J. Prod. Res.* **55**(17), 5062–5078 (2017)
17. Treethidataphat, W., Pattara-Atikom, W., Khaimook, S.: Bus arrival time prediction at any distance of bus route using deep neural network model. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp. 988–992. IEEE (2017)
18. Wang, D., Zhang, J., Cao, W., Li, J., Zheng, Y.: When will you arrive? estimating travel time based on deep neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
19. Wentworth, J.: Expert systems in transportation. Technical report, AAAI Technical Report WS-93-04 (1993)
20. Xu, H., Ying, J.: Bus arrival time prediction with real-time and historic data. *Cluster Comput.*, 1–8 (2017). <https://doi.org/10.1007/s10586-017-1006-1>

21. Yang, S., Yu, X., Zhou, Y.: Lstm and gru neural network performance comparison study: taking yelp review dataset as an example. In: 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI), pp. 98–101. IEEE (2020)
22. Zhang, X., Yan, M., Xie, B., Yang, H., Ma, H.: An automatic real-time bus schedule redesign method based on bus arrival time prediction. *Adv. Eng. Inform.* **48**, 101295 (2021)