Traffic Monitoring and Anomaly Detection based on Simulation of Luxembourg Road Network

Lin Zhu, Rajesh Krishnan, Aruna Sivakumar, Fangce Guo and John W. Polak

Abstract—Traffic incidents which commonly result from traffic accidents, anomalous construction events and inclement weather can cause a wide range of negative impacts on urban road networks. Developing a high efficiency and transferable traffic incident detection system plays an important role in solving the imbalance caused by traffic incidents between traffic demand and capacity. However, the existing literature on transferability of traffic incident detection is rather limited. The objective of this paper is to provide an accurate and transferable incident detection approach based on the relationship between traffic variables and observed traffic incidents, in particular at a network level. We propose a deep learning based method which has been calibrated using part of the collected traffic variables and the pre-assigned traffic incidents and then tested against the rest of the dataset. The proposed method is compared to other benchmarks commonly used in traffic incident detection, in terms of detection rate, false positive rate, f-measurement and detection time. The results indicate that the proposed method is significantly promising for traffic incident detection with high accuracy and transferability compared to the more widely used techniques in the literature.

I. INTRODUCTION

Traffic incidents, i.e., non-recurring events including accidents, road works and special activities, can disrupt the normal traffic flow and eventually result in severe delays or traffic network failures [1]. It has been widely accepted that effective traffic detection methods and implementation of proactive mitigation strategies can significantly contribute to the reduction of the impact of incidents [2]. Furthermore, as a key function of intelligent transport systems, reliable and prompt detection of traffic incidents is critical to enable traffic managers to proactively alleviate congestion under different levels of abnormal traffic conditions [3]. Anomaly detection in this paper refers to traffic incident detection.

Traditional traffic incident detection methods have been proved valid in many applications [4], [5]. Despite their effectiveness in detecting traffic incidents on any given arterial road, traditional detection methods still present certain challenges. Firstly, most methods have been developed and tested in the context of highways where the topology and traffic patterns are rather stable or simple compared to urban networks [3], [4]. Nevertheless, traffic detection on urban networks has been rarely discussed in the literature due to its high level of complexity arising from multi-agents such as pedestrians, different types of vehicles and the highly connected network topology [6]. Secondly, most studies that are focused on individual links have the limitation of low transferability [7]. For example, a traffic incident detection algorithm that is effective in triggering an accurate alarm on typical roads may only work well in the same or similar traffic patterns based on which its parameters were calibrated. Thus, it seems impossible for one detection method to perform well under a wide range of different traffic patterns on different types of road networks.

To address the aforementioned gaps, Convolutional Neural Networks (CNN) has been proposed to detect traffic incidents at the network level using flow and occupancy data from across the network. Three major contributions of this study include: (1) the CNN-based model is tested using data generated from a SUMO based microscopic traffic simulation environment and the validity of the proposed model is demonstrated; (2) the transferability of the proposed model is demonstrated by applying the method to efficiently detect traffic incidents at different locations on the network; and (3) the performance of the proposed method is successfully tested for detecting traffic incidents.

The rest of this paper is organised as follows. Section II reviews previous studies in traffic incident detection, deep learning techniques and their applications in transportation. The framework of the proposed deep-learning based method is presented in Section III. This is followed by the description of the experimental design with details of the simulation environment in Section IV. Section V evaluates the proposed methods followed by relevant discussion. Conclusions with a summary of findings and recommendations for future research are included in Section VI.

II. LITERATURE REVIEW

There is a vast body of literature devoted to modelling traffic incidents and automatic incident detection over the last few decades. Traffic incident detection was first developed in the early 1970s based on the occupancy measures on the upstream and downstream at fixed road sections [8]. This type of statistical algorithm detects significant differences between observed data and traffic characteristics predicted by prior probability or by identifying the outliers based on the principle of standard normal deviation [9]. On the other hand, another type of statistical technique is based on recognising an enormous pattern by using a fundamental flow-speed-occupancy diagram, such as the well-known McMaster algorithm [10]. However, the performance of the aforementioned algorithms lies in the accuracy of the thresholds chosen for identifying traffic incidents [9].

Given that traffic pattern changes caused by traffic incidents cannot evolve arbitrarily in space and time,
recently, a couple of data-driven models, particularly with
the state-of-the-art artificial intelligent techniques, have
been developed to detect traffic incidents. For instance,
[11] introduced a hybrid machine learning classifier that
combined neural networks and decision tree to analyse
traffic incidents. [12] applied convolutional neural networks
to detect traffic incidents by incorporating spatial correlations
captured by a connectivity matrix among neighbouring
ing edges in a simulated road network. [13] employed a
dictionary-based method to map twitter data into a
high dimensional binary vector and then identify the
traffic incidents with high-dimensional feature spaces. The
underlying principle of these data-driven methods is to
analyse versatile measures or data sources so as to recognise
the pattern shift as an evidence of a possible incident
occurrence. However, although a wide range of important
contributions are found in the literature, most studies have
been devoted to the context of the freeway while limited
attention has been paid to the context of urban networks
because of its complexity and difficulties in distinguishing
between usual congestion and traffic incidents.

Deep learning algorithms learn a high dimensional feature
via a sequence of non-linear transformations and these
transformations allow deep learning to efficiently deal with
sophisticated correlations between input and target output
[14]. The existing and newly emerging deep learning
algorithms have attracted increasing attention and have been
proved to be superior in some transportation studies [2]. For
instance, [15] proposed a Deep Neural Networks architecture
for traffic flow prediction; [14] employed a deep neural with
a sparse auto-regressive technique for variable selection in
traffic flow prediction. In an early study by [16], a Long
Short-Term Memory neural network was employed for traffic
speed prediction and the results suggested that the proposed
deep learning architecture could achieve the best prediction
performance in terms of both accuracy and stability.

Among all deep learning algorithms, CNN as a
special kind of deep feed-forward neural networks is
commonly applied to classify images. Its advantages in
supervised learning lie in the ability to set a variety
of multilayer perceptrons that are designed to require
minimal preprocessing because of their translation invariance
characteristics and shared-weights architecture [17]. Like
almost every other neural network it consists of three levels
of layers, i.e., an input layer, hidden layers and an output
layers. However, the major difference in the architecture is
that a CNN consists of multiple hidden layers which are
typically composed of convolutional layers, pooling layers,
fully connected layers and normalisation layers.

CNN has been extensively applied in the transportation
research domain especially for traffic prediction and
prediction problems because of its effectiveness in
classifying distinguished traffic states. For instance, [18]
indicated that CNN significantly outperformed the traditional
machine learning methods to identify the distribution of
transportation modes. The effective performance benefits
from the power of the CNN algorithm in extracting
higher-level features through multiple layers of non-linear
processing units. Therefore, the use of CNN for traffic
incident detection has been proposed in this paper. Given
that CNN models work well in spatially related tasks, they
are expected to deliver promising results in classifying traffic
incidents.

III. METHODOLOGY

A CNN-based incident detection algorithm was proposed
in this paper to detect anomalous conditions by comparing
traffic flow values with the historical traffic pattern.

A. Deep Learning Model

Artificial neurons are basically processing units that
usually find correlation between input variables \( X = (x_1, x_2, ..., x_n) \) and target output \( Y = (y_1, y_2, ..., y_n) \) through
an activation function \( f(\cdot) \) with weights \( W = (w_1, w_2, ..., w_n) \)
and bias \( b \). Mathematically, the full process of a neuron
operation can be defined as shown in Eq. 1:

\[
f(x) = \sum_{i=1}^{N} x_i \times w_i + b
\]  

where \( N \) is the size of input vector. Weights \( W \) and
input variables \( X \) have the same dimension \( \mathbb{R}^n \) while
the activation function is the transformation \( f(\cdot) : \mathbb{R} \rightarrow \mathbb{R} \). Multilayer networks are formed by grouping multiple
processing units aforementioned together. CNN is a typical
multilayer network and it typically consists of three type
of processing units, i.e., convolution, pooling and fully
connected layers.

In the convolution, a fixed-size window \( k \times k \) runs over
the input matrix \( x \) of size \( M \times M \) to define a region
of interest. And after that variable values inside the window are
created as the input with size of \( (M-k+1) \times (M-k+1) \) for
neurons with an operation like formula given in Eq.1, where
the feature is then extracted via an activation function \( f(\cdot) \).
Rectified Linear Unit (ReLU) formulated as Eq. 2 is used
here as an activation function due to its ability in detecting
high-frequency features in a local neighbourhood [19].

\[
f(x) = \max(0,x)
\]  

Typically, after the convolutional layer, there are pooling
layers to reduce the variance of features by running a
fixed-size window over the features to reduce the number
and optimise the gain of the features. The commonly used
operation is to select the maximum value over the feature
region generated by convolutional layers, as this max process
ensures that the significant features can be obtained for
detection even with varying levels of translations [19].

After several rounds of convolutional and pooling
operations, fully-connected layers make use of most
parameters to learn all neurons in the previous layer, and
output to the current layer where the spatial notion of the
matrix is reduced to that of a one-dimensional vector. To
prevent overfitting caused by parameters-dominated fully
connected layers, the dropout approach [20] which basically
drops a couple of neuron outputs randomly is employed. The advantages of this technique lie in the capabilities of decreasing the number of neurons, improving the speed of training and making the model practically effective. Finally, a classification layer is employed to calculate the class probability for each instance. The most common classifier is the softmax [21], as derived in Eq. 3, which gets the probability for each \(j\)th class given a sample vector \(X\).

\[
h_{W,b}(X) = P(y = j \mid X; W,b) = \frac{\exp X^T w_j}{\sum_{k} \exp X^T w_k} \tag{3}
\]

The objective is to minimise the loss function based on updated weights and bias, as expressed in Eq. 4.

\[
\arg\min_{W,b} J(W,b) = -\frac{1}{N} \sum_{i=1}^{N} (y_i \times \log h_{W,b}(x^{(i)}) \\
+ (1 - y_i) \times \log(1 - h_{W,b}(x^{(i)}))) \tag{4}
\]

where \(y\) represents a possible class, \(x\) is the input data, \(i\) is a specific input, and \(N\) represents the total number of datasets. With the cost function defined, the CNN can be trained in order to minimise the loss by using Stochastic Gradient Descent (SGD) to gradually update the weights and bias in search of the optimal solution, such that:

\[
W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha \frac{\partial J(W,b)}{\partial W_{ij}^{(l)}} \tag{5}
\]

\[
b_{ij}^{(l)} = b_{ij}^{(l)} - \alpha \frac{\partial J(W,b)}{\partial b_{ij}^{(l)}} \tag{6}
\]

where \(\alpha\) denotes the learning rates which represent the learning intensity in each step. The training step of a CNN generally consists of: (1) a feed-forward stage to pass all information from the first until the classifier; and (2) a back-propagation stage to calculate errors \(\delta\) and partial derivatives for weights and bias, which propagates these errors through all the layers from the classifier back to the first. The label of the supervised model can be defined as:

\[
f(x_t) = \begin{cases} 
1, & \text{if an incident occurs at time } t, \\
0, & \text{otherwise.} 
\end{cases} \tag{7}
\]

One example of a CNN model with a matrix input and an output of probability of traffic incident is shown in Fig. 1.

### B. Evaluation Metrics

The performance measures, Detection Rate (DR), False Positive Rate (FPR), F-measurement and Mean Time to Detection (MTTD), which are commonly used in automatic incident detection research [22] are selected in this paper to evaluate the proposed method in comparison with other conventional methods. Considering aggregated data interval, MTTD here mainly refers to computation time for each detection. In addition to the evaluation indexes aforementioned, a confusion matrix (see Fig. 2) and Receive Operating Characteristic (ROC) curve of the proposed CNN method is compared to those of conventional benchmarks to further evaluate the detection performance.

\[
FPR = \frac{FP}{FP+TN} \tag{8}
\]

\[
DR = \frac{TP}{TP+FP} \tag{9}
\]

\[
F\text{-Measurement} = \frac{2}{\frac{1}{DR} + \frac{1}{Precision}} \tag{10}
\]

\[
MTTD = \frac{1}{n} \sum_{i=1}^{n} (t_d - t_0) \tag{11}
\]

### IV. SIMULATION EXPERIMENTAL DESIGN

In order to test the model in a network level, a well-calibrated simulation model of the road network in the City of Luxembourg [23] has been used. Overall, the simulated vehicles made about 25 million trips during three months. Traffic flows are collected by inductive loop detectors at 5-minute intervals. Fig. 3 shows the topology of the simulation network.

#### A. Simulation Scenarios

To set up a realistic incident simulation, the traffic incidents are assigned with a random uniform sampling of network edges that are sources of traffic flow data detected. Two incident scenarios have been set up: (1) regular traffic

![Fig. 1: One example of a CNN model with a matrix input and an output of the probability of traffic incidents](image1)

![Fig. 2: Confusion matrix](image2)

![Fig. 3: Maps of the simulation case study: (a) the real-world location in Luxembourg; and (b) topology in the simulation network](image3)
conditions; and (2) incident scenarios with vehicle blocking upstream of detectors in the road with a variety of severity in terms of various locations and time of day. Two assumptions are made in this paper which generating incident scenarios. First, the duration of incidents conforms to a log logistic regression, and thus the percentage of incident duration of (30, 60, 90, 120) (in minutes) is (0.2, 0.4, 0.3, 0.1). Second, the number of lanes blocked sampled from the population (1, 2) with weights of (40%, 60%) for a three-lane road near a large junction. According to the level of severity, all traffic incidents are classified into four levels in addition to a baseline scenario without incidents (see Table I). The severity levels depend on two factors: i.e., the number of lanes blocked and the duration of incidents. The main parameters of these simulations are listed in Table II.

### Table I: Severity Levels and Incident Design Scenarios

<table>
<thead>
<tr>
<th>Level</th>
<th>Severity</th>
<th>Duration (min)</th>
<th>No. Lanes Blocked</th>
<th>Assumed Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Near Junctions</td>
<td>Level 1</td>
<td>Severe</td>
<td>60-120</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Level 2</td>
<td>Serious</td>
<td>0-60</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Level 3</td>
<td>Moderate</td>
<td>60-120</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Level 4</td>
<td>Light</td>
<td>0-60</td>
<td>1</td>
</tr>
<tr>
<td>Off Junctions</td>
<td>Level 1</td>
<td>Moderate</td>
<td>60-120</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Level 2</td>
<td>Light</td>
<td>0-60</td>
<td>1</td>
</tr>
</tbody>
</table>

1. A car brake is defined as a kind of traffic incidents by SUMO.
2. The incidents are randomly sampled in space and in time.

### Table II: Simulation Parameter Values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMO simulation duration (s)</td>
<td>86400</td>
</tr>
<tr>
<td>Simulation area (km²)</td>
<td>8</td>
</tr>
<tr>
<td>Total number of edges</td>
<td>635</td>
</tr>
<tr>
<td>Total length of edges (km)</td>
<td>89</td>
</tr>
<tr>
<td>Vehicles inserted</td>
<td>287034</td>
</tr>
<tr>
<td>Number of traffic stream composition</td>
<td>8</td>
</tr>
<tr>
<td>Number of traffic incidents</td>
<td>270</td>
</tr>
<tr>
<td>Number of loop detectors</td>
<td>219</td>
</tr>
<tr>
<td>Data resolution (s)</td>
<td>300</td>
</tr>
</tbody>
</table>

B. Simulation Validation

An example of simulated traffic flow under the normal traffic condition and abnormal condition caused by traffic incidents is shown in Fig. 4. The multi-lane road was sampled near junctions whereas the single-lane road was extracted far away from a large junction.

Given the traffic incidents during peak hours, it is noted that traffic incidents have a significant impact on traffic flows upstream and downstream whereas the impact on downstream is larger than the upstream. On the other hand, there is an immediate drop and surge before and after incidents on single-lane or off-junction roads because the incidents can block the whole single-lane roads while vehicles can traverse to other lanes in the same direction for the multi-lane roads near junctions.

V. RESULTS AND DISCUSSION

In this section of the paper, the detection of traffic incidents using SUMO simulated traffic data is compared with another methods based on conventional machine learning techniques for the same detection and sample of datasets.

### A. Representation of Model Inputs

Considering the warming up and cooling down period of simulation, as well as the balance of datasets, data from 7:00 to 19:55 are used in this study as the highly reliable data input for the proposed model and conventional machine learning models. The input for conventional machine learning methods are directly extracted from simulated data in a time-serious format. However, to learn the spatial correlation, a format of matrix is acquired as the input for the proposed CNN model. Thus, traffic flow data were converted into matrix-based input by making use of a method so called connectivity matrix.

Fig. 5 shows an example of a converted connectivity matrix with traffic flow values. Given the link flows on each edge and its connection with neighboured links in the network, the connectivity matrix could be generated. Specifically, the general process of connectivity matrix consists of three stages: (i) finding the direct connection between a pair of nodes; (ii) representing the direct connection with 1 whereas non-connection with blank; and (iii) filling the value of traffic flows by extracting data from detectors between the paired nodes.

### B. Benchmarks and Model Configuration

RF [24] and MLP were selected as two representative methods for comparison with the proposed incident detection algorithm in this study. For a fair comparison, only traffic flow data were utilised to calibrate all the models. These methods have capabilities of capturing the complex and non-linear relationship between different features [25], [26].

As for the input for machine learning methods aforementioned, the time series traffic flow data can be directly input into the models without the connectivity.
Fig. 5: Model input for the CNN method: (a) connectivity matrix; and (b) visualisation with traffic flows input

TABLE III: Model Configurations

<table>
<thead>
<tr>
<th>Model</th>
<th>Input Parameters</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>112 × 112 × 1, 14040</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Conv1 (32, 3 × 3, 1) + R + MaxPool</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Conv2 (32, 3 × 3, 1) + R + MaxPool</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Conv3 (64, 3 × 3, 1) + R + MaxPool</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fully Connected Layer 1 + Dropout</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fully Connected Layer 2 + Sigmoid</td>
<td></td>
</tr>
<tr>
<td>RF</td>
<td>3 × 14040</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10 decision tree with the depth of 3</td>
<td></td>
</tr>
<tr>
<td>MLP</td>
<td>3 × 14040</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hidden layers size of (10, 2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Regularisation parameter of 0.00001</td>
<td></td>
</tr>
</tbody>
</table>

1. 14040 is the sample size: 70% for training and 30% for testing.
2. Conv: convolutional layer; and R: ReLU [27], derived as Eq. 2.
3. MaxPool is with kernel size of (2 × 2).

Fig. 6: Confusion matrix of (a) CNN; (b) MLP; and (c) RF

TABLE IV: Performance of the Proposed Method and its Competitors

<table>
<thead>
<tr>
<th></th>
<th>DR</th>
<th>FPR</th>
<th>F-Measurement</th>
<th>MTTD</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>95.97%</td>
<td>0.24%</td>
<td>97.58%</td>
<td>4.91 (sec)</td>
</tr>
<tr>
<td>MLP</td>
<td>94.69%</td>
<td>0.55%</td>
<td>96.46%</td>
<td>5.60 (sec)</td>
</tr>
<tr>
<td>RF</td>
<td>93.59%</td>
<td>0.18%</td>
<td>96.61%</td>
<td>11.20 (sec)</td>
</tr>
</tbody>
</table>

Note: Network-wide computation time is estimated by multiplying the computation time of one link with the total number of links.

Another visual way to interpret the results of classification is via a confusion matrix for each method. Fig. 6 shows the confusion matrix with normalisation by the size of classes (i.e., number of elements in each class) in order to have a more visual interpretation of which class is being mis-classified in case of class imbalance. Specifically, the four values inside of each matrix represent precision, false positive rate, false negative rate and recall from the top left to right bottom respectively. Both cases caused by either wrongly indicating an incident or improperly identifying a non-incident instance, namely Type I Error and Type II Error in statistics, are presented in the confusion matrix clearly. It is obvious that the deep learning classifier has relatively small percentages in both cases and this proves its ability in dealing with the problem of class imbalance.

It is worthy to mention that models with the lowest false positive rate or false negative rate do not necessarily have the best detection rate or precision out of all possible models, but they are important indicators of whether a model works effectively by making use of all available information in the data.

Fig. 7 shows the ROC curve of the CNN classifier and two competitors. Generally, ROC curve is another common way to evaluate and intuitively visualise the performance of binary classifiers, which is generated by plotting the true positive rate versus the false positive rate for all thresholds ranging from 0 to 1. The understanding of ROC curve is quite straightforward with those of accurate classifiers in the upper left corner whereas those of poor classifiers close to the diagonal reference line which essentially represents a classifier that is not better than a random guessing. In this evaluation, the fact of all curves gathering in the upper left corner suggests that both the proposed CNN model and common machine learning alternatives perform well in the classification of traffic incidents, with low FPR and high true positive rate. This uniform high performance probably may result from the inherent limitation of simulation, i.e., low variation of traffic flow and simplified incident scenarios. The zoomed ROC curve shows that notwithstanding the generally good performance of all the methods, the CNN performed best, no matter what the threshold set. The high performance
of the proposed CNN model can also be confirmed from the metrics (see Table IV). The consistent superior performance of the proposed model for traffic incident detection based on both the Luxembourg network together with the Sioux-Falls network [12] demonstrated its potential transferability to other urban networks.

VI. CONCLUSION

The main contribution of this paper is a network-level incident detection method based on the CNN architecture that can be used for large urban road networks. To show the superiority of a CNN, its detection accuracy was compared with MLP and RF which have been broadly used in traffic incident detection fields. Through a series of experiments with different properties of traffic incidents using data from a well-calibrated simulation model of the City of Luxembourg, the superior accuracy and scalability of the proposed method was demonstrated. The CNN-based algorithm is promising for large-scale traffic incident detection and thus deserves further investigation. Future work includes conducting an empirical study with a more realistic experimental design with varying characteristics of traffic incidents or a real-world case study. Furthermore, exploring the relationship between the input matrix and the traffic state on links using underlying principles of traffic flow theory is a challenging and worthwhile endeavour.

REFERENCES