

World Conference on Transport Research - WCTR 2019 Mumbai 26-31 May 2019

A random forest incident detection algorithm that incorporates contexts

Jonny Evans^{a*}, Ben Waterson^a, Andrew Hamilton^b

^a *University of Southampton, Southampton, SO16 7QF, U.K.*

^b *Siemens Mobility Limited, Sopers Lane, Poole, BH17 7ER, U.K.*

Abstract

A major problem faced by state of the art incident detection algorithms is their high false alert rates, which are caused in part by failing to differentiate incidents from contexts. Contexts are referred to as external factors that could be expected to influence traffic conditions, such as sporting events, public holidays and weather conditions. This paper presents RoadCast Incident Detection (RCID), an algorithm that aims to make this differentiation by gaining a better understanding of conditions that could be expected during contexts' disruption. RCID is based on a previously developed random forest traffic forecasting algorithm, RoadCast, which uses contextual data to create forecasts of traffic conditions that could be expected if no incident occurred. RCID compares these forecasts with real-time conditions, and raises alerts when there is a sufficient difference. RCID was evaluated on loop detector flow data and city council incident logs from Southampton, U.K. Comparisons were made with and without context, and to a state of the art algorithm, RAID. RCID was found to outperform RAID in terms of detection rate and false alert rate. RCID was also found to have a 25% lower false alert rate when incorporating contextual data. This improvement suggests that if RCID were to be implemented in a Traffic Management Centre, operators would be distracted by far fewer false alerts from contexts than is currently the case with state of the art algorithms, and so could detect incidents more effectively.

© 2018 The Authors. Published by Elsevier B.V.

Peer-review under responsibility of WORLD CONFERENCE ON TRANSPORT RESEARCH SOCIETY.

Keywords: Random forest; Traffic flow prediction; big data; machine learning; context

1. Introduction

Road congestion places a burden on citizens worldwide. In 2016 alone, road congestion cost U.S. drivers more than \$295 billion, U.K. drivers £30 billion, and German drivers €69 billion (Cookson and Pishue (2016)). A major

* Corresponding author: Jonny Evans. *E-mail address:* jonny.evans@soton.ac.uk

cause of this congestion is from incidents (Chin et al. (2016)). Incident detection algorithms (IDAs) help traffic management centres (TMCs) detect incidents more quickly, allowing their disruption to be minimised by responding more quickly and effectively.

The causes of disruption in traffic conditions can be categorised into two main types, incidents and contexts. Incidents are defined as unexpected events that disrupt traffic conditions (Cambridge Systematics (2001)). Examples include vehicle collisions, illegal parking and unloading, vehicle breakdowns and emergency roadworks. Contexts are referred to as external factors that are planned in advance or predictable, and could be expected to influence traffic conditions at a particular time in the future. Examples include planned roadworks, sporting events, rush hours, schools closing and weather conditions. The key difference is that contexts could be expected to occur, but incidents are inherently unexpected.

High false alert rates have been found to be the ‘primary and most commonly cited’ deterrent of the deployment of IDAs in TMCs (Williams and Guin (2007)). This limitation is often reported to be caused by failing to differentiate between disruption from contexts and incidents (Balke (1993), Parkany and Xie (2005)). High false alert rates are a problem in practice because they distract TMC operators, which has often led to IDAs being ignored and discarded (Williams and Guin (2007), Parkany and Xie (2005)).

Vast arrays of IDAs have been presented in the literature, but few have given focus to the problem of differentiating incidents from contexts (Parkany and Xie (2005)). Of those IDAs that have, many different approaches have been taken. Some assumed that incidents and contexts are characteristically different in terms of traffic conditions, and so attempted to ‘learn’ the conditions expected in both situations from traffic data (Persaud et al. (1990), Payne and Tignor (1978)). But others have argued that they are too similar to tell apart from traffic conditions alone (Cook and Cleveland (1974)). Some IDAs left the responsibility of differentiation with the TMC operator, such as RAID (Cherrett et al. (2002)). However, this approach is unlikely to be suitable for implementations in large networks with congestion occurring frequently, because of the time required of the operator to filter out the false alerts (Williams and Guin (2007)).

Some IDAs have taken data from external sources to improve the understanding of the prior likelihood of incidents occurring (Lam et al. (2016)). Such sources include the weather, road geometry and speed limits. In situations where an incident is deemed more likely to have occurred, IDAs have been made more sensitive to raising alerts, which has been found to improve performance. Although this approach does not explicitly tackle the problem of differentiating incidents from contexts, it does show that IDA performance can be improved by incorporating external data sources. However, in real-world implementations, the number of incidents that occur in each scenario (e.g. weather condition, speed limit etc.) is often very infrequent. This means that this approach would either require vast amounts of data for training, or a very limited understanding of the prior likelihood of an incident would be gained.

Presented is a novel IDA, RoadCast Incident Detection (RCID), which is the first to be based on a traffic forecast that incorporates contextual data. The approach is to raise alerts when real-time traffic conditions differ from a context-based traffic forecast. This approach attempts to use contextual data to better understand the variation in traffic conditions that can be expected from contexts, allowing contexts to be better differentiated from incidents, and hence reducing false alerts and improving detection rates. RCID also has the advantage of only attempting understanding of conditions that could be expected to occur in the case that no incident occurs, and so should require comparatively less data for training. IDAs that attempt to understand conditions expected in both incident and non-incident scenarios often require far more training data because of the infrequency of incidents (Lee et al. (1998), Khan and Ritchie (1998)). The aim of this research is to understand the extent to which the proposed approach is able to differentiate incidents from contexts, and hence improve on the performance of state of the art IDAs.

2. Methodology

The methodology of RCID can be described as two key steps. Firstly, to create a forecast of a target traffic variable (e.g. flow) for what would be expected if no incident were to occur (herein ‘expected’). Then, to compare this forecast to real-time values of the target variable, and to raise an alert when a sufficient difference is observed. The sections below describe the traffic forecasting algorithm, and the incident detection logic.

2.1. Traffic forecasting algorithm

Integral to RCID is an accurate traffic forecasting algorithm that forecasts expected traffic conditions. For RCID to be able to differentiate incidents from contexts, the algorithm needs to be able to accurately forecast the disruption from contexts, but be unable to accurately forecast the disruption from incidents (which could be inferable if, for example, recent traffic condition observations were used as input).

A previously developed random forest algorithm, RoadCast, was considered for use as the required traffic forecasting algorithm (Evans et al. (2018)). RoadCast was developed with the aim to forecast traffic conditions at a horizon of up to one year. As such, it used input features that would account for the medium and long term variation in traffic conditions (such as the day of the week), rather than being based on recent traffic observations. It also incorporated contextual data with the aim of improving its accuracy.

RoadCast used one random forest algorithm for each detector and each target variable being forecasted. A random forest approach was chosen because it was most accurate in preliminary tests relative to other machine learning and statistical approaches. It was also found to have quick training and testing times relative to other complex machine learning algorithms, which would be important for the practicality of implementation in ITS applications. Algorithm 2 describes the random forest algorithm used in RoadCast, which is an ensemble method that uses a collection of decision trees (algorithm 1). (Breiman (2001)) provides detail of the theory of the random forest algorithm. The algorithm was developed using the Scikit-learn library in Python (Pedregosa et al. (2011)).

Algorithm 1. Decision tree algorithm

Procedure: Training (set of training messages Z^{tr})
 Create a node B_0 and assign all training messages Z^{tr} to it
 While every leaf has more than M messages assigned to it:
 Find the leaf node B_i with the most messages
 From a random subset of features of size S , find the attribute a to split B_i 's messages into two subsets such that the sum of the variances of each subset's target variable values is minimised
 Create child nodes B_j and B_{j+1} from B_i
 Assign B_i 's messages to B_j and B_{j+1} according to their value of a
 End procedure

Algorithm 2. Random forest algorithm

Procedure: Training (set of training messages Z^{tr})
 For a pre defined number of trees K do:
 Create a bootstrap random sample Z_r^{tr} from Z^{tr} of size $|Z^{tr}|$
 Create a decision tree T_r with Z_r^{tr} using algorithm 1
 End procedure
 Procedure: Testing (set of testing messages Z^{ts})
 For each message x in Z^{ts} do:
 Predict a value y_i for message x using each of the decision trees $T_1 \dots T_k$
 Return the mean of the predicted values \bar{y}
 End procedure

RoadCast was tested by forecasting messages of five minute flows and average speeds from loop detectors in Southampton, U.K., and was compared to a historical average, which is a commonly used predictor in ITS applications (Chrobok (2000), Syrjarinne (2016)). Contexts that could be expected to disrupt Southampton's traffic conditions were incorporated as input features in the algorithm. Overall, RoadCast was found to be more accurate than the historical average by 4.4% and 4.0% mean squared error for flow and average speed respectively. It was

also found to be able to use contexts to improve its forecasts, which it did by ‘learning’ from the disruption caused by previous occurrences of contexts in the training data. Comparing the flow forecasts of RoadCast and the historical average, RoadCast was 32% more accurate over the Christmas holiday, 27% more accurate over Easter, and 7.3% more accurate on the day of a football match (note that disruption could only be seen for a couple of hours of the day). The benefit of using machine learning to incorporate contexts was that it could automatically ‘learn’ how each context, or combination of contexts, could be expected to affect each detector at each time.

It was clear that the forecasts produced by RoadCast would be unable to forecast the disruption from incidents, because the horizon used was up to one year and incidents cannot be predicted before they occur (incidents’ disruption rarely last for multiple hours, so the disruption could not be forecast if a horizon of multiple hours were used). RoadCast also demonstrated the ability to accurately forecast traffic conditions, including the disruption from contexts. Because of this, the methodology of RoadCast was chosen to be developed for use as the traffic forecasting algorithm within the incident detection algorithm RCID.

A key consideration of RCID was the amount of manual calibration required for implementation. The time, expertise and manual labour required to calibrate IDAs is another common reason for lack of use in TMCs (Williams and Guin (2007), Parkany and Xie (2005)). Commonly required manual calibration requirements include the setting of algorithm parameters (often at each detector individually), creating traffic simulations of the road network for training, and the collection and pre-processing of various datasets for training (traffic, context, incidents etc.). Because of this, standard methods to encode data into input features were developed, and a previously developed automatic optimisation algorithm was re-used to calibrate RoadCast (Evans et al. (2018)).

The standard encoding methods can be seen in table 1. These methods were developed for the intention that they could be re-used to encode different contextual features when implementing RCID in new locations (e.g. St Andrew’s Day in Scotland), ensuring accuracy while saving time and expertise required for implementation. Note that a multiple day event context (with reference) is an event which ends on a different day than it starts, and has a particular time/day of interest (i.e. the reference) during the event which can occur at different times on different occurrences, such as Christmas Day during the Christmas holiday (which can occur on different days of the week, and different durations from the start of the holiday). The use of this type of context would allow RoadCast to differentiate between different important days during the event. The modified day of week feature would stop an issue that decision trees would often split the training data based on the day of the week high up the tree, and hence were unable to forecast using contexts that happened to occur on different days of the week in the training data.

Table 1. Encoding methods and features

| Feature type | Standard encoding method | Feature used in this study |
|---|--|--|
| Time of day | Hour of day + (minutes/60) | Time of day |
| Day of week | Integer ranging from 0 to 6 based on the day of the week | Day of week |
| Modified day of week | If during a multiple day event (with reference): 7 Else: Integer ranging from 0 to 6 based on the day of the week | Modified day of week (used when a multiple day event (with reference) is included) |
| Single day events | If on the day of the event: The number of days + hours/24 + minutes/1440 + until the start of the event Else: 100 | Football matches, half marathon event. |
| Multiple day events (without reference) | If during the event: The number of days + hours/24 + minutes/1440 + until the end of the event Else: 0 | Easter, other public holidays. |

| | | |
|--------------------------------------|---|--|
| Multiple day events (with reference) | If during the event: | Christmas (defined as starting on the first public holiday day before Christmas Day, and ending on the first working day after New Year's Day) |
| | Else: | |
| | The number of days + hours/24 + minutes/1440 + until the reference time | |
| | 100 | |

The optimisation algorithm can be seen in algorithm 3 below. Its aim is to find the optimal contextual features and random forest parameters at each detector, by running a grid search method with cross-validated tests on the training data with different combinations of contexts and random forest parameters. It was found to result in accuracy improvements because certain detectors were more suited to certain parameter values (which appeared to be correlated to the amount of noise at each detector), and because contexts that did not disrupt a detector's traffic conditions would at times result in over-fitting, due to decision trees splitting on the contextual feature unnecessarily. The optimisation algorithm did not require any manual calibration, but improved RoadCast's accuracy by tailoring it to each detector. Further explanation of the optimisation algorithm can be found in reference (Evans et al. (2018)).

Algorithm 3. RoadCast optimisation algorithm

```

Procedure: Context inclusion (set of training messages  $Z^{tr}$ , set of contextual features  $A$ )
  Shuffle the order of the messages
  Set the benchmark score as the score on  $Z^{tr}$  with 'time of day' and 'day of week' features only
  For each feature in  $A$ :
    If the score does not improve when the feature is added:
      Remove the feature from  $A$ 
    End if
  End for
  Set the benchmark score as the score with the features currently in the algorithm
  For each feature in  $A$ :
    Remove the feature from the algorithm and find the algorithm's score
    If the score improves on the benchmark on at least two folds:
      Remove the feature from  $A$ 
    End if
  End for
  If a multiple day event (with reference) feature is included:
    Replace the 'day of week' feature with 'modified day of week' end if
End procedure
Procedure: Grid search parameter optimisation (set of training messages  $Z^{tr}$ , set of features included in the algorithm  $F$ )
  for  $M$  in [2, 5, 10, 25, 100, 200]:
    for  $S = 1$  to  $|F|$ :
      Find the score with parameters  $M$  and  $S$ 
    End for
  End for
  Return the parameters that achieved the best score,  $M^*$  and  $S^*$ 
  Retrain the algorithm on all available training data with parameters  $M^*$ ,  $S^*$  and  $K = 100$ 
End procedure

```

The implementation procedure of RoadCast is to first identify contexts local to the network being implemented (e.g. Southampton F.C. football matches in Southampton). Then, historical traffic data and contextual data are collected over a particular period for training. At least one year of historical data is recommended, so that all annually occurring contexts can be 'learnt' from in training. For the future time period to be forecasted, data for RoadCast's inputs must also be collected, including contextual data (schedules of contexts) and information on the time of day and day of week. Next, the contextual data is encoded using the standard encoding methods in table 1. The optimisation algorithm is then run on the historical data. At this point, forecasts for the future time period are ready to be made.

The RoadCast algorithm presented in reference (Evans et al. (2018)) would produce a prediction of a single value for each message, representing the algorithm's 'best guess'. This prediction would not suit the incident detection

application because it would not account for prediction uncertainty. Clearly, the uncertainty of a forecasting algorithm's prediction can vary based on the message being forecast. For example, football matches in Southampton appeared to have more variation in disruption between occurrences than public holidays, resulting in more uncertainty in future forecasts. As such, RCID would improve its performance if it were less sensitive to raising alerts when the forecast was less certain, and vice versa. Hence, it would be more suitable for RCID to raise alerts when real-time values of the traffic variable fell outside a range of expected values, i.e. a prediction interval, rather than a pre-set difference from a single value prediction. A benefit of the random forest algorithm is that there exist methods to produce prediction intervals (Meinshausen (2006)). As such, RoadCast would be modified to produce prediction intervals, which would be used as input to RCID.

A prediction interval is an estimate of an interval for which future observations (of the target variable) will fall into with a given probability. The method in reference (Meinshausen (2006)) was implemented to acquire these prediction intervals. A random forest's forecast is the mean of each tree's forecast, and each tree's forecast is the mean of the target variable values in the tree's predicted leaf. Instead of using this, prediction intervals were created by taking the appropriate percentiles of all the target variable values of the messages in every tree's predicted leaf. For example, a 95% interval is the range from the 2.5th and 97.5th percentiles of the values. This means that real-time traffic variable values should fall within the prediction interval approximately 95% of the time.

2.2. Incident detection logic

This section describes RCID's use of the traffic forecasting algorithm's prediction intervals, in order to raise incident alerts in real-time. In a preliminary test on the training data, RCID would simply raise an alert when real-time values of the target variable fell outside of the prediction interval. However, variation from noise in the traffic data would result in many unnecessary false alerts. As such, a persistence test of three messages was introduced. This would ensure that alerts would only be raised when the underlying trend of the target variable had truly deviated from what the forecasting algorithm expected. This persistence test would improve RCID's false alert and detection rate, but would worsen its average time to detect.

3. Data

3.1. Traffic data

Southampton City Council provided the traffic data for this study. Figure 1 shows the location of the 109 single inductive loop detectors used. 726 days worth of data was collected from 16th March 2015 to 16th March 2017 (5 days of data were missing). The first year of data was used for training, and the period between 14th December 2016 and 16th March 2017 was used for testing. Flow values from detectors' messages, i.e. the number of vehicles in each five minute period (over the lane of the detector) were used as the target variable in this study. RoadCast would be implemented on each detector separately. At times, some detectors would return messages with zero flow due to detector system fault. As such, all messages of zero flow (plus one message before and afterwards) were removed. Although this method would remove some representative messages (e.g. during the night), it would ensure that none of the unrepresentative messages would be considered in the training or evaluation of RCID.

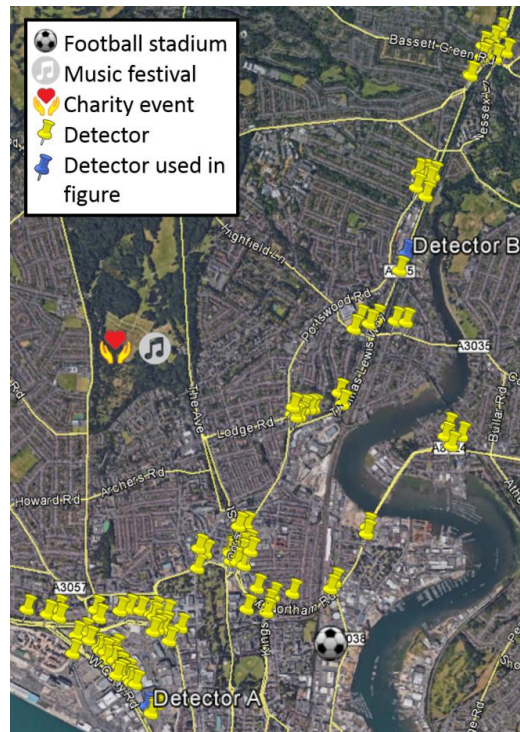


Fig. 1. Locations of the detectors used in this study. This image was created with Google Earth.

3.2. Contextual data

In a previous study, the disruptive contexts in Southampton were identified. The methods employed to identify these contexts can be found in reference (Evans et al. (2018)). In this study, these disruptive contexts were again used as inputs to RoadCast. Table 1 shows each of the features used in this study, alongside the method used to encode each feature.

A caveat of this study was that the contextual data used to create RoadCast's forecasts was collected after the contexts took place. If RoadCast were to make forecasts into the future, it would need to use schedules of these contexts, which may change before the event (such as rescheduled football matches). If contexts were rescheduled, RoadCast could account for this if it re-made its forecasts with updated contextual features, albeit at a shorter forecasting horizon.

3.3. Incident data

Incident data was collected from a Twitter feed provided by Southampton City Council and Balfour Beatty (Southampton City Council (2018)). The feed takes incident logs created by operators at the Council's TMC, and disseminates incident information to the public via 'tweets'. The tweets covered the testing period of 14th December 2016 to 16th March 2017. By comparing this dataset with the available loop detector data and cross-referencing with other online sources, including the STATS19 crash dataset (UK Government (2018)), this Twitter feed was judged to have sufficient coverage and reporting quality to evaluate RCID.

However, not all of the tweets on the feed were suitable for the evaluation of RCID. Firstly, many described disruption from contexts rather than incidents. As such, only tweets with a description of an incident were considered. RCID could not be reasonably expected to detect incidents that did not cause any disruption to a

detector's traffic conditions. Hence, to ascertain which detectors were affected by which incidents, each tweet of an incident was investigated by manually observing nearby loop detectors' traffic data and historical average values. Only tweets of incidents which visibly disrupted at least one detector's traffic conditions (of any of its variables) were considered. After completing this process, 28 cases of an incident disrupting a detector's traffic conditions were identified.

4. Results

Using the described methodology, RCID was implemented on the study's traffic flow, context and incident datasets. Figure 2 shows how often the optimisation algorithm included each context (out of a possible 109 detectors). It can be seen that holiday features were used most often, and that the football feature was used more often than the half marathon feature due to the greater travel demand created. It could be seen that holiday features affected detectors throughout the city, but event contexts were only included at detectors on routes into and out of the event location.

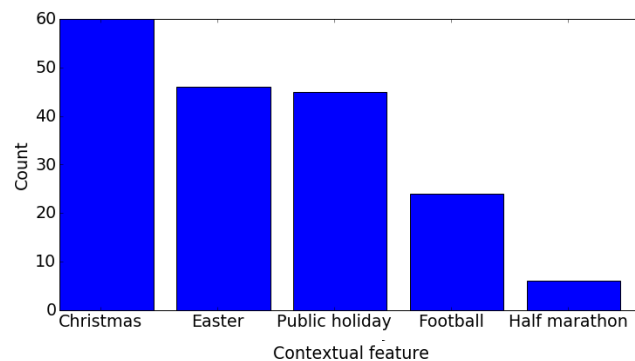


Fig. 2. Bar chart showing the number of times each context was chosen for use by the optimisation algorithm.

The following sections evaluate the performance of RCID, and make comparisons to an existing IDA, RAID.

4.1. IDA comparison

RCID would be compared to a state of the art IDA, RAID (Cherrett et al. (2002)). RAID used average loop-occupancy time per vehicle (ALOTPV) and average time-gap between vehicles (ATGBV) to detect incidents. ALOTPV is the average time period that each vehicle spends occupying the road space above a loop detector, and ATGBV is the average time period in-between each vehicle occupying a detector. Each variable was calculated directly from the 'occupied' or 'non-occupied' states of the loop detectors which were sampled every 0.25 seconds. The IDA would judge a message as being representative of an incident if it was above the 85th percentile of the training data ALOTPV values, and below the 15th percentile of ATGBV values in the given peak or off-peak period. Peak periods were defined as being 07:00-09:30 and 16:00-19:00. If the values broke these thresholds for three consecutive messages during the off-peak period, or four consecutive messages during the peak period, an incident alert would be raised. This alert would then stop when either of the thresholds was not met. Although RAID was originally developed for use on 30 second values of ALOTPV and ATGBV, it is thought that the logic would transfer across to the five minute values used in this study.

RCID would also be tested multiple times with different prediction intervals in order to understand the trade-off between different performance metrics. With a greater prediction interval, RCID would be less sensitive to raising alerts, meaning that a relatively better false alert rate but worse detection rate would be expected. The prediction intervals used were 90%, 93%, 95%, 97% and 99%. To understand whether the incorporation of contexts can

improve IDAs’ performance, RCID would also be tested with and without contextual data. That is, RCID (with context) would use a version of RoadCast with access to all the available input features (described in table 1), and RCID (without context) would use a version of RoadCast that only used the input features ‘time of day’ and ‘day of week’.

4.2. Performance metrics

The most commonly used performance measures of IDAs are detection rate (DR), false alert rate (FAR) and average time to detect. Unfortunately, the exact time of incidents was not stated in the incident tweets. Because there would be a variable delay between incidents occurring, operators detecting them, and tweets being posted, the time-stamp of tweets would also be unsuitable for evaluating RCID’s average time to detect. As such, only the detection and false alert rate were used as performance metrics.

RCID would be judged to have correctly detected an incident if an alert was raised while an incident was disrupting the detector’s traffic conditions (this period was ascertained by comparing the detector’s traffic data with a historical average). DR was defined as the number of correctly detected incidents divided by the total number of incidents (from the Twitter dataset). FAR was defined as the number of messages where an alert was raised incorrectly, divided by the total number of messages where an incident was not occurring. Another metric, FAR_{pdpd} , was also used to give a more clear understanding of the number of false alerts that TMC operators could expect when implemented. FAR_{pdpd} was defined as the number of false alerts raised per detector per day. Note that an incident alert could span multiple consecutive messages.

4.3. Performance

RCID was found to outperform RAID in terms of detection rate and false alert rate (with a 95% and 97% prediction interval). RCID was also found to be able to reduce its false alert rate by at least 25% by incorporating contextual data (at least 25% at every prediction interval used). Such improvements could be seen to be because of an increased ability to forecast the disruption caused by contexts, and hence differentiate contexts from incidents more effectively.

As expected, there is a trade-off to be made between detection and false alert rate. Figure 3 shows that with a low percentage prediction interval, RCID (with context) had a better DR and worse FAR than RAID, and vice versa for higher percentage intervals. However, for 95% and 97% intervals, RCID (with context) had a better DR and FAR than RAID. Comparing RAID to RCID (with context) with a 97% prediction interval, RCID had a 27% higher detection rate (68% against 41%) and a 0.29% lower false alert rate (0.49% against 0.78%). At this prediction interval, it was also found to improve its false alert rate from 0.77% to 0.49% by using contexts.

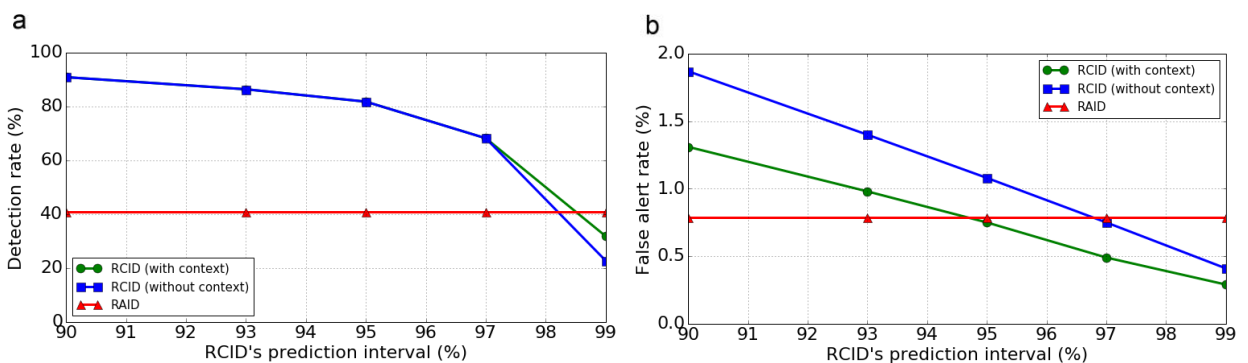


Fig. 3. IDAs’ performance. (a) Detection rates (b) False alert rates.

A survey of TMC operators found that acceptable IDA performance boundaries would be at least 88.3% detection rate, and at most 1.8% false alert rate (Ritchie and Abdulhai (1997)). With a 90% prediction interval, RCID (with context), met these boundaries, with a 90.9% DR and 1.31% FAR. However, RCID (without context) and RAID did not meet these boundaries. This suggests that if RCID (with context) were to be implemented in a TMC, operators may find the performance acceptable enough to detect incidents effectively, unlike previous IDAs which were often ignored or disabled.

Figure 4 shows how RCID used the football context to ‘learn’ what disruption could be expected, resulting in it not raising a false alert before the match. RCID (without context) raised a false alert because it did not accurately forecast the context’s disruption. In general, RCID (with context) was able to ‘learn’ what disruption could be expected from each of the contexts used, and so was better at differentiating incidents from contexts, and hence had a lower false alert rate.

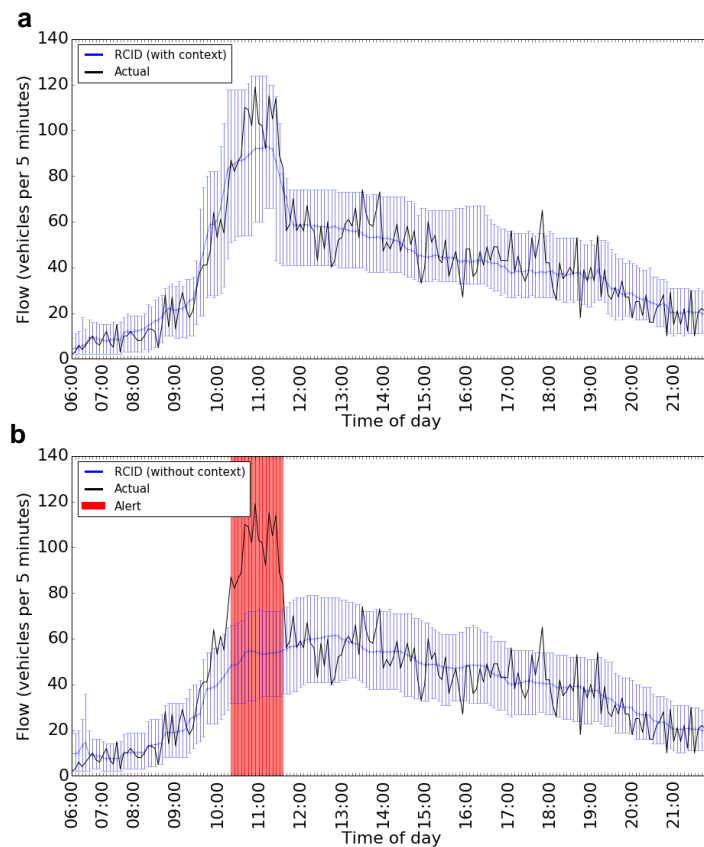


Fig. 4. RCID’s alerts (with and without context) on the day of a Premier League Football match against Leicester F.C., which kicked off at 12:00 at St Mary’s Stadium. No incident occurred. Sunday 22nd January 2017, at detector B. RCID used a 90% prediction interval. Note that the forecasts and prediction intervals are indicated by the blue lines and error bars respectively, and the red highlighted areas are times when the IDA raised an alert. (a) RCID (with context) (b) RCID (without context).

At the typical time of day and day of the week of a context’s disruption, RCID (without context) would often create prediction intervals wide enough to cover the context’s disruption, both when the context occurred and when it didn’t. This occurred more often and to a greater extent when higher percentage prediction intervals were used. Figure 5 shows a wide prediction interval caused by the disruption from occasional weekday evening football matches. This may have caused the incident to go undetected if it occurred an hour later. In general, RCID was seen

to be less effective at detecting incidents when not using contexts (particularly with high percentage prediction intervals), because it would produce more naive and uncertain prediction intervals. In this study, RCID’s detection rate was (largely) the same with and without context because contexts (coincidentally) did not disrupt any of the 28 incidents in the test dataset, but this may not be the case for repeated tests on different datasets. Figure 5 also shows RAID failing to detect an incident because ALOTPV and ATGBV values were not disrupted sufficiently. In general, RAID was found to be somewhat effective at detecting congestion, but performed worse than RCID because it would raise false alerts during context caused disruption, and it would fail to detect incidents that didn’t cause congestion.

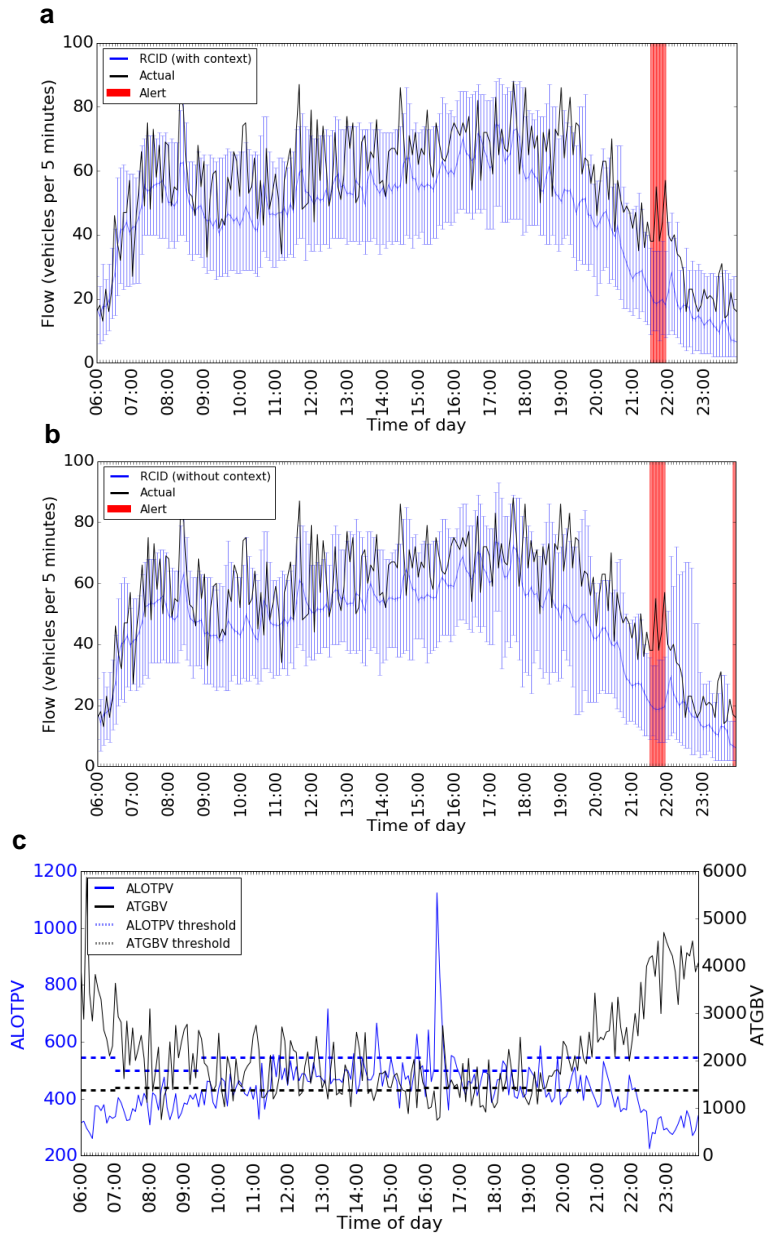


Fig. 5. RCID (with and without context) and RAID’s alerts at a time where an emergency roadworks incident caused one lane of a nearby roundabout to be closed, causing disruption between 6pm and 11pm. Thursday 15th December, at detector A. A 93% prediction interval was used.

(a) RCID (with context) (b) RCID (without context) (c) RAID.

RCID (with context)'s false alert rate was limited by occasional inaccurate predictions by RoadCast, caused by variations in the traffic data that was not accounted for. Some causes of variation may have been missed, and others may not have been suitable for incorporation, such as noise or disruption during particularly busy shopping days, which could be identified (and verified by Southampton City Council tweets), but not predicted beforehand. The detection rate was most limited by failing to detect incidents that caused minor amounts of disruption. In these cases, the prediction intervals were too wide because of RoadCast's forecasting uncertainty, which stemmed from the unaccounted causes of variation in the data.

RCID (without context) often raised false alerts when contexts caused disruption (as can be seen in figure 4). However, in some cases it would not raise false alerts for contexts, particularly for contexts that occurred frequently at a particular time or day of the week, such as football matches at 3pm on Saturdays. As can be seen in figure 6, at times the prediction interval was wide enough to cover the context's disruption, because the messages in the predicted leaves were from both times when a match was occurring and when it wasn't. With a 95% prediction interval, one could assume that if a context caused disruption at a particular time and day of the week on less than 2.5% of occasions in the training period, RCID (without context) could be susceptible to raising false alerts on these occasions in the testing period. However, such wide prediction intervals made RCID more susceptible to failing to detect incidents at times when the particular context does not occur. In general, RCID produced more naive and uncertain prediction intervals when contexts weren't incorporated, and hence was less effective at detecting incidents.

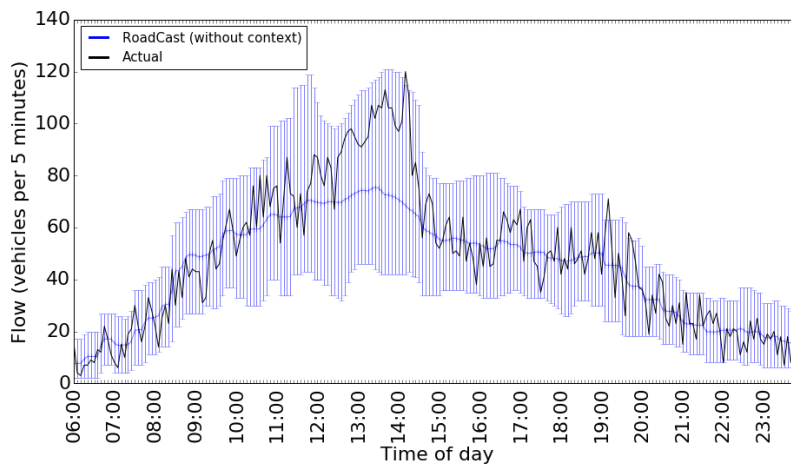


Fig. 6. RCID (without context) with a 95% prediction interval, on a day when no incident occurred. Saturday, 4th February 2017, at detector B. Premier League football match against West Ham F.C. kicked off at 15:00 at St Mary's Stadium.

In general, RAID was found to be able to detect congestion, but performed worse than RCID (with context) due to an inability to differentiate contexts' disruption from incidents. Its performance was limited in two ways, it would raise false alerts during context caused congestion, and it would fail to detect incidents that didn't cause congestion. Also, at times, the pre-defined on and off peak thresholds did not meet their objective of capturing the time of day variance in ALOTPV and ATGBV (see that the on-peak ALOTPV threshold is lower than the off-peak threshold in figure 5 (c)). These thresholds appeared not to capture the variation in the study's traffic data because different detectors had different peak times, and peak times differed for different days of the week.

For all of the IDAs tested, many of the false alerts came from a few detectors which were particularly noisy or appeared to have a step change in values. For example, the detector which produced the most false alerts for RCID appeared to have visibly higher flow values (and hence false alerts) after 22nd August 2016. The cause of this step change could not be found, but could have been caused by a change in nearby road capacity or travel demand, such as a new road lane or nearby shopping mall being built. This is an issue for which all IDAs that attempt to understand the expected traffic conditions could be expected to suffer from. However, if this issue was identified by

an operator in a TMC, this issue could easily be rectified by retraining the algorithm on data in the time period since the step change. Another limitation of each of the IDAs may have been the average time to detect. Unfortunately, this couldn't be evaluated in this study because the exact time of incidents occurrence was unknown. However, based on the persistence test used, it could be expected to be at least 15 minutes, which is higher than the reported value of many IDAs presented in the literature. This issue stemmed from the IDAs using messages over long time periods (i.e. 5 minute messages). If 30 second messages were used instead of 5 minute messages, the IDAs could have used a persistence test over a shorter time period, and hence detected incidents more quickly.

5. Conclusions

This paper aimed to tackle the problem of state of the art IDAs creating unnecessary false alerts by failing to differentiate incidents from contexts. Such false alerts distract operators, and had led to many IDAs being disabled or simply ignored. This paper presented and evaluated RCID, a novel random forest incident detection algorithm which aimed to use contextual data to better differentiate incidents from contexts, and hence improve on the performance of state of the art IDAs. RCID was evaluated on loop detector flow data and TMC incident logs from Southampton, U.K. Comparisons were made with and without context, and to a state of the art IDA, RAID.

RCID was found to outperform RAID in terms of detection rate and false alert rate. RCID was also found to reduce its false alert rate by at least 25% when incorporating contextual data (at least 25% at every prediction interval used). This improvement came from RCID's ability to differentiate incidents from contexts by 'learning' how contexts could be expected to disrupt traffic conditions. This improvement suggests that if RCID were to be implemented in a Traffic Management Centre, operators would be distracted by far fewer false alerts from contexts than is currently the case with state of the art algorithms. This would enable operators to detect incidents more effectively, and hence respond more effectively in order to minimise the disruption caused.

A benefit of the random forest algorithm used is that methods exist to interpret its forecasts (Palczewska et al. (2013)). Hence, with further work, it may be possible for RCID to provide information on its reasoning and decision making process, rather than simply raising alerts. Such information could be a message to operators of 'no incident present, disruption caused by Southampton F.C. football match', or 'incident present, disruption also caused by Southampton marathon'. This information has not been supplied to operators of an IDA previously. Doing so could improve TMC operators' trust and effectiveness of using IDAs, and could provide information that would be useful for operators' in responding to incidents.

6. Acknowledgement

The authors wish to thank Southampton City Council for providing the traffic data used in this research.

References

- A.Palczewska, J.Palczewski, R.M.Robinson, and D.Neagu, Interpreting random forest models using a feature contribution method, 2013, in IEEE 14th International Conference on Information Reuse and Integration (IRI), pp. 112–119.
- A. R. Cook and D. E. Cleveland, Detection of freeway capacity-reducing incidents by traffic-stream measurements, 1974, *Transportation Research Record*, vol. 495, pp. 1–11.
- B.Cookson, Graham; Pishue, 2016, INRIX Global Traffic Scorecard, Technology report.
- B.M. Williams and A. Guin, Traffic management center use of incident detection algorithms: Findings of a nationwide survey, 2007, *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 351–358.
- B.N. Persaud, F.L. Hall, and L.M. Hall, Congestion identification aspects of the mcmaster incident detection algorithm, 1990, *Transportation Research Record*, no. 1287.
- Cambridge Systematics et al., Reliability: Providing a highway system with reliable travel times, 2001, Special Report - National Research Council, *Transportation Research Board*, vol. 260, pp. 113–116.
- Chrobok et al., Three categories of traffic data: Historical, current, and predictive, 2000, *Proceedings of the 9th IFAC Symposium Control in Transportation Systems*, p250-255.

- E. Parkany and C. Xie, A complete review of incident detection algorithms and their deployment: what works and what doesn't, 2005 Technology report for the New England Transportation Consortium.
- H.J. Payne and S.C. Tignor, Freeway incident-detection algorithms based on decision trees with states, 1978, *Transportation Research Record*, no. 682.
- J. Evans, B. Waterson, and A. Hamilton, Roadcast: An algorithm to forecast this year's road traffic, 2018, in *Proc. of the 97th Annual Meeting of the Transportation Research Board*.
- K.N. Balke, An evaluation of existing incident detection algorithms, 1993, Interim Report No. FHWA-RD-75-39 for the Texas Department of Transportation.
- L. Breiman, Random forests, 2011. *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001. [16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830.
- N. Meinshausen, Quantile regression forests, 2006, *Journal of Machine Learning Research*, vol. 7, no. Jun, pp. 983–999.
- S.G. Ritchie and B. Abdulhai, Development testing and evaluation of advanced techniques for freeway incident detection, 1997, California Partners for Advanced Transit and Highways (PATH).
- S.I. Khan and S.G. Ritchie, Statistical and neural classifiers to detect traffic operational problems on urban arterials, 1998, *Transportation Research Part C: Emerging Technologies*, vol. 6, no. 5, pp. 291–314.
- S. Lee, R.A. Krammes, and J. Yen, Fuzzy-logic-based incident detection for signalized diamond interchanges, 1998, *Transportation Research Part C: Emerging Technologies*, vol. 6, no. 5, pp. 359–377.
- S-M. Chin, O.Franzese, D.L. Greene, H-L. Hwang, and R. Gibson, 2004, Temporary losses of highway capacity and impacts on performance: Phase 2. United States. Department of Energy.
- Southampton City Council, SCC Highways' Twitter account, <https://twitter.com/scchighways>, accessed March 8, 2018.
- P. Syrjarinne, Urban Traffic Analysis with Bus Location Data, 2016, PhD thesis, Universitatis Tamperensis.
- T. Cherrett, B. Waterson, M. McDonald, R. Clarke, A. Bangert, and R. Morris, Improved network monitoring using utc detector data 'RAID', 2002, *Traffic Engineering and Control*, vol. 43, no. 4, pp. 135–137.
- UK Government, "Road Safety Data," <https://data.gov.uk/dataset/road-accidents-safety-data>, accessed March 8, 2018.
- W.H. Lam, M.L. Tam, and X. Li, Automatic traffic incident detection algorithm for both rain and no-rain conditions, 2016, *Asian transport studies*, vol. 4, no. 2, pp. 330–349.