Exploring Transferability of Real-time Crash Prediction Models using Transfer Learning under Imbalanced Data Setting

Cheuk Ki, Man[1], **Mohammed Quddus, Athanasios Theofilatos**
School of Architecture, Building and Civil Engineering, Loughborough University, Loughborough LE11 3TU United Kingdom, c.k.man@lboro.ac.uk
[b]*Centre for Transport Studies, Department of Civil and Environmental Engineering, Imperial College London, SW7 2AZ, United Kingdom, m.quddus@imperial.ac.uk*
School of Architecture, Building and Civil Engineering, Loughborough University, Loughborough LE11 3TU United Kingdom, a.i.theofilatos@lboro.ac.uk

Abstract

Real-time crash prediction is a heavily studied area given their potential applications in proactive traffic safety management. However, one of the fundamental issues relating to the application of these models is spatio-temporal transferability. The present paper attempts to address this gap of knowledge by combining Generative Adversarial Network (GAN) and transfer learning to examine the transferability of real-time crash prediction models under an extremely imbalanced data setting. Initially, a baseline model was developed using Deep Neural Network (DNN) with crash and microscopic traffic data collected from M1 Motorway in the UK in 2017. The dataset utilized in the baseline model is naturally imbalanced with 257 crash cases and 16,359,163 non-crash cases. To overcome data imbalance issue, Wasserstein GAN was utilized to generate synthetic crash data. Non-crash data were randomly under-sampled due to computational limitations. The calibrated model was then applied to predict traffic crashes for five other datasets obtained from M1 (2018), M4 (2017 & 2018 separately) and M6 Motorway (2017 & 2018 separately) by using transfer learning. Model transferability was compared with standalone models and direct transfer from the baseline model. The study revealed that direct transfer is not feasible. However, models become transferable temporally, spatially, and spatio-temporally if transfer learning is applied. The predictability of the transferred models outperformed existing studies by achieving high Area Under Curve (AUC) values ranging between 0.69 to 0.95. The best transferred model can predict nearly 95% crashes with only a 5% false alarm rate by tuning thresholds. Furthermore, the performances of transferred models are on par with or better than the standalone model. The findings of this study proves that transfer learning can improve model transferability under extremely imbalanced settings which helps traffic engineers in developing highly transferable models in future.

Keywords: Transferability, Transfer Learning, Imbalanced Dataset, Generative Adversarial Network, Oversampling

[1] * Corresponding author. Tel.: +4407522195148;
E-mail address: c.k.man@lboro.ac.uk

1. Introduction

Traffic crashes have significant implications to society with 1.35 million lives lost annually worldwide [1]. Current Intelligent Transport Systems (ITS) focused on a proactive safety paradigm which aims to anticipate crash occurrence in real-time to prevent crash from happening rather than reducing the impacts of crashes [2], [3]. Development of real-time crash prediction models made feasible with disaggregated traffic data available in real time [4]. Real-time crash prediction studies became popular amongst researchers with various statistical [5] and machine learning techniques [6], [7] were used to classify "crash case" from "non-crash case". Relatively satisfactory goodness-of-fit were attained with the use of full dataset classification and deep neural networks in recent studies data [8], [9]. However, transferability of real-time crash prediction models remained an area yet to be fully explored. Transferability in real-time crash prediction models is particularly important as a transferable model can save time and efforts in building new models and associated data collection for every setting [10]. Moreover, a transferable model would aid traffic managers to easily apply trained models to other settings with minimal adjustments required.

International literature indicates that past relevant studies used matched-case control sampling for transferability assessment, which might not be compatible with current Big Data driven approach in real-time crash prediction [11]–[13]. Limited transferability was recorded in the real-time crash prediction models. Such findings indicate that multiple models are needed to predict crashes at different segments in different periods. Moreover, using matched-case control is sensitive to sampling bias and thus not being able to capture the rare and stochastic nature of crash event. Models that are not generalisable to different temporal and spatial settings would prove to be costly in terms of data collection, pre-processing, training, evaluation and imposes practical challenges in assessing real-time traffic safety. Therefore, this study aimed to resolve the inadequacy of transferability assessment in current literature by employing full dataset classification whilst ensuring a good model transferability when apply the real-time crash prediction model in another temporal/spatial settings.

Implementation challenges arise in employing full dataset classification. Since crash is a rare event, the resulting dataset becomes highly 'imbalanced' with a multitude of non-crash cases for every crash case. Classifying an extremely imbalanced dataset might lead to undesirable high overall accuracy, but low sensitivity phenomenon (i.e., not correctly identifying crash cases), since a machine learning model might treat the minority class (also known as a class of interest) as noise [14].

To overcome the class imbalance issue, three common approaches can be applied. These include:

(i) resampling
(ii) cost-sensitive learning
(iii) ensemble methods

To improve temporal and spatial transferability of real-time crash prediction models, transfer learning is adopted in this study. Transfer learning is a machine learning approach commonly used in classification which the ML model is trained from the data in one domain and then applied to a different but a related domain [15]. This method is known to reduce training time and improve generalisability of the models [16].

Therefore, this study aims to contribute to current knowledge by assessing the temporal and spatial transferability of real-time crash prediction models, by applying transfer learning. In this paper, a Deep Neural Network (DNN) is trained with data collected from M1 Motorway, UK for 2017, by using a sample of 500,000 non-crash data as the baseline model. To the best of our knowledge, this study is the first to combine two state-of-art machine learning techniques, namely GAN and transfer learning, to boost real-time crash model predictability in addressing data imbalance and model transferability.

Given the dataset was still extremely imbalanced, Wasserstein Generative Adversarial Network (WGAN) was deployed to oversample crash cases. Subsequently, the model developed from M1 2017 is directly transferred to M1 2018 for temporal transferability, M4 and M6 2017 for spatial transferability and M4 and M6 2018 for spatio-temporal transferability. In the meantime, transfer learning is applied on these datasets by fitting the trained baseline model to the test datasets. Model prediction results are compared, and thresholds are also tuned for the transferred models for the best trade-off between sensitivity and specificity.

2. Methodology

2.1. Research Framework

In this study, there are three main steps to conduct the transferability assessment. They are: (i) oversampling crash cases through WGAN, (ii) developing DNN models and (iii) applying transfer learning on different datasets. To commence the study, datasets for all sites at different years (i.e., M1 2017, 2018; M4 2017; 2018, M6 2017 and 2018 are prepared. M1 2017 dataset was selected for developing a deep neural network model as the baseline model. Datasets for M1 were collected along Junction 1 to 30 with a total length of 239km. For M4 Motorway data were collected between Junction 2 to Junction 21 with a stretch of 220km. For M6 Motorway, it varies from 3 to 4 lanes with data collected from Junction 1 to Junction 32, distance between the segments is 220 km as well. Traffic parameters such as traffic flow (count), occupancy (in %), speed (kph), headway (deci-seconds) was collected for each lane every minute throughout the year. Data quality was assessed, and any erroneous data were deleted, for instance data containing missing values, negative values or conditions where speed > 0 whilst flow = 0 etc. Each crash is matched with the nearest upstream loop detector for the respective travelling direction using British Grid Coordinates between crash sites and loop detector locations. The nearest upstream loop detector is denoted as the crash segment (C). To capture pre-crash conditions, data for one upstream (U) and one downstream (D) segments were collected wiht minute-level traffic data aggregated in five-minutes intervals into six time slices 30 minutes before crash. Non-crash data can be defined as all traffic data where no crashes had occurred. Yet, traffic data period of 30 minutes after a crash are discarded to avoid the disruptive impacts of crash on normal traffic. From the data aggregation, various traffic parameters are calculated from flow, speed, occupancy and headway. Parameters calculated include the average of speed, occupancy and headway; standard deviations of flow, speed, occupancy and headway. Aggregated flow is also calcualted. For speed, the the coefficient of variation of speed and within lane, and between lanes standard deviation of speed were calculated. A total of 234 (i.e. 13 parameters × 3 segments × 6 time slices) variables were derived.

From the dataset, all data belongs to time slices 1 are not selected and the non-crash data are randomly undersampled to 500,000 entries due to computational limitations with HPC facility Lovelace provided by Loughborough. Data is hence standardised for better model convergence because models might be prone to erroneous predictions with different scales [7]. Next, the dataset is split into training and test sets in a 70/30 ratio. From the training set, WGAN is then initialized to oversample the crash data using Keras package from Python 3.6 [17]. Hyperparameters of WGAN are then tuned. The weights for the step which the classification accuracy between synthetic crashes and real crashes is closest to 0.5 using a separate classifier with the same hyperparameters as the critic network would be selected to generate synthetic crashes. The training dataset is then balanced with the synthetic data. The same procedure is also repeated for other dataset in preparation for transfer learning. With the training dataset for the baseline model balanced, Deep Neural Network (DNN) is therefore trained. Hyperparameters were empirically tuned to find the optimal model. With the best model tuned for based on AUC values, this model predictability is directly tested on datasets in different spatio-temporal settings. In the meantime, transfer learning is applied which weights for the best baseline model (i.e. M1 2017 model) is saved and fitted to other datasets (i.e. M1 2018, M4 2017, M4 2018, M6 2017 and M6 2018). The weights are fine-tuned empirically by customising the layers to be frozen from the source model during training. The model performance of the transferred model using transfer learning is compared with direct transfer under the imbalanced test set as well as the standalone models from each dataset. The threshold is tuned using a sensitivity-specificity curve from the best model. Experimental framework for the model transferability is displayed in **Figure 1**. The subsequent subsections would describe the methodology of WGAN, DNN and Transfer Learning.

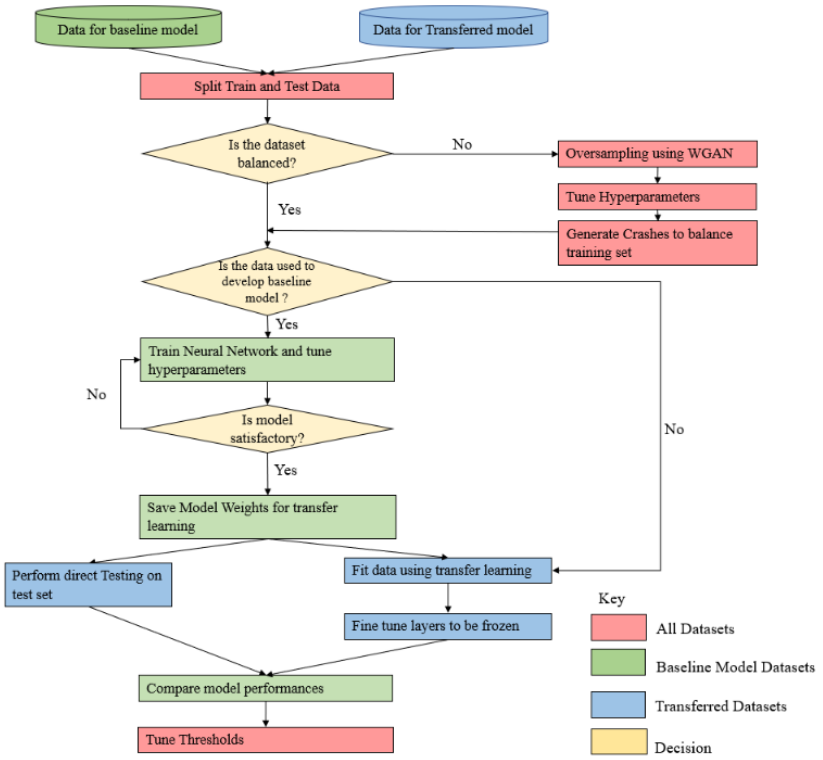


Figure 1: Schematic Diagram for the Experiment for Model Transferability

2.2.Wasserstein Generative Adversarial Network (WGAN)

WGAN is a variant of Generative Adversarial Network. Generative Adversarial Network consisted of two networks, Generative Network ($\boldsymbol{G}$) and Discriminative Network ($\boldsymbol{D}$). These two networks compete in a zero-sum game manner where the Generative Network ($\boldsymbol{G}$) keeps on generating synthetic data from the noise sample ($\boldsymbol{z}$). The Discriminative Network ($\boldsymbol{D}$) thus classifies whether the synthetic sample is real or fake against the original data ($\boldsymbol{u}$). A unique solution of Nash-equilibrium is achieved where the discriminator cannot distinguish whether the data is real or synthetic with the posterior probabilities as 0.5 in every data [18]. To formalise GAN in a minimax game setting, Equation 1 explained the objective function of GAN:

$$\boldsymbol{min_G max_D C}\,(\boldsymbol{G},\boldsymbol{D}) \;=\; \boldsymbol{E_{x\sim p_{data}}}[\log\,(\boldsymbol{D}(\boldsymbol{x})] + \boldsymbol{E_{z\sim p_z}}[(1-\log\left(\boldsymbol{D}\left(\boldsymbol{G}(\boldsymbol{z})\right)\right)] \tag{1}$$

However, Generative Adversarial Network has long been criticized for its instability in training [19]. Therefore, Wasserstein GAN was proposed [20] for a better training ability. The architecture of WGAN is similar to GAN but the Discriminator Network ($\boldsymbol{D}$) is replaced by a Critic Network ($\boldsymbol{C}$) with a linear activation instead of a sigmoid one. Another significant difference is that WGAN adopted the Wasserstein distance than Jensen-Shannon distance metric from GAN to train the Generator Network. Wasserstein distance is a better distance metric than JS distance as the latter is proven to be too strong to converge under low dimension manifolds. Comparatively, Wasserstein distance can converge and provide a differentiable model. Therefore, WGAN training is more stable than GAN.

To formalize the Wasserstein distance, Consider probability distributions of $\boldsymbol{P_G}$ and $\boldsymbol{P_r}$, Wasserstein metric measured the cost of transporting probability from one probability distribution to another rather than point distance [21]. Equation 2 denoted the Earth Mover distance (also known as Wasserstein-1):

$$\boldsymbol{W}(\boldsymbol{P_r},\boldsymbol{P_G}) = \inf_{\boldsymbol{\gamma}\in\boldsymbol{\Pi}(\boldsymbol{P_r},\boldsymbol{P_G})} \boldsymbol{E}_{(x,y)\sim\boldsymbol{\gamma}}[||\boldsymbol{x}-\boldsymbol{y}||] \tag{2}$$

From Equation 2, Wasserstein distance aimed to find the infimum plan to transport mass from $\boldsymbol{x}$ to $\boldsymbol{y}$ in order to transform probability distribution from $\boldsymbol{P_r}$ to $\boldsymbol{P_G}$, which is denoted by $\boldsymbol{\gamma}(\boldsymbol{x},\boldsymbol{y})$. $\boldsymbol{\Pi}(\boldsymbol{P_r},\boldsymbol{P_G})$ is the set of all joint distribution for $\boldsymbol{\gamma}(\boldsymbol{x},\boldsymbol{y})$ whose marginals are $\boldsymbol{P_r}$ and $\boldsymbol{P_G}$ respectively [20].

2.3. Deep Neural Network (DNNs)

To perform real-time crash prediction the training data is binary classification between crash and non-crash cases. The training data is defined in Equation 3.

$$(\mathbf{x_1}, \mathbf{y_1}), (\mathbf{x_2}, \mathbf{y_2}), \dots, (\mathbf{x_k}, \mathbf{y_k}), \qquad \mathbf{y_k} \in \{\mathbf{0}, \mathbf{1}\} \tag{3}$$

$\boldsymbol{y_k}$ refers to the class label with outcome as either 0 (non-crash) or 1 (crash) to the $\mathbf{x_k}$ as the matrix of the traffic parameters. DNNs is a neural network with more than 3 layers accumulated [22]. In a DNN, neurons receive input from the previous layer with some simple computation done in the current layer and passed to the next layer by activation functions [22]. The activation functions nonlinearly transformed the aggregated outcome from the hidden layer. Two activation functions (tanh and ReLu) were considered. Hyperbolic tangent (tanh) function is known for faster convergence with its symmetry at 0. Hyperbolic tangent (tanh) activation function is defined as $\mathbf{tanh}(\mathbf{x}) = \frac{\mathbf{2}}{\mathbf{1+e^{-2x}}} - \mathbf{1}$. Rectified linear unit (ReLu) is the other activation function considered. It is a half-wave rectifier and less computationally intensive. This function is defined as $\mathbf{ReLU}(\mathbf{x}) = \mathbf{max}(\mathbf{z}, \mathbf{0})$. At the output layer, a sigmoid activation function $\mathbf{y_{k_p}} = \frac{\mathbf{1}}{\mathbf{1+e^{-Z}}}$ is deployed so that the class is predicted with a range between 0 to 1.

2.4.Transfer Learning (TL)

Homogeneous transfer learning is suitable for current study given the feature spaces and labels amongst source and target dataset are the same. To formalize homogeneous transfer learning, it can be explained as follows:

Consider a domain $\boldsymbol{D}$, it is composed with feature space $\boldsymbol{X}$ and marginal probability distribution $\boldsymbol{P}(\boldsymbol{X})$, where $\boldsymbol{X} = \{\boldsymbol{x_1}, \dots, \boldsymbol{x_n}\} \in \boldsymbol{X}$. From which, $\boldsymbol{x_n}$ would be the nth feature vector in $\boldsymbol{X}$, it would be 195 variables in this study. For domain $\boldsymbol{D}$, the task $\boldsymbol{T}$ is the goal for the domain (i.e. predicting crash in real-time with data in M1 2017) given the label space $\boldsymbol{Y_i}$ with the predictive function $\boldsymbol{f}(\cdot)$ to learn the from feature vector and label pairs $\{\boldsymbol{x_i}, \boldsymbol{y_i}\}$ where $\boldsymbol{x_i} \in \boldsymbol{X}$ $\boldsymbol{and}$ $\boldsymbol{y_i} \in \boldsymbol{Y}$. Therefore, the source knowledge (M1 2017) can be defined as $\boldsymbol{DS} = \{(\boldsymbol{x_{S1}},) \dots, ,)\}$, $\boldsymbol{where}$ $\boldsymbol{x_{Si}} \in \boldsymbol{X_S}$ is the ith data instance of $\boldsymbol{D_S}$ and $\boldsymbol{y_{Si}} \in \boldsymbol{Y_S}$ is the corresponding class label for $\boldsymbol{x_{Si}}$. Similarly, for the target domain, it can be defined as $\boldsymbol{D_T} = \{(\boldsymbol{x_{T1}}, \boldsymbol{y_{T1}}), \dots, (, \boldsymbol{y_{Tn}})\}$ where $\boldsymbol{x_{Ti}} \in \boldsymbol{X_T}$ is the ith data instance of $\boldsymbol{D_T}$ and $\boldsymbol{y_{Ti}} \in \boldsymbol{Y_T}$ is the corresponding class label for $\boldsymbol{x_{Ti}}$. Correspondingly, the source domain would be $\boldsymbol{D_s}$ with task $\boldsymbol{T_s}$ and predictive function $\boldsymbol{f_s}$ and the target domain would be $\boldsymbol{D_T}$ with task $\boldsymbol{T_T}$ and predictive function $\boldsymbol{f_t}$. The aim of transfer learning is to improve $\boldsymbol{f_t}$ with the related information from $\boldsymbol{D_s}$ and $\boldsymbol{T_s}$. for homogeneous transfer, the condition where $\boldsymbol{X_s} = \boldsymbol{X_T}$ depicted the homogeneous transfer and $\boldsymbol{X_s} \neq \boldsymbol{X_T}$ would indicate a heterogeneous transfer [23].

When applying transfer learning, one of the common approaches is to fine tune the weights trained from the source model to the target model by freezing different amounts of layers for the level of details to be adopted from the source knowledge to the transferred model. The idea of freezing layers indicated that the frozen layer is excluded from backpropagation calculation. Therefore, the gradients from the frozen layer are not updated during the model fitting process whereas unfrozen layers are subjected to backpropagation and weights updating process [24]. Yet, this approach is not compatible in current study as the pre-trained models are not dedicated for crash prediction purpose. Figure 2 showed the transfer learning process for this study.

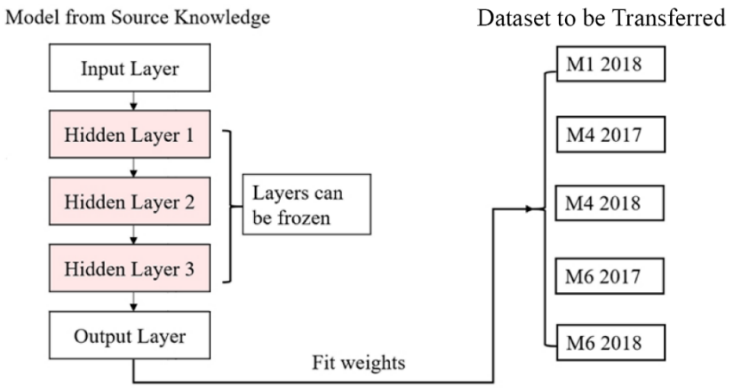


Figure 2: Transfer Learning Process

Referring to Figure 2, it clearly displayed that the hidden layers 1, 2 and 3 of the model from source knowledge (M1 2017) can be frozen during when fitted to the datasets of the target knowledge datasets. In this study, fine tuning on freezing hidden layer 1, l&2 and 1&2&3 would be tested.

2.5. Evaluation Criteria of the Classifiers and Transfer Learning Models

To tune and evaluate the performances real-time crash prediction models and transfer learning models, a confusion matrix is populated each time to display the classification performances of the test data according to a default threshold of 0.5 from sigmoid function. A confusion matrix provides four values:

True Positive (TP) – the number of crash cases correctly predicted as crash
False Positive (FP) – the number of crash cases incorrectly predicted as crash
True Negative (TN) – the number of non-crash cases correctly predicted as non-crash
False Negative (FN) – the number of non-crash cases incorrectly predicted as non-crash.

On the basis of these four values, the following metrics can be calculated: accuracy (rate of both crash and non-crash correctly predicted), sensitivity (rate of crash correctly predicted as crash) and specificity (rate of non-crash predicted as non-crash). In this paper, optimised precision is also used as metric to measure the probability of a correct crash predicted. Under an imbalanced test dataset, using precision metric is likely to produce skewed results because the prediction of one class dominates the test set. For example, if the non-crash data dominates the test set, the precision would be pushed to a low value even specificity of the classifier is high. Optimised precision accounts for the proportion of both classes in the test set as well as using 'Relationship Index' (RI) which measured the deviance between sensitivity and specificity. Equation 4 presented the calculation of Optimised Precision.

$$Optimised\ Precision = \mathrm{P} - \mathrm{RI} = SenN_p + SpeN_n - \frac{|Spe - Sen|}{Spe + Sen} \quad (4)$$

Where Sen is sensitivity, Spe is specificity, N_p proportion of the positive case in test set, N_n proportion of the negative case in test set.

To combine the false positives and true positives into a composite index with all thresholds considered, the metric considered would be the AUC value. This value is the area under Receiver Operating Characteristic (ROC) Curve by plotting the false positive and true positive rates at all thresholds. AUC value ranges between 0 to 1 and the value closer to 1 indicates better predictability of the classifiers. The best real-time crash prediction models and transferred models are determined based on the AUC values.

Regarding threshold tuning, Sensitivity-Specificity Curve is used to identify the best threshold. This curve plots the sensitivities and specificities across all thresholds using the probabilities from the testing set. The intersection points between the two curves indicate the threshold where the sensitivity equals to specificity. This method is deployed in similar crash prediction studies [25].

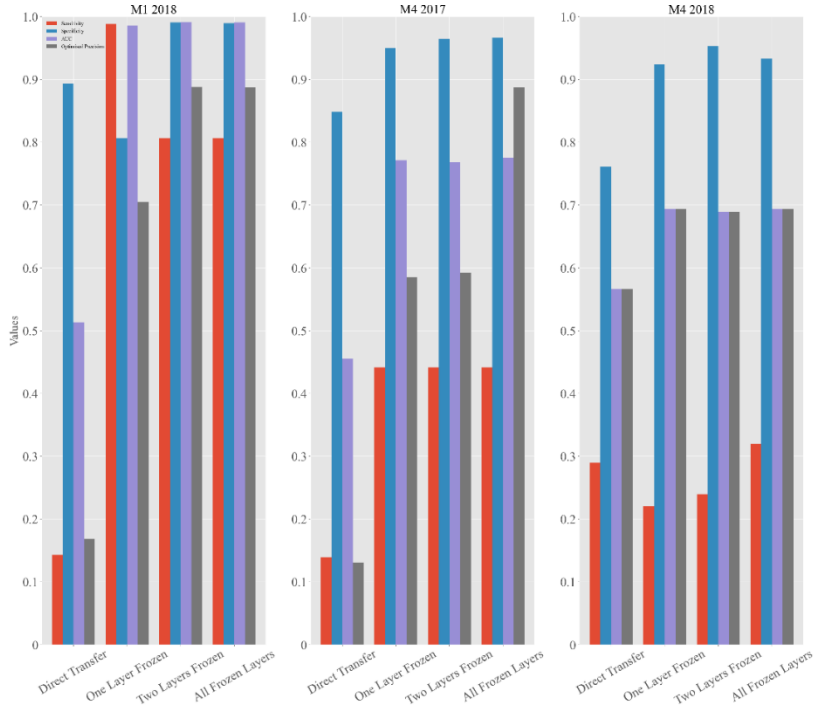
3. Analysis and Results

For all the dataset, given that crash cases are the minority case, they are all oversampled with WGAN. Even though when the baseline model was trained using balanced dataset, it is important to fit the dataset to be transferred under a balanced dataset. The reason of fitting a balanced transferred dataset is to avoid the weights of the trained model learning more from the non-crash case and hence overlooking the class of interest which is the crash case.

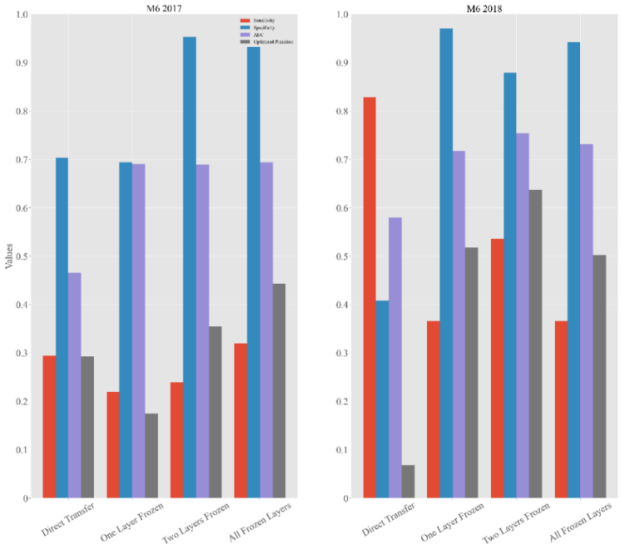
Only crash cases in the training set are selected for training WGAN. Using M1 2017 as an example, 180 out of 257 (70% of total) crash cases were selected in WGAN training. Regarding the hyperparameters of Wasserstein Generative and Discriminative networks, three layers of hidden layer is set. In each hidden layer the number of nodes is tested by the multiples of 64. To prevent overfitting, a dropout layer of 0.2 following each hidden layer is applied. Such setting is capable to strike a balance between extracting complex nonlinear information and avoid overfitting [26]. In each layer, the chosen activation function is rectified linear units (ReLu) to prevent model from suffering vanishing gradient problem [27] and the learning rate is fixed at 0.0002 with satisfactory results in other GAN training studies [28]. In terms of batch size, it was tested by multiples of 32 but the maximum batch size is subjected to the number of training data available. The optimal hyperparameters are selected via grid search with classification performances between real and synthetic data from the separated classifier were recorded every 100 steps of the training with total 50,000 steps trained. Optimal step was found when the classifier produces classification accuracy closest to 0.5 (Nash-equilibrium).

With the synthetic samples generated from WGAN, the baseline model for the source knowledge (M1 2017) needs to be firstly trained to enable transfer learning. The training of the model began with the dataset from M1 2017 split into training and testing sets. The training set is balanced with WGAN synthetic data whereas non-crash data is undersampled to 500,000 cases due to computational limitations. Afterwards, hyperparameters for the deep neural network are empirically tuned. Optimal hyperparameters were chosen for 3 hidden layers, 3000 nodes in each layer. Tanh is deployed as activation function with 0.2 set as dropout, regularization L2 parameter was set as

0.01 avoid overfitting [26] . Model is trained with 20 epochs at learning rate of 0.001 with a batch size of 20,000. The summary of model performance from direct testing from the baseline model and transferred models with layers being fine-tuned are presented at **Figure 3a to 3e** for 5 respective datasets. Standalone models of respective dataset to be transferred are developed as a benchmark to compare with the transferred results. Summary of comparison between standalone models (benchmark) and transferred results are shown in **Figure 4**.



(a) M1 2018 (b) M4 2017 (c) M4 2018



(d) M6 2017 (e) M6 2018

Figure 3: Transferability Results: (a) M1 2018; (b) M4 2017; (c) M4 2018; (d) M6 2017; (e) M6 2018

Figure 3a to 3e presents the results of the transferability tests. Comparing results between directly testing the data from other period or motorways with transfer learning, it is apparent that transfer learning achieves much better results than testing the data directly from the model developed using M1 2017 data. All models recorded at least 20% increase in terms of AUC values. The results reflected that direct transfer is not possible with AUCs of merely around 0.5, indicating the model have almost no ability to transfer. On the contrary, under transfer learning, models with AUCs over 0.7 showed promise to predict in other temporal or spatial settings, irrespective of layers to be fine-tuned during transfer. Similar to AUCs, the direct transfer also reflected a relatively low optimised precision when comparing with performances under transfer learning.

Referring to the case of temporal transferability, the sensitivity increased to over 0.9 and the accuracy and specificity increased by about 10%. Similar results were observed in spatial transferability test when fitting data to M4 2017, where the sensitivity improved from 0.14 to 0.44. Improvements were also recorded when the model was fitted on another location (M6), yet the sensitivity was lower than direct model testing. However, given the AUC from the transfer learning results were much higher, it indicated that threshold tuning would improve the sensitivity of the model by trading off accuracy and specificity.

In terms of spatio-temporal transferability, the model indicated that direct transfer is proven to be infeasible with unsatisfactory results for both datasets with AUC values lower than 0.6 and the optimised precision were as low as 0.06 for M6 2018. Under transfer learning, its model predictability improved to a more satisfactory level. Yet, the model predictability is still below par with a modest AUC for M4 2018 (0.69) and M6 2018 (0.73). The reason for attributing to the fluctuating results might be due to significant differences in traffic dynamics or road geometry in M4 and M6 2018 compared to M1 2017. Therefore, when the weights from the model trained on M1 2017 is fitted to the M4 and M6 2018 dataset, its model predictability may not be optimal.

Different layers of fine-tuning were tested in transfer learning as well. Between different fine-tuned layers, it is observed that the transferred results were slightly better when the model froze more than one hidden layer. Yet, the differences in AUCs are not significant. Between the standalone models and the transferred models, most of the transferred models have similar model performances with the standalone models. In addition, 4 Out of 5 transferred datasets slightly outperformed the standalone models (**Figure 4**).

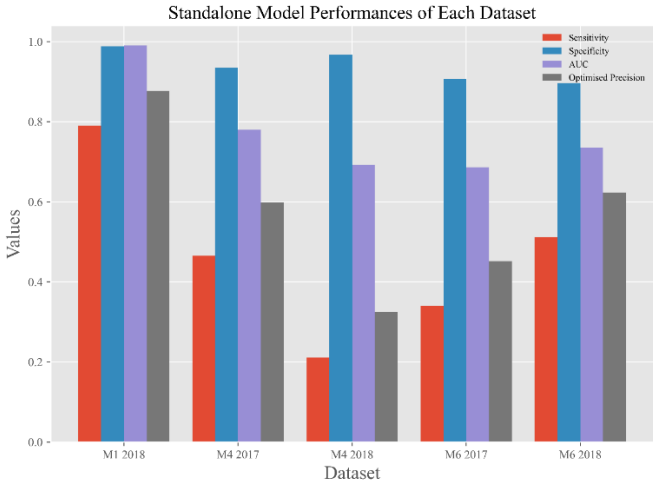


Figure 4: Standalone Model Performances for Each Transferred Dataset

Based on the results from the transfer learning, threshold tuning is applied to trade-off between sensitivity and specificity from the best transferred model in each dataset. From the sensitivity-specificity graphs, the thresholds were the intersection point between the sensitivity curve and the specificity curve at different thresholds between 0 and 1. The thresholds tuned were low. This is because most of the posterior probabilities were close to 0 under the imbalanced test set with the dominance of non-crash data. With the tuned thresholds, **Table 1** displayed their relative accuracy, sensitivity, and specificity for the transferred model for the test set.

Table 1: Model Performances for the Test Set after Threshold Tuning

Data	Transfer	Threshold	Sensitivity	Specificity	Optimised Precision
M1 2018	Temporal	0.12	0.95	0.95	0.95
M4 2017	Spatial	0.028	0.697	0.719	0.703
M4 2018	Temporal and Spatial	0.02	0.66	0.69	0.668
M6 2017	Spatial	0.065	0.64	0.653	0.642
M6 2018	Temporal and Spatial	0.11	0.682	0.681	0.680

4. Discussion

Transferability of real-time crash prediction models is an important area to study. A transferable model could save time and effort to build separate models through collecting data extensively. The applicability of transferring the

model temporally, spatially, or spatio-temporally is investigated in this study as studies focused on the model transferability has been limited. To the best of our knowledge, the present study is the first to use transfer learning to improve transferability of a real-time crash prediction model.

More specifically, a total of six different datasets were utilized with the data collected between 2017 to 2018 along M1 Junction 1 to 30, M4 Junction 2 to 21, and M6 Junction 1 to 32. Afterwards, Wasserstein Generative Adversarial Network is used to oversample crash cases from the naturally imbalanced dataset with a sample of 500,000 non-crash data. A three-layered feedforward deep neural network was developed to predict M1 2017 data as the source knowledge. Transferability assessments are conducted on 5 other datasets through: (i.) Direct testing from the baseline model, (ii.) Models fitted with transfer learning through fine-tuning layers and (iii.) standalone models of the datasets. Metrics such as AUC, accuracy, sensitivity, specificity and optimised precision were used to assess model performance. The last step was threshold tuning for the transferred models using sensitivity-specificity curve to balance the sensitivity and specificity. In general, the transferability tests using transfer learning have shown promising results. Compared to testing direct testing from the baseline model, the AUC value for transfer learning greatly improved by at least 0.20 on average (from around 0.50 to 0.70). Optimised precision was also increased by at least 0.10. Such result indicated that direct transfer is not viable, something that was also suggested by [29]. Another possible explanation could be that the study areas were much larger than other studies. Hence, the traffic dynamics and road geometry of the locations vary more than transferring a model to nearby segments like the study by [12]. Thus, it was reasonable to conclude that some degree of direct transfer could not be observed in our study.

Between the best transferred model and the standalone model, it was interesting to note that the performance of the best transferred model has negligible differences with the standalone model amongst all five datasets. More specifically, the AUCs of the best transferred model from 4 out of 5 datasets slightly outperformed the Standalone Model. The satisfactory performance from the best transferred model indicates that the baseline model for transfer learning is a good feature extractor and a weight initialization agent as the dataset to be transferred can fully benefit from the pre-trained model. High predictability has also been a proven benefit of transfer learning [15]. On the other hand, the standalone models might not learn the features of crashes and non-crashes fully given the number of samples for oversampling were limited. This promising result has not been observed in other studies comparing transferred models with standalone models. Yet, comparing the results of the original model and transferred model, the present study proved that transfer learning would not significantly deteriorate the model predictability, whereas about 10-20% reduction in predictability were recorded [13]. In addition, the sensitivity of the models under transfer learning outperformed other studies with 0.99 AUC, 0.80 sensitivity, 0.95 specificity and 0.89 optimised precision for temporal transferability. These results could be considered more promising than [11], [29] who applied both machine learning and statistical methods. Regarding spatial transferability, the best transferred model achieved an AUC of 0.77, sensitivity of 0.70 with 0.70 specificity for spatial transferability. The spatial transferability results for this study are slightly worse off compared to [30]. However, it is noted that the degree of spatial transfer for this study was much greater, whereas [30] the tested models were applied on different segments and travel directions within the same motorway. However, the results for both temporal and spatial transfer result in modest transferred results with AUCs at low 0.70.

Despite a few study limitations such as a lack of crash data in testing and the undersampling of non-crash data, this study demonstrated that model transferability vastly improved when transfer learning is deployed. More importantly, transfer learning can overcome the limitation of traditional DNN models which cannot be directly transferred when tested on another dataset. Future research can focus on testing the minimum amount of data to be required to ensure transferability. Similarly, testing on transferability could be conducted if neural network models are applied with a combination of different dataset trained and tested in different temporal and spatial settings.

5. Conclusions

Transferability has yet to be fully explored in real-time crash prediction studies. The present study confirmed that transfer learning can improve the transferability of a real-time crash prediction model in either temporal, spatial or spatio-temporal setting. This study adds to current knowledge and could provide a solid basis to further investigate model transferability. Additionally, the present paper is the first to carry out a transferability assessment with models developed closest to full dataset classification setting, rather than a matched case-control sampling. To the best of our knowledge, this study is also the first to combine two state-of-art machine learning techniques, namely GAN and transfer learning, in order to boost real-time crash model predictability in addressing data imbalance and model transferability. More importantly, this study suggested that transferability could be improved by using transfer learning under extremely imbalanced setting.

This study revealed that direct transfer is not feasible as models become transferable temporally, spatially and spatio-temporally under transfer learning. With respect to temporal transferability, the use of transfer learning improves the AUC value significantly (i.e., from 0.51 to 0.98). The enhancement is less significant in spatial transferability, with the AUC value increasing from 0.45 to 0.78 for M4 2017 dataset and from 0.58 to 0.75 for M6 2018 dataset. This finding shows that transfer learning is able to improve transferability of real-time crash prediction models under extremely imbalanced settings apart from saving time and efforts in building multiple models for different sections of a motorway by time period. As for the practical implications, our approach could assist traffic engineers in developing highly transferable models in future. Lastly, results from this study could also help traffic managers to predict crashes in other temporal and spatial settings if a trained model has already been in place. Yet, data for the new temporal or spatial setting is still needed to be collected.

Acknowledgment

No Sponsors Declared.

References

1. WHO, “Road traffic injuries,” World Health Organisation, 2018.
2. Abdel-Aty M., Shi Q., Pande A., and R. Yu, “Real-Time Traffic Safety and Operation,” in Safe Mobility: Challenges, Methodology and Solutions: Volume 11, D. Lord and S. Washington, Eds. Emerald Publishing Limited, 2018, pp. 175–204.
3. A. Pirdavani et al., “Application of a Rule-Based Approach in Real-Time Crash Risk Prediction Model Development Using Loop Detector Data,” Traffic Inj. Prev., 2015. 16: p.786–791.
4. Hassan H. and M. Abdel-Aty, “Predicting reduced visibility related crashes on freeways using real-time traffic flow data,” J. Safety Res., 2013. 45: p. 29–36.
5. Abdel-Aty M., Uddin N., Abdalla F., Pande A., and L. Hsia, “Predicting freeway crashes based on loop detector data using matched case–control logistic regression,” Transp. Res. Board, 2004. 1897: pp. 88–95, 2004.
6. R. Yu and M. Abdel-Aty, “Utilizing support vector machine in real-time crash risk evaluation,” Accid. Anal. Prev., 2013.
7. A. Theofilatos, C. Chen, and C. Antoniou, “Comparing Machine Learning and Deep Learning Methods for Real-Time Crash Prediction,” Transp. Res. Rec. J. Transp. Res. Board, 2019. 2673: (8) p. 169–178.
8. Yang K., Wang X., Quddus M., and R. Yu, “Deep Learning for Real-Time Crash Prediction on Urban Expressways,” 2018.
9. Cai Q., Abdel-Aty M., Yuan J., Lee J., and Y. Wu, “Real-time crash prediction on expressways using deep generative models,” Transp. Res. Part C Emerg. Technol., 2020. 117: p. 102697.
10. Hossain M., Abdel-Aty M., Quddus M., Muromachi Y., and S. N. Sadeek, “Real-time crash prediction models: State-of-the-art, design pathways and ubiquitous requirements,” Accid. Anal. Prev., 2019. 124: p. 66–84.
11. Xu C., Wang W., Liu P., Guo R., and Z. Li, “Using the Bayesian updating approach to improve the spatial and temporal transferability of real-time crash risk prediction models,” Transp. Res. Part C Emerg. Technol., vol. 38, pp. 167–176, Jan. 2014.
12. Shew C., Pande A., and C. Nuworsoo, “Transferability and robustness of real-time freeway crash risk assessment,” J. Safety Res., vol. 46, pp. 83–90, Sep. 2013.
13. Sun J. and J. Sun, “A dynamic Bayesian network model for real-time crash prediction using traffic speed conditions data,” Transp. Res. Part C Emerg. Technol., vol. 54, pp. 176–186, May 2015.
14. Beyan C. and R. Fisher, “Classifying imbalanced data sets using similarity based hierarchical decomposition,” Pattern Recognit., vol. 48, no. 5, pp. 1653–1672, May 2015.
15. Goodfellow I., Bengio Y., and A. Courville, Deep Learning. The MIT Press, 2016.
16. Guo Y., Shi H., Kumar A., Grauman K., Rosing T., and R. Feris, “SpotTune: Transfer Learning through Adaptive Fine-tuning,”. 2018.
17. F. Chollet, “Keras.” 2015.
18. I. J. Goodfellow et al., “Generative Adversarial Networks,” Jun. 2014.
19. Arjovsky M. and L. Bottou, “Towards Principled Methods for Training Generative Adversarial Networks,” Jan. 2017.
20. Arjovsky M., Chintala S., and L. Bottou, “Wasserstein GAN,” 2017.
21. Adler J. and S. Lunz, “Banach Wasserstein GAN,”. 2018.
22. LeCun Y., Bengio Y., and G. Hinton, “Deep learning,” Nature, 2015. 521: p. 436–444.
23. Pan S. and Q. Yang, “A Survey on Transfer Learning,” IEEE Trans. Knowl. Data Eng., 2010. 22: p. 1345–1359.
24. Xiao X., Mudiyanselage T., Ji C., Hu J., and Y. Pan, “Fast Deep Learning Training through Intelligently Freezing Layers,” in 2019 International Conference on Internet of Things and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing and IEEE Smart Data, 2019. p. 1225–1232.
25. Yuan J., Abdel-Aty M., Gong Y., and Q. Cai, “Real-Time Crash Risk Prediction using Long Short-Term Memory Recurrent Neural Network,” Transp. Res. Rec., 2019. 2673: p. 314–326.
26. Srivastava N., Hinton G., Krizhevsky A., Sutskever I., and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” J. Mach. Learn. Res., 2014. 15: p. 1929–1958.
27. Glorot X., Bordes A., and Bengio Y., “Deep Sparse Rectifier Neural Networks,” in Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011.15: p. 315–323.
28. Kingma D. and J. Ba, “Adam: A Method for Stochastic Optimization,” 2014.
29. Pande A., Das A., Abdel-Aty M., and H. Hassan, “Estimation of Real-Time Crash Risk: Are All Freeways Created Equal?,” Transp. Res. Rec. J. Transp. Res. Board, 2011. 2237: p. 60–66.
30. You J., Fang S., Zhang L., and X. She, “Real-Time Crash Risk Prediction Models and Transferability Analysis on Freeways,” J. Tongji Univ. (Natural Sci.), 2019. 47: p. 347–352.