IET Intelligent Transport Systems

Special Section: Selected Papers from the 23rd ITS World Congress, Melbourne 2016

IET Journals
The Institution of Engineering and Technology

Automatic classification of traffic incident's severity using machine learning approaches

ISSN 1751-956X
Received on 21st February 2017
Revised 28th June 2017
Accepted on 12th July 2017
E-First on 11th August 2017
doi: 10.1049/iet-its.2017.0051
www.ietdl.org

Hoang Nguyen[1] ✉, *Chen Cai*[1], *Fang Chen*[1]

[1]*Data61 – CSIRO, 13 Garden Street, Eveleigh NSW 2015, Australia*

✉ *E-mail: hoang.nguyen@data61.csiro.au*

Abstract: During daily work at a Transport Management Centre (TMC), the operators have to record and process a large volume of traffic information especially incident records. Their tasks involve manual classification of the data and then decide appropriate operations to clear the incidents on time. A real-time automatic decision support system can minimise an operator's responded time and hence reduce congestion. Besides standard descriptions (e.g. incident location, date, time, lanes affected), severity is an important criteria that operators have to evaluate based on all available information before any control commands can be issued. The NSW TMC and the research organisation Data61 in Sydney have collaborated to discover and visualise frequent patterns in historical incident response records, leading to the automatic classification of severity levels among past incidents using advanced machine learning, active learning and outlier detection techniques. The experiments were executed using 4 years TMC's incident logs from 2011 to 2014 which includes >40,000 records. The classification model achieved nearly 90% accuracy in five-fold cross-validation and is expected to help the TMC to improve its procedures, response plans, and resource allocations.

1 Introduction

1.1 Context

Unplanned traffic incidents are detected using multiple sources including direct communications, CCTV cameras and road sensors. Transport operators rely on these heterogeneous information streams and complex computer interfaces to detect, confirm and clear incidents on the road network. Their tasks typically involve a large amount of manual work from incident classification, incident site clearance, traffic management, management of emergency services as well as information dissemination to authorities and the public [1]. With the support of the state-of-the-art machine learning (ML) approaches, the classification step can be automatically achieved by looking at the historical incident data. This paper will discuss the joint research project between Data61 and NSW Transport Management Centre (TMC) that aims to minimise the manual work for incident classification during incident management process, with the ultimate goal for reducing the impacts of incidents on the road network.

TMC monitors and manages the NSW state road network 24 h a day, 7 days a week, 365 days a year (Fig. 1). During morning and afternoon peak travel times, major events and unplanned incidents, the TMC monitors and coordinates Sydney's public transport operations across trains, buses ferries and light rail. The TMC works to deliver consistent travel times for both road and transport customers, on a daily basis. To do this, specially trained staff use advanced systems to monitor traffic and transport flows in real time, and respond to incidents as quickly as possible.

1.2 Motivation and challenges

Faster detection and processing of incidents brings a huge benefit to the community. TMC reported that on average, around 150 incidents per day occur during peak hours. Rough calculations show that every 1 min of delay on a typical urban road equates to approximately $90,000 economic impacts which equates to $23.4 million per year.

Dealing with a large amount of information, transport operators need to make decisions within a very short time-frame. Working under this pressure, they can become overloaded and fatigued over time, introducing the potential for increased time to assess situations and variability in decision making. Furthermore, it may take them longer to perform management tasks. These kinds of limitations in humans can increase incident clearance time. On the other hand, the existing computing facilities are very fast as they are able to process the Big Data and make decision in less than a second. ML models learn from several similar incidents recorded

Fig. 1 *The NSW TMC, in Sydney*

in historical data and then can be used to quickly generate suggestions for the most probable classes of incident.

Being able to detect and automatically classify the severity of an incident would allow TMC to build systems with automatic classifying capabilities. These systems can be further extended to provide decision support and automation. These features would allow operators to focus on higher level operational decision making and reduce their manual workload.

However, automatic incident data processing is challenging because of:

- *The complexity of road networks and massive volumes of data*: incident data on the entire traffic road network has been recorded every day and accumulated for many years.
- *Heterogeneous traffic patterns*: the traffic patterns on roads vary across days of a week and hours of a day. Different road segments often have distinct time-variant traffic patterns.
- *Data sparseness and distribution skewness*: even though a large number of sensors probing the traffic on roads are available, there are many roads that have only a small number of samples given the large size of road networks in a major city like Sydney. Moreover, a few road segments are travelled by thousands of vehicles in a few hours, while some segments may only be driven on several times in a day.

1.3 Contribution and impact

Our incident severity classification system has been developed to assess its usefulness in assisting traffic and incident management on a day-to-day basis as well as during special events. Incidents were classified by advanced ML algorithms before being presented to TMC operators in real time to support the incident management process.

Our system can also provide rich insights into the performance of previous incident operations by training from large and comprehensive data records. Standard operating procedures could benefit from historical records by emphasising strategies that proved useful, and improving on less successful ones. This type of classification system can then be incorporated in TMC operations systems to provide decision support and automation.

2 Related work

Classification is a popular task in ML where the categories of the instances are automatically assigned based on a train set of historical data. In TMC logs, incident severity is defined by the amount of influence an incident may have on normal traffic flow.

To the best of our knowledge, there was limited research on applying ML to incident severity classification from the textual logs. The closest work on crash severity classification based on road characteristics was proposed by Nowakowska [2]. The author used logistic regression in the form of the proportional odds ratio model and continuation ratio logits to identify the features that have a statistically significant influence on accident severity. In an earlier study, the pattern recognition method for road traffic accident severity in Korea was first introduced by Sohn and Shin [3]. Three data-mining techniques (neural network, logistic regression, and decision tree) were used to select a set of influential factors and to build up classification models for accident severity. This work was further developed by Sohn and Lee using the combination of data fusion, ensemble and clustering to improve the classification accuracy for the severity of road traffic accidents [4]. In another research conducted by Tesema *et al.*, the road traffic accident was classified using adaptive regression trees [5]. Miranda-Moreno *et al.* identified incident hot spots by incorporating accident severity and vehicle occupancy using a Bayesian accident risk analysis framework [6]. The overview of traffic accident analysis using ML paradigms was investigated by Chong *et al.* [7].

Instead of investigating the influence of incident on traffic flow, many research groups have developed the models for injury severity classification [8–11]. A popular research topic using incident logs is duration/clearance time prediction. The clearance time of urban traffic accident in Abu Dhabi was modelled using fully parametric hazard-based method with emphasis on the accelerated failure time metric [12]. Accident duration and its mitigation strategies were also studied on South Korean freeway systems [13]. The objective of this study is to analyse the critical factors that affect accident duration by means of an accelerated failure time metric model and to develop strategic plans and mitigation measures for reducing accident duration. In recent research, Dimitriou and Vlahogianni developed the rule-based fuzzy modelling approach for freeway accident duration estimation with the interaction of rainfall and traffic flow [14].

In summary, most previous researches in the literature were tested on data sets spanning only several months. They were also limited by a specific road type (e.g. highway) or area (e.g. urban). Our experiments were conducted based on a large data set with 4 years of records and it covered an entire state of Australia rather than just a selection of main roads or areas. The model trained on larger and more general datasets have a number of advantages for real-world incident classification. For example, a single model can classify incidents for the whole traffic network rather than training and testing different models on different road types (e.g. highway vs normal roads). Furthermore, using several years of data supports the model to learn additional repetitive patterns of incidents or congestions within specific areas and times, e.g. classify incidents related to road closures on Anzac day every year. As a consequence, our model is expected to be more powerful and practical in real-time incidents classification for NSW road network.

In terms of research methodology, the authors from previous works mainly applied available ML approaches (e.g. neural network, logistic regression, decision tree, adaptive regression trees, Bayesian network and ensemble methods) on their data without any consideration of choosing the best train set for improving model accuracy or reducing training and predicting speed. In our study, besides investigating popular ML methods for incident severity classification, two additional experiments on data selection are performed including outliers detection/removal for improving model's performance and three different active learning strategies for selecting the optimal train set (faster training and predicting).

3 Data descriptions and visualisation

3.1 Data description

Training data for this project includes official incident logs provided by the TMC. The data set covers 4 years periods from 2011 to 2014. Incidents of type accident, breakdown and hazard (A, B, H), generated by human operators (not automatic alarms) were investigated.

As the result, 15 features were extracted from the historical incident response database, as listed in Table 1. In total, 42,102 historical records were processed. As mentioned above, severity was used as a target label for supervised learning. In operations, when all the required information is collected for a specific incident, an operator will evaluate its severity based on a number of factors such as location (e.g. in motorway or in the city), lanes affected (e.g. one lane or three lanes blocked), time of the day (e.g. peak hour or normal time, day or night), day of the week (e.g. weekday or weekend), and whether injuries were reported. Operators also have the ability to assign a severity value ranging between one and three where the higher number represents a more serious incident in terms of affecting normal traffic and having a longer duration.

3.2 Data visualisation

Fig. 2 shows a heatmap of incident locations in NSW for the period of 4 years from 2011 to 2014. Heatmaps display colours on the map to represent the density of incidents within areas. As can be seen from this figure, there are two ‘hot’ areas where incidents happened very often:

Table 1 Incident response features used

Feature	Values
incident reporter	E.g. member of public, police
incident type	E.g. accident, breakdown
incident subtype	E.g. bus, car, motorcycle, closure
hour of day	hour from midnight
day of week	1–7
morning/afternoon peak	indicate whether an incident happened during peak hours
day/night	indicate whether an incident happened during day/night.
weekdays/weekend	indicate whether an incident happened during weekdays/weekend.
traffic direction	E.g. North, West
lanes affected	number of lanes affected
personal injury	whether any personal injury was reported
sector ID	12 possible areas managed by the TMC
road	road name and type
suburb	suburb name
incident severity	estimated by operator: 1–3

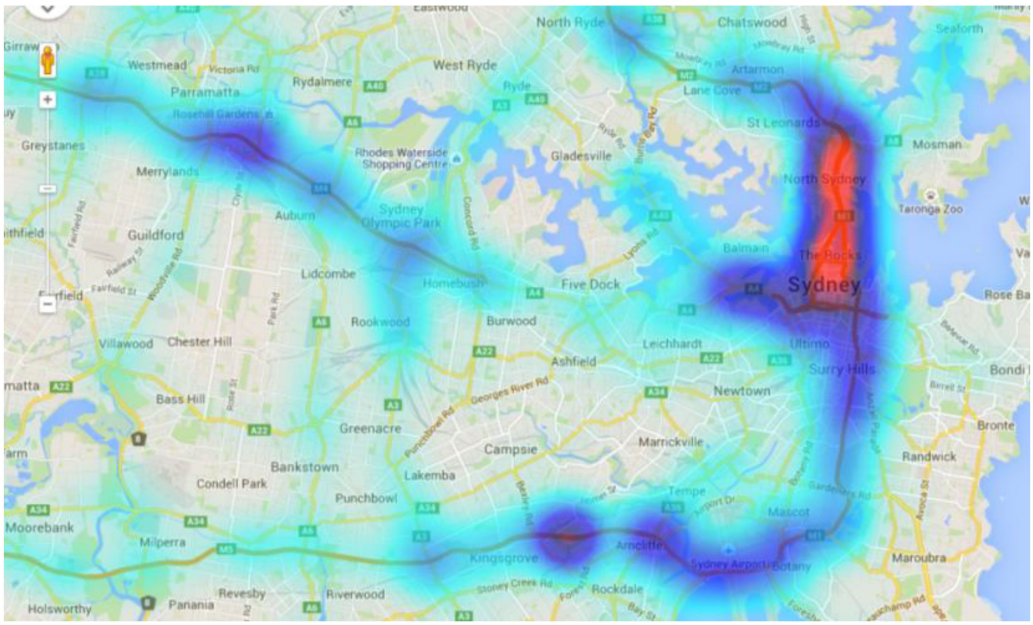

Fig. 2 *Heatmap of traffic incidents in NSW, Australia*

- Around the Sydney Harbour Bridge and tunnels including Sydney CBD.
- Along the motorways: M2, M4 and M5.

The timeline map (Fig. 3) provides the comprehensive view of all past incidents at any given day or time. This visualisation also supports to view the sequence of accidents, which helps to identify the potential propagation patterns of past incidents.

Fig. 4 shows the distribution of the incidents over time of a day and day of a week. The colours represent incidents' severity (1 is blue, 2 is red and 3 is cyan). As expected, most incidents happened during the morning and afternoon rush hours (7–9 a.m. and 4–6 p.m.) and it is noticeable that more incidents occur during weekdays. The distribution of severity is quite similar among the days and hours where the majority of them are at middle level (level 2).

The majority of incidents (around 62%) were related to breakdown, 23% were hazard and only 15% were marked as accident (Fig. 5). The incidents were further classified into subtypes where most of them related to car (C) (>50%), bus (B), truck (T) or road closure (Cl).

4 Generic ML process

ML approaches train models that represent the data in both a general and more accurate way. The basic advantage of supervised ML is that the incident characteristics and classification rules can be automatically learnt through training examples.

Given a set of instance-label pair (x_i, y_i), $i = 1,\ldots, t$, $x_i \in R^n$, $y_i \in Z^k$, the supervised ML system will learn a classifier F that maps instances to classes, $F : X_t \rightarrow Y_t$. In other words, x_i represents an input incident data and y_i represents the corresponding severity label.

In this paper, the experiments were executed using Weka, the popular open source ML library [8]. Since individual ML method has its own advantages and disadvantages, several ML algorithms which cover major types of classifiers have been investigated including

- *Generative*: Naïve Bayes [9].
- *Instance based*: *k*-nearest neighbour (k-NN) [10].
- *Discriminative*: Support vector machines (SVMs) [11] and decision tree (C4.5) [12].

4.1 Naïve Bayes

This classifier builds a predictive model by assuming conditional independence between features given class labels as presented above. For a conditional probability model, it assigns to vector $\boldsymbol{\phi}(x_j)$ probabilities

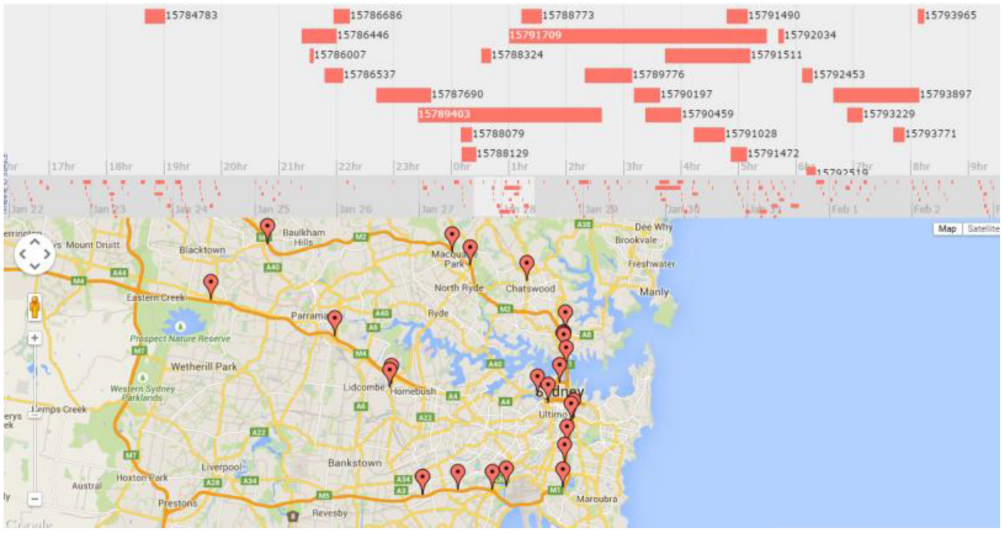

Fig. 3 *Timemap of traffic incidents in NSW, Australia*

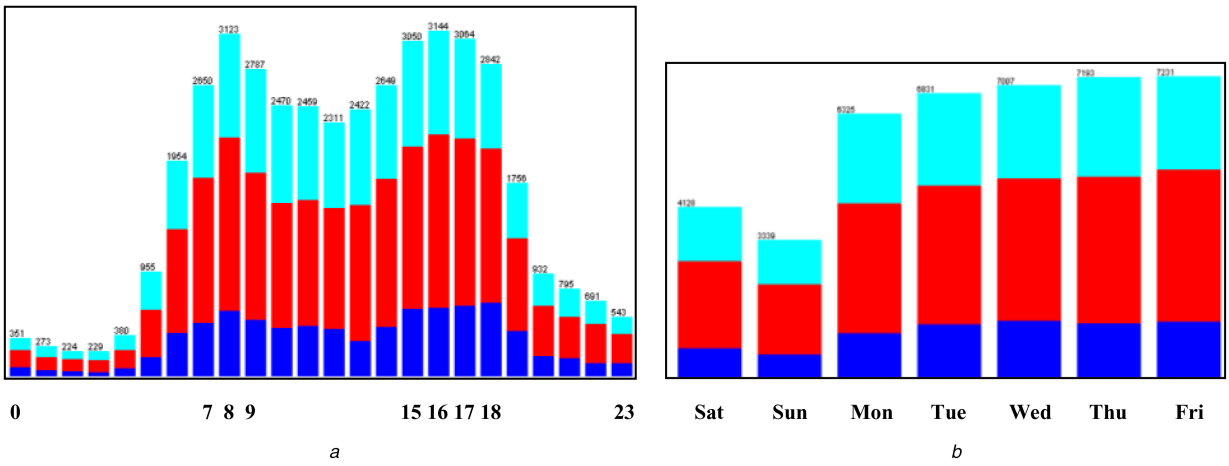

Fig. 4 *Distribution of the incidents*
(a) Time of incident within a day, ***(b)*** Day of incident within a week

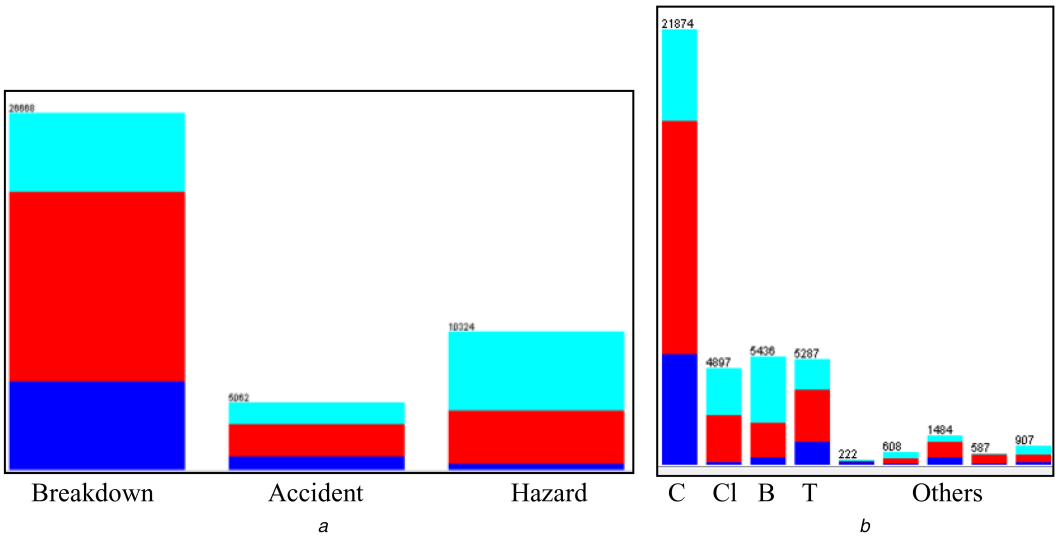

Fig. 5 *Majority of incidents*
(a) Incident types, ***(b)*** Incident subtypes

$$p(C_k \mid \varnothing^1(x_j), \ldots, \varnothing^m(x_j)) \tag{1}$$

for each possible outcomes or classes.

Assume each feature F_i is conditionally independent of every other feature F_j for $j \neq I$, given the category C. The joint model can be expressed as

$$\begin{aligned} p(C_k \mid \varnothing^1(x_j), \ldots, \varnothing^m(x_j)) &\propto p(C_k, \varnothing^1(x_j), \ldots, \varnothing^m(x_j)) \\ &\propto p(C_k)p(\varnothing^1(x_j) \mid C_k)p(\varnothing^2(x_j) \mid C_k)p(\varnothing^3(x_j) \mid C_k)\ldots \\ &\propto p(C_k)\prod_1^m p(\varnothing^i(x_j) \mid C_k). \end{aligned} \tag{2}$$

That means under the same independence assumptions, the conditional distribution over the class variable C is

$$p(C_k \mid \varnothing^1(x_j), \ldots, \varnothing^m(x_j)) = \frac{1}{Z}\, p(C_k) \prod_1^m p(\varnothing^i(x_j) \mid C_k) \quad (3)$$

where the evidence $Z = p(\phi(x_j))$ is a scaling factor dependent only on $\varnothing^1(x_j), \ldots, \varnothing^m(x_j)$, that is, a constant if the values of the feature variables are known.

To pick the hypothesis that is most probable, the common method is maximum a posteriori or MAP decision rule. Then the corresponding classifier is the function that assigns a class label $\hat{y} = C_k$ for some k as follows:

$$\hat{y} = \arg \max_{k \in \{1, \ldots, K\}} p(C_k) \prod_1^m p(\varnothing^i(x_j) \mid C_k) \quad (4)$$

The advantage of Naïve Bayes is it is reasonably simple in implementation based on statistics (counting). If the conditional independence assumption actually holds, it will converge quicker than discriminative models.

4.2 k-nearest neighbour

The label of the test data point is determined by the most similar instances in the past. The data vectors are projected into a multi-dimensional feature space and the distances between the vectors are computed. Euclidean distance is the most popular means for computing these distances:

$$\| \phi_K(x) - \phi_K(y) \| = \sqrt{K(x, x) + K(y, y) - 2K(x, y)} \quad (5)$$

The similarity between instances and their k-NN is calculated and the class is determined according to the most common incident among its k-NN (k is a positive number and generally small). This method is an instance-based learning that generates predictions using only local instances and all computation is deferred until classification.

There is no cost for learning process for k-NN method and it requires no assumptions about the data distribution or characteristics of the concepts to learn. It is also adaptable when more data is available. However, the k-NN model cannot be interpreted and it is computationally expensive when the data set is very large.

4.3 Support vector chines

In ML, SVM is one of the most popular classification methods. Traditional SVM is a non-probabilistic binary linear classifier where the training process is performed by constructing a multi-dimensional hyperplane that optimally separates data into different categories. In general, the optimal hyperplane is the one that has the largest distance to the nearest training points of any class (largest margin). Support vectors are the data points that lie closest to the optimal decision hyperplane. In the prediction process, new examples are mapped into the same space and their categories will be decided based on which side of the hyperplane they fall on. The SVM classifier solves the unconstrained optimisation problem with loss function $\xi(w; x_i, y_i)$:

$$\min_x \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi(w; x_i, y_i), \quad (6)$$

where $C > 0$ is a penalty parameter. For SVM, the two common loss functions are L1-SVM $= \max(1 - y_i w^{\mathrm{T}} x_i)$ and L2-SVM $= \max(1 - y_i w^{\mathrm{T}} x_i)^2$. In the testing phase, we predict a data point x as positive if $w^{\mathrm{T}} x > 0$, and negative otherwise. For multi-class categorisation like in this study, we combine several binary-classifiers using one-versus-rest method to accomplish the task. In this method, one constructs k classifiers, one for each class. The nth classifier constructs a hyperplane between class n and the $k - 1$ other classes.

SVMs have a strong theoretical support regarding overfitting, and with a well-chosen kernel they usually work well even the data is not linearly separable in the base feature space.

4.4 Decision tree (C4.5)

Decision tree describes the classification of training data by constructing the model of decisions with possible sequences in the form of a tree-like structure. In the decision tree, the leaves are the final category of a document and the branches are the sequences of the features that lead to the conclusion of a category. The tree representation is simple for human comprehension and its easy interpretation is the main advantage it has over other decision support methods. The decision trees can handle feature interactions and they are non-parametric. However, they are easily overfit and required to rebuild the trees when new data available.

C4.5 is a popular algorithm which has the capacity to generate a classifier in the form of a decision tree. The method was originally based on a divide and conquers approach with a gain criterion where the tree elements were built from smaller components and the poorly performed elements were discarded.

4.5 Evaluation metrics

Five-fold cross-validation was used as evaluation method in this project. Cross-validation is a technique for assessing how the results of a statistical analysis will generalise to an independent data set [13]. The labelled data is split into two parts, one called the training set and the other the test set. The model is built based on the training set only and it will be used to predict the output values for the data in the test set.

The primary evaluation metrics are precision (P), recall (R) and F1-score (F). Precision is the rate of retrieved instances that are correct while recall is the proportion of relevant instances that are retrieved. F1 score is the harmonic mean of precision and recall. They are calculated based on the true positive (TP), true negative, false positive (FP) and false negative:

$$P = \frac{\mathrm{TP}s}{\mathrm{TP}s + \mathrm{FP}s}, \quad R = \frac{\mathrm{TP}s}{\mathrm{TP}s + \mathrm{FN}s}, \quad F = \frac{2\mathrm{PR}}{P + R}$$

5 Classification results

In this section, the classification results for each of four ML methods are presented with the detailed per class performance along with confusion matrix. A confusion matrix contains information about actual and predicted classifications done by a classification system:

i. Naïve Bayes (Table 2)

i. Discussions (Table 3)

i. SVMs (Table 4)

i. Decision tree (C4.5) (Table 5)

6. Discussions

Among the four ML methods, decision tree (C4.5) has achieved the best performance with F1-score of 0.86. While decision tree's performance is usually low for a large-scale corpus with a large number of features, it is most suitable for the problems with a small number of attributes as in the incident records. Furthermore, the important feature that makes the greatest contribution to TMC is that a decision tree's result can be easily traced by travelling up the tree from the leaves.

Fig. 6 illustrates part of a decision tree derived from the incident data. In this figure, the detailed prediction path to conclude the highest severity for the most right branch is displayed. By inspecting the comprehensive prediction path, we can see

Table 2a *Continued*

Class	TP rate	FP rate	Precision	Recall	F1-score
1	0.766	0.060	0.753	0.766	0.759
2	0.818	0.160	0.830	0.818	0.824
3	0.812	0.093	0.803	0.812	0.807
weighted avg.	0.806	0.119	0.806	0.806	0.806

Table 2b Naïve Bayes performance on five-fold cross-validation and confusion matrix

Classified as	1	2	3
1	6206	1427	465
2	1524	16,792	2211
3	517	2010	10,902

Table 3a *Continued*

Class	TP rate	FP rate	Precision	Recall	F1-score
1	0.649	0.062	0.715	0.649	0.680
2	0.823	0.229	0.774	0.823	0.798
3	0.793	0.079	0.825	0.793	0.809
weighted avg.	0.780	0.149	0.779	0.780	0.779

Table 3b Three-NN performance on five-fold cross-validation and confusion matrix

Classified as	1	2	3
1	5253	2498	347
2	1737	16,885	1905
3	358	2426	10,645

Table 4a *Continued*

Class	TP rate	FP rate	Precision	Recall	F1-score
1	0.776	0.050	0.788	0.776	0.782
2	0.846	0.153	0.841	0.846	0.844
3	0.846	0.072	0.847	0.846	0.846
weighted avg.	0.833	0.107	0.833	0.833	0.833

Table 4b SVMs performance on five-fold cross-validation and confusion matrix

Classified as	1	2	3
1	6287	1454	357
2	1460	17,374	1693
3	232	1838	11,359

Table 5a *Continued*

Class	TP rate	FP rate	Precision	Recall	F1-score
1	0.837	0.059	0.772	0.837	0.803
2	0.861	0.115	0.877	0.861	0.869
3	0.869	0.050	0.890	0.869	0.879
weighted avg.	0.859	0.084	0.864	0.859	0.860

Table 5b C4.5 performance on five-fold cross-validation and confusion matrix

Classified as	1	2	3
1	6782	1021	295
2	1699	17,680	1148
3	309	1452	11,668

which data fields/values were used to make a decision and how many similar incidents were classified with the same class in the past. It is evident that this branch pattern has over 730 records of this type and has a confidence level of ∼85.5%. The confidence level is considered as the probability the classifier will have a correct prediction. Similar incidents with different class labels (14.5%) are also worth investigating for outlier detection.

As an example of application of this method in operations, the TMC could decide to set an acceptable confidence level threshold (e.g. 80%) for the system to automatically accept the computational severity value and operators only need to review the incidents

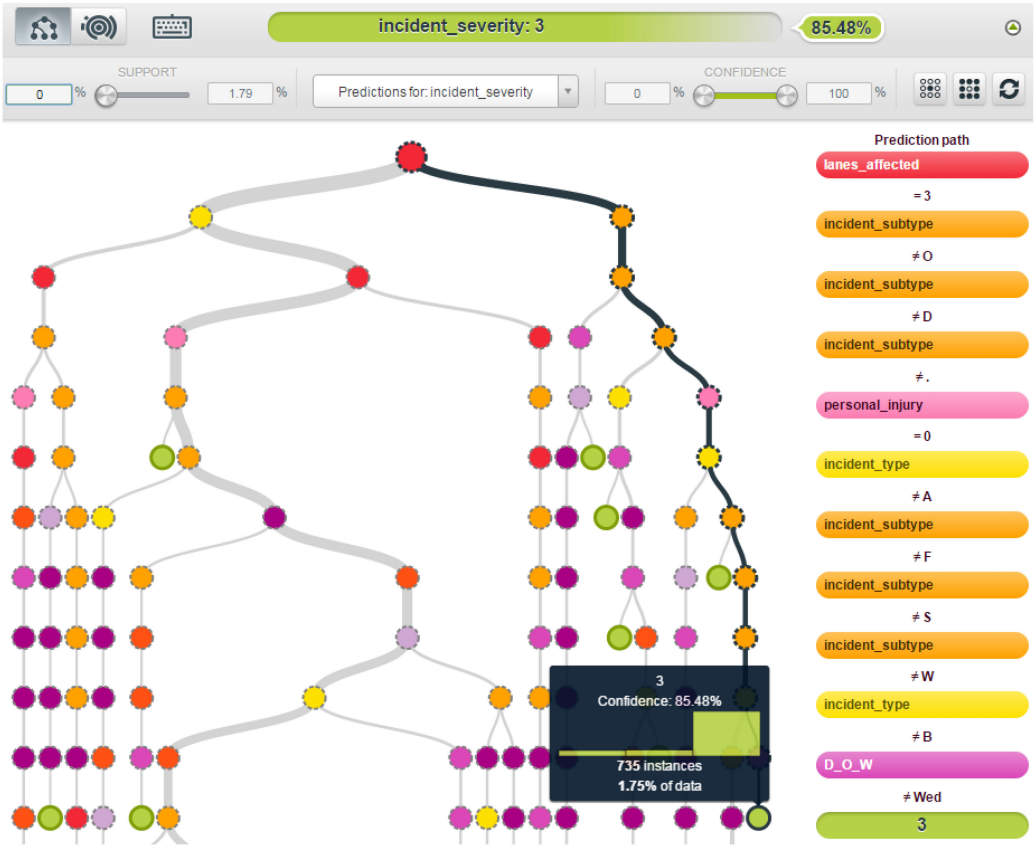

Fig. 6 *Part of a decision tree derived from the incident data with detailed prediction path for the most right branch example*

under this threshold. As a result, the work load on staff will be reduced significantly as they will not be required to assess the classification for every incident. The additional advantage of decision tree is that it can make a decision based on incomplete records, e.g. only five fields were required from the prediction path in Fig. 5. Hence, most classification tasks can be accomplished much sooner than when all the incident information is recorded.

From the C4.5 classification model and results, the following features have the most significant contribution to incident severity (listed in order of most important to least important): lanes_affected, incident_subtype, sector_id, hour_of_day and traffic_direction. Any lane closed or blocked in an incident will directly reduce the capacity on road hence it is ranked as the most important feature. Besides affected lane, the incident type, location (e.g. CBD or busy motorway) and time (e.g. peak hours) also impact traffic flow.

The three-NN algorithm has the lowest F1-score (~0.78) because many incidents may appear very similar but still have different severity levels due to a difference in time or location. The Naïve Bayes model result is slightly >0.80 while SVMs model correctly classified the incidents with a 0.833 F1-score. From the confusion matrices of all methods, the middle class (severity two) is the most difficult one to be separated from the two other classes.

7 Outliers detection and removal process

In this study, outlier (anomaly) is defined as the incident record which does not follow the normal pattern of the data distribution for a given class. In other words, if a low severity incident X is found very similar to many others past incident which were classified as high severity, there is a chance that X could be an outlier. As the outliers are not representative for the data distribution, they can add noise into the train set which may cause adverse effect to ML performance. As a consequence, outliers are usually detected and filtered out from train set before learning process [14]. It is also useful for transportation operators to study the outliers to identify the contribution factors that affected the classification outcome.

7.1 Outlier detection method

As we have studied several ML algorithms, the ensemble method can be utilised for outliers detection. The key idea and assumption behind this method is that the instances where all the proposed algorithms made the wrong predictions are worth to review [14].

The outlier detection and evaluation process is summarised as below:

- Execute five-fold cross-validation process for each of ML algorithms.
- For each algorithm, retrieve a set of all the instances with wrong prediction.
- Execute an intersection of the sets generated from Step 2, the resulted set contains the instances where all the ML algorithm failed to predict.
- Re-execute five-fold cross-validation with outliers removed from the TRAIN set only. This will made the test set consistent and comparable before

to evaluate the effect of outlier detection on unseen data.

7.2 Outliers removal results

Among over 40,000 incident records, there are 2028 incidents have been marked as potential outliers. Table 6 presents the ML performances before and after outlier removal process. The F1-scores for all algorithms have been increased by 1–4% by removing anomalies in the train set. The kNN ($k = 3$) method has experienced the lowest improvement of just 1% because it considered three most similar records hence lower the chance to pick up outlier as class label. As a result, removing outliers had a moderate effect on kNN method. Decision trees still have the highest F1-score of 89.4% with additional 3.4% improvement.

8 Active learning

Active learning (AL) allows the model to evaluate and choose the most informative records from the dataset to learn from instead of

Table 6 ML performances before and after outliers removal process

Algorithms	Before			After		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Naïve Bayes	0.806	0.806	0.806	0.848	0.848	0.847
kNN	0.779	0.780	0.779	0.790	0.789	0.789
SVMs	0.833	0.833	0.833	0.860	0.860	0.860
decision trees	0.864	0.859	0.860	0.895	0.894	0.894

making random selection. Given the same train size, the performance of active learners dominated random leaners in most cases [15]. Outlier detection methods are usually performed on the whole data set to remove 'bad' instances, AL strategies instead help to query the 'good' instances from the beginning to train the model. As soon as new records are available, these AL algorithms can also help to decide which records should be added to current train set and which records are redundant and could be ignored. When size and prediction time of the ML model are limited to strictly fit into the TMC's incident processing system, applying AL strategies is expected to bring higher model's performance than random sampling of train data.

There are two limitations in the available implementations of AL algorithms. Firstly, most AL packages only support binary classification and they are not applicable for multi-class datasets. Secondly, learning processes are performed based on single-query routine which requires updating and evaluating model after each selection. As a consequence, the single-query AL systems are considered as computational expensive on large datasets. To overcome these limitations, our contributions include extending the available AL algorithms to support multi-class datasets (using one vs the rest method) and batch learning (allowing model to make several queries in one AL round). In this section, three popular active learning approaches including uncertainty, self-confidence and balance exploration and exploitation are investigated for incident severity classification.

8.1 Uncertainty sampling

Uncertainty AL is based on the kernel machines and used uncertainty estimation as its selection strategy [16–18]. When data is converted to vectors and projected into higher dimensional space, the most uncertain record, which is considered as the most informative instance, is the one that lies closest to the decision hyperplane. For each record x with feature vector $\phi(x)$, the shortest distance from $\phi(x)$ to the hyperplane w_i is computed by $|w_i.\ \phi(x)|$. Therefore, uncertainty sampling uses the current model to select an unseen instance which is closest to the decision boundary.

8.2 Self-confidence

Self-confidence algorithm selects the record which is expected to come with minimal error probability when added to the train set [19]. As this error rate is unknown, the algorithm estimates future error rate using a log-loss function, e.g. using the entropy of the posterior class distribution on a sample of the unseen instances. Each record from the unseen data is examined by adding it to the train set and estimating the resulting future error rate as described in (7), then the instance with the smallest expected log is chosen. For each record x, from the unseen pool U, the algorithm trains a new model P' over $L'(x;\ y) = \{L' \cup (x, y)\}$ and the expected log-loss is defined as

$$E\left(\widehat{P'}_{L'}(x, y)\right) = \frac{1}{|U|} \sum_{y' \in Y, x' \in} \widehat{P'}(y' \mid x') \log \widehat{P'}(y' \mid x') \tag{7}$$

8.3 Balance exploration and exploitation (balance-EE)

Uncertainty-based AL algorithms choosing the examples near the boundary is considered efficient in 'exploitation'; however, it does not carry out 'exploration' by searching for potential other large areas in the data that it might wrongly predict [20]. Uncertainty method is suitable in case the initial model has the knowledge of all the important areas in the data distribution, e.g. the initial training set should contain instances from all areas. Osugi *et al.* proposed the balance-EE method that randomly determine whether exploration or exploitation will be used. After each exploitation step is chosen, the algorithm evaluates the efficiency of the exploration to adjust its probability p of exploring again [21].

Let h and h' be the hypothesis before and after the new example from exploitation is added. The change induced from h to h' $d(h, h') \in [-1; +1]$ is evaluated. If $d(h, h')$ is positive, the exploration was efficient and p will be kept high and vice versa.

For each of the hypotheses $h(.), h'(.)$, vectors of the predictions of h and h' are defined as $H = (h(x_1), h(x_2),\ ..., h(x_n))$ and $H' = (h'(x_1), h'(x_2),\ ..., h'(x_n))$. Then $d(h, h')$ is defined as

$$d(h, h') = 3 - 4\frac{(H, H')}{\| H \| \| H' \|} \tag{8}$$

The probability p for exploration will be updated as

$$p' = \max(\min(p\mu \exp(d(h, h')), 1 - \varepsilon), \varepsilon) \tag{9}$$

where ε defines the upper and lower bounds for the value of p, and μ is a learning rate for updating p.

8.4 Active learning results

This section compares the performances on incident data of random sampling and three active sampling algorithms presented above. The CMCS incident data was randomly divided into 50% train and 50% test (21,051 records each). Using decision tree C4.5 as base model, active learning was performed on the train set only and then predicted on the test set. Initial train data included 50 random instances and the queries were executed in batches of 10 in each learning trial, then the model was retrained. After 100 rounds of selection, the total number of selected examples was 1050 (20 times smaller than test set) for each sampling strategy. Fig. 7 shows the accuracy of the three AL algorithms and random sampling in classifying the incident severity.

All methods started from same train sets with F1-score of <66%, active learning performances are consistently higher than random sampling for all three algorithms. For the first 20 AL rounds, Uncertainty and self-confidence methods have comparable performances at nearly 75% which are 2% better than balance-EE and 7.5% better than random sampling. After 50 rounds, the uncertainty AL outperforms all other methods with a significant gap with random sampling ranged from 8 to 10%. At the end of the learning process, random sampling only reaches 71.5%, self-confidence and balance-EE have similar F1-scores of slightly over 78% while uncertainty's score exceeds 81%. The score of 81% for uncertainty sampling that was trained on only 1050 incident records is reasonably high (test on 50% of the unseen data with >20,000 records) when compared with the five-fold cross-validation performance (80% train and 20% test) presented in Section 5.

9 Conclusions

This research project carried out detailed analyses, visualisations and classification of the incidents reported from entire NSW state road network. It is capable of providing useful information to a TMC in a real-time manner. This can help the TMC to respond more quickly to incidents that could potentially affect the normal flow of traffic, and enable the TMC to make better management

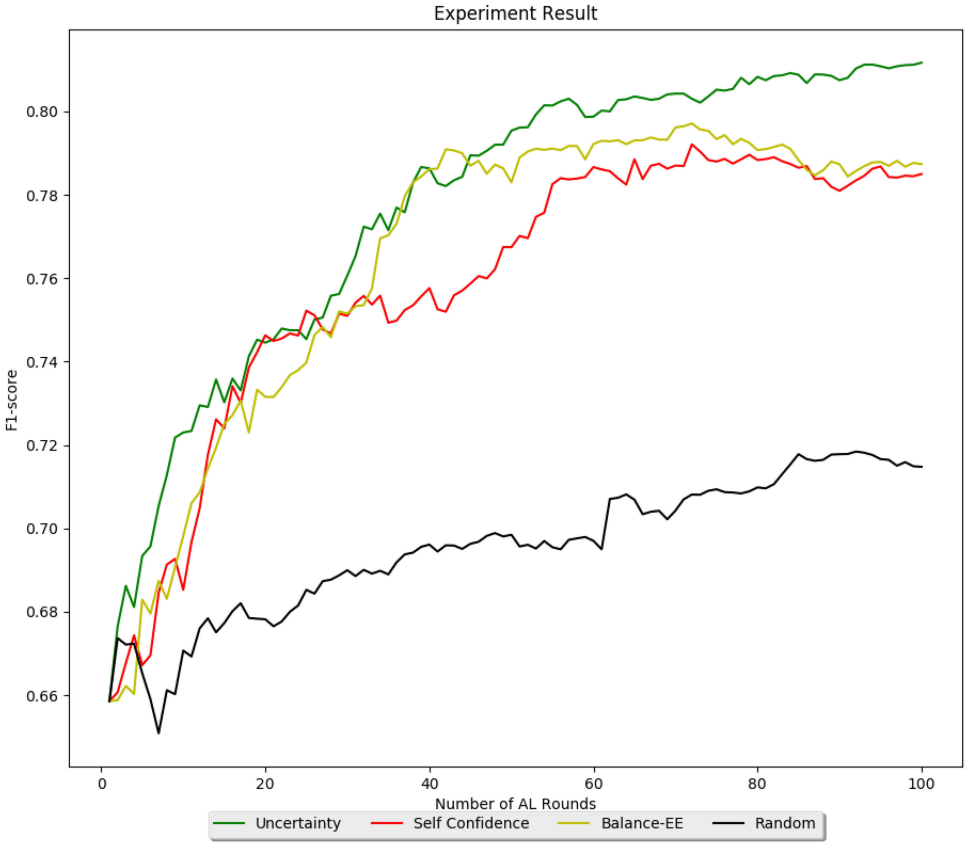

Fig. 7 *F1-scores of three AL algorithms and random sampling in classifying the incident severity*

decisions with respect to the operations of the transport network. The performances of most ML algorithms used in this study have been improved significantly by identifying and removing outliers in the train set. Moreover, in case there are restrictions in model size or prediction time, active learning strategies can help to decide the optimal train set with significantly higher performance than random selection of the data of the same size.

Using empirical studies, we have recommended the appropriate ML method for incident severity classification. The model is capable to learn from several years of incident data and make prediction in real time. To further improve classification accuracy, the anomalies detection process based on ensemble method is applied to remove outliers from the datasets. Finally, AL algorithms especially uncertainty sampling can be utilised to optimise the train set or select the most informative incident record to update the model's knowledge.

10 References

[1] Taib, R., Yee, D., Fang, F., *et al.*: 'Improved incident management through anomaly detection in historical records'. Proc. of the 21th ITS World Congress, Detroit, USA, 2014, pp. 11–23

[2] Nowakowska, M.: 'Logistic models in crash severity classification based on road characteristics'. Transportation Research Record: Journal of the Transportation Research Board, No. 2148, Transportation Research Board of the National Academies, Washington, DC, 2010, pp. 16–26

[3] Sohn, S.Y., Shin, H.: 'Pattern recognition for road traffic accident severity in Korea', *Ergonomics*, 2001, **44**, (1), pp. 107–117

[4] Sohn, S.Y., Lee, S.H.: 'Data fusion, ensemble and clustering to improve the classification accuracy for the severity of road traffic accidents in Korea', *Safety Science*, 2003, **41**, (1), pp. 1–14

[5] Tesema, T.B., Abraham, A., Grosan, C.: 'Rule mining and classification of road traffic accidents using adaptive regression trees', *Int. J. Simul.*, 2005, **6**, (10), pp. 80–94

[6] Miranda-Moreno, L., Fu, L., Ukkusuri, S., *et al.*: 'How to incorporate accident severity and vehicle occupancy into the hot spot identification process?'. Transport. Res. Record, Washington, D.C., 2009, pp. 53–60

[7] Chong, M., Abraham, A., Paprzycki, M.: 'Traffic accident analysis using machine learning paradigms', *Informatica*, 2005, **29**, (1), pp. 89–98

[8] Hall, M., Frank, E., Holmes, G., *et al.*: 'The WEKA data mining software: an update', *ACM SIGKDD Explorations Newsletter*, 2009, **11**, (1), pp. 10–18

[9] Rish, I.: 'An empirical study of the naive Bayes classifier'. IJCAI 2001 workshop on empirical methods in artificial intelligence, IBM New York, 2001, vol. **3**, no. 22, pp. 41–46

[10] Lewis, D.: 'Naive (Bayes) at forty: the independence assumption in information retrieval'. Machine learning: ECML, Springer Berlin Heidelberg, 1998, pp. 4–15

[11] Joachims, T.: '*Text categorization with support vector machines: learning with many relevant features*'. In Claire, N., Céline, R. (Eds.): (Springer, Berlin Heidelberg, 1998), pp. 137–142

[12] Quinlan, J.R.: '*C4. 5: programs for machine learning*' (Elsevier, 2014)

[13] Kohavi, R.: 'A study of cross-validation and bootstrap for accuracy estimation and model selection'. IJCAI, 1995, vol. **14, no**. 2, pp. 1137–1145

[14] Hodge, V.J., Austin, J.: 'A survey of outlier detection methodologies', *Artif. Intell. Rev.*, 2004, **22**, (2), pp. 85–126

[15] Settles, B.: 'Active learning literature survey'. University of Wisconsin, Madison, 2010, **52,** (55–66), 11

[16] Schohn, G., Cohn, D.: 'Less is more: active learning with support vector machines'. ICML, 2000, pp. 839–846

[17] Tong, S., Koller, D.: 'Support vector machine active learning with applications to text classification', *J. Mach. Learn. Res.*, 2001, **2**, pp. 45–66

[18] Campbell, C., Cristianini, N., Smola, A.: 'Query learning with large margin classifiers'. ICML, 2000, pp. 111–118

[19] Roy, N., McCallum, A.: 'Toward optimal active learning through Monte Carlo estimation of error reduction'. ICML, Williamstown, 2001, pp. 441–448

[20] Nguyen, D.H., Patrick, J.D.: 'Supervised machine learning and active learning in classification of radiology reports', *J. Am. Med. Informatics Assoc.*, 2014, **21**, (5), pp. 893–901

[21] Osugi, T., Kim, D., Scott, S.: 'Balancing exploration and exploitation: a new algorithm for active machine learning'. Fifth IEEE Int. Conf. on Data Mining, IEEE, November 2005, p. 8–pp