Distributed Collaborative Incident Validation in a Self-Organised Traffic Control System

Ingo Thomsen[a], Torben Brennecke and Sven Tomforde[b]
Intelligent Systems, Kiel University, 24118 Kiel, Germany

Keywords: Traffic Management, Organic Traffic Control, Traffic Incident Detection, Incident Validation.

Abstract: The continuous trend of raising traffic volumes in urban areas causes waiting times and exhaust emissions. As one promising response to these challenges, increasingly intelligent and adaptive traffic management systems are being developed. For instance, self-organised approaches such as the Organic Traffic Control offer advantages in terms of scalability and robustness compared to traditional systems. This can be increased by taking locally detected incidents into account. To improve the accuracy of automatically detected incidents and to allow for integration in the traffic control strategies, this paper proposes algorithms for the validation of potential incidents. This is done by incorporating respective insights of varying levels from neighbouring intersections and consequently determining a neighbour-supported view of local incident information.

1 INTRODUCTION

The tendency of high traffic demands generated by public transportation or private vehicles is unbroken. This high load leads to congestions, increased pollution and travel delays. For this reason, modern traffic management systems are important for handling these negative effects in urban traffic networks. Typically, these road networks do not feature highly prioritised roads and are not characterised by exits and driveways but by intersections, equipped with traffic light controllers (TLC). The management of these TLCs can be organised in a centralised or decentralised manner or by using various levels in between.

One example of a fully distributed approach is the "Organic Traffic Control" (OTC), a self-adaptive and self-organised traffic management system which offers routing advice to traffic participants in terms of infrastructure-based route guidance (Sommer et al., 2016) as well as flow-dependent and proactive adaptation of TLCs. Such a system can offer decentralised, intersection-centred incident detection (Thomsen et al., 2021a). To this end, OTC will be adapted to use information beyond traffic flows when adapting the TLCs, but without relying on a global view. With less knowledge about the traffic situation, it is desirable to improve the detection accuracy.

Based on the first concept presented in (Tomforde and Thomsen, 2022), a generalised approach is proposed for a collaborative incident validation which is independent of the underlying incident detection mechanism. This generality is realised by handling various levels of knowledge about an incident which is detected with an associated confidence value. It makes use of local neighbourhoods and the properties of traffic volumes as they pass from one intersection to another, resulting in comparable patterns.

The remainder of this article is organised as follows: The next Section 2 provides a brief overview of self-organised traffic control and incident detection, while Section 3 outlines the network and incident model as well as the assumed knowledge levels used in this work. Section 4 then explains the proposed validation approach, followed by an evaluation in Section 5. The final Section 6 offers some conclusions and an outlook on future work.

2 BACKGROUND

This collaborative validation of detected incidents is proposed in the context of the self-organised OTC system whose components are outlined here to provide the background of this work.

[a] https://orcid.org/0000-0002-0850-4786
[b] https://orcid.org/0000-0002-5825-8915

Thomsen, I., Brennecke, T. and Tomforde, S.
Distributed Collaborative Incident Validation in a Self-Organised Traffic Control System.
DOI: 10.5220/0011725000003479
In *Proceedings of the 9th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2023)*, pages 152-159
ISBN: 978-989-758-652-1; ISSN: 2184-495X

2.1 Self-Organised Traffic Control

Traffic light controllers are vital for managing urban road networks. Widely-used systems to control the TLCs are SCOOT (Robertson and Bretherton, 1991), SCATS (Sims and Dobinson, 1980), MOVA (Vincent et al., 1990), and UTOPIA/-SPOT (Mauro and Taranto, 1990): Centralised controller gather data about the traffic situation which is analysed for the change of the signalisation policies. These approaches have limits, as they require all sensor data to be processed centrally and in a timely manner. This results in a single point-of-failure and limits the real-time capabilities.

Alternatively, self-adaptive and self-organised (SASO) systems rely on decision-making, based on the locally assessed traffic situation. This is done by local intersection controllers (IC) which can potentially communicate with neighbouring ICs. Exemplary systems use predictive control (Oliveira and Camponogara, 2010) or multi-agent approaches and fuzzy-logic (Gokulan and Srinivasan, 2010).

2.2 Organic Traffic Control

The SASO system which is the basis for the proposed collaborative incident validation, is the Organic Traffic Control (OTC) (Prothmann et al., 2009). It is designed based on "Organic Computing" (Müller-Schloer and Tomforde, 2017): Principles found in nature are transferred to technical systems to achieve "life-like" behaviour. Small, autonomous entities are used in decentralised structures and combined with machine learning techniques for local adaptation.

The OTC system adheres to the Observer/Controller paradigm (Tomforde et al., 2011). Figure 1 shows how it is realised as multi-level architecture on top of the "System under Control and Observation" (SuOC), the actual (simulated) road network. The detector readings are retrieved and processed in the observer of "online" Layer 1 to create an abstract state of the underlying traffic conditions. This state description is used by the controller component which employs reinforcement learning to modify the traffic signalisation accordingly. Here, a Learning Classifier System is used, namely a variant of Wilson's "Extended Classifier System" (XCS) (Wilson, 1995). When Layer 1 is faced with previously unknown situations, the "offline" Layer 2 is activated: It uses an evolutionary algorithm to create new rules for the XCS which are evaluated using a traffic simulator (Aimsun, 2021).

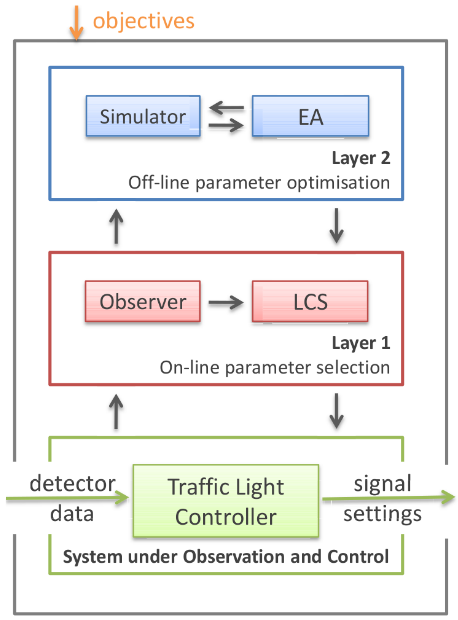

Figure 1: Overview of the multilevel OTC architecture.

2.3 Traffic Incident Detection

The OTC serves as a basis for the "Automated Incident Detection" (AID). Approaches to this have been researched since the seventies. They often rely on loop detectors. For instance, the California Algorithm family (Payne, 1975; Payne and Tignor, 1978) follows decision tree structures based on thresholds. Later techniques may use time series analysis (Ahmed and Cook, 1980), filtering and smoothing-based algorithms (Stephanedes and Chassiakos, 1993), mathematical traffic-flow-model-based algorithms (Lin and Daganzo, 1997), or the usage of probe vehicles to estimate traffic flows (Jenelius and Koutsopoulos, 2013).

There are common limitations to the above as they are designed for highways, rely on previously measured travel times, or do not differentiate the incidents types. To compensate for this, we proposed a novel clustering-based approach for AID in urban road networks (Thomsen et al., 2021b): Induction loop readings are analysed in the form of time series data. With clustering techniques, such as DBSCAN(Ester et al., 1996), significant traffic flow changes can be detected. We demonstrated that especially for traffic demand, appropriate detection accuracy can be achieved.

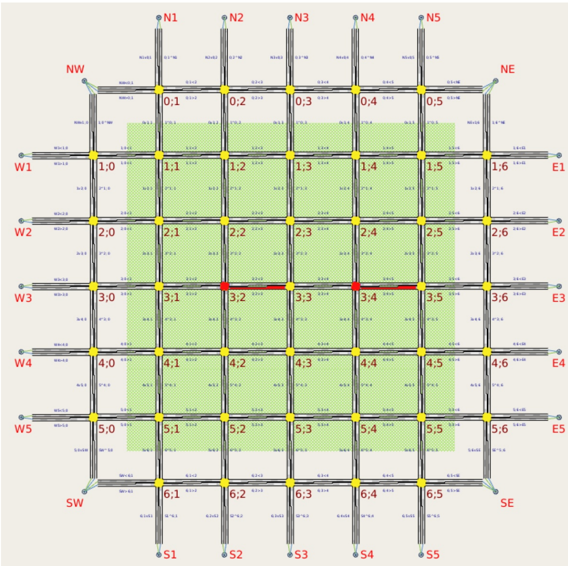

Figure 2: A 7x7-Manhattan grid with two exemplary incidents (marked in red) as simulated using Aimsun. The green area marks the junctions where the TLCs are controlled by the OTC.

3 ROAD NETWORK AND INCIDENT MODEL

The traffic simulator Aimsun Next ((Aimsun, 2021)) is accepted as a realistic model and provides close-to-reality simulations of the road networks and incidents.

3.1 Urban Road Networks

The approaches in Section 4 partly rely on the topology around a junction which is about to validate a locally reported incident. Here, the junctions in the regular Manhattan network Fig. 2 have 4 neighbours and are equipped with TLCs with a phase-based signalisation. The connecting double-lane sections have detectors at both ends.

3.2 Incidents

A traffic incident is an event of a certain duration which changes the traffic capacity of a road, such as depicted in Fig. 3: (SC) complete closing of a section in one direction, (LC) closure of one full-length lane in a multi-lane section, (PLC) partial blockage of a lane, and (TC) blocking of a turning within an intersection. Albeit no traffic incident in itself, the detection validation can be used to recognise certain technical defects at a junction (e.g. loss of function of a traffic light or a detector).

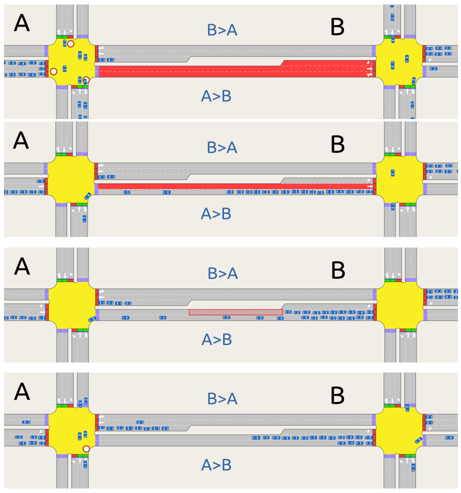

Figure 3: The simulated incident types under consideration: section closure (SC), lane closure (LC), partial lane closure (PLC), and turn closure (TC).

3.3 Incident Detection

The proposed validation is not based on a specific approach to incident detection. The actual mechanism is expected to provide for each incident the information $i = (type, t, c, pos)$, where $type$ denotes one of the 4 incident types, t the start time, c the confidence, and pos the position proposed by the detection. With regard to $type$ and pos, the following 3 knowledge levels (KL) can be distinguished:

1. KL: Neither incident type nor location is known: There is some kind of incident at this junction.
2. KL: The location is known (the specific section or the turn in case of a turn closure).
3. KL: Additionally, the incident type is known.

4 VALIDATION APPROACH

The proposed approaches are decentralised in that a “consensus service” at each controlled junction uses local information on detected incidents (see Section 3.3). This service is managed by a “consensus controller” which is able to communicate with adjacent counterparts about detected incidents. A probability p is always included with a validation result: $(false, p)$ for a false positive and $(true, p, pos, type)$ otherwise. The different algorithms take a possible “associated incident” into account. Figure 4 illustrates such a concurrent incident, where the consensus service in 1 has been notified about a closure of section A towards 2. Simultaneously, a closure of the downstream section B originating at 2 will be an as-

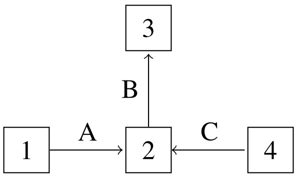

Figure 4: Example of an associated incident. If there is an incident in section A, an incident in B would be an associated incident, an incident in C would not.

sociated incident. The first two algorithms mainly work with thresholds, while the other two employ case distinctions and thresholds. Also, not all algorithms work at all knowledge levels.

The thresholds are chosen with regard to the topology of the road network. In the case of the Manhattan grid with 4 neighbouring junctions for each consensus controller, the thresholds and case distinctions are based on assumptions: neighbouring incidents can be used to confirm locally reported incidents. The thresholds are then chosen as estimates which can be optimised, e.g. using a grid search or machine learning techniques, such as reinforcement learning.

4.1 Algorithm 1

The first basic algorithm works only for KL 1. For this reason, it only decides whether the reported incident I_0 with its confidence c_0 is regarded a true or false positive with a certain probability. Possibly, the directly adjacent services are queried for any associated incidents and respective confidences. This low-complexity approach uses the thresholds $\theta_a = 0.8$ and $\theta_2 = 0.65$ to reach a consensus in two steps.

1. If $c_0 > \theta_a$ the confidence is high enough on its own. It is used as the resulting probability for the direct successfully validated incident $(true, c_0)$.
2. Otherwise, the consensus service asks all its adjacent neighbours for incidents which are combined as I and their confidences C. Two cases can occur:
   (a) At least one neighbouring incident has a confidence higher than θ_a and the highest one, c_{max}, is used to calculate the probability of the successfully validated incident: $(true, \frac{c_0+c_{max}}{2})$
   (b) Otherwise, all confidences $c_i < \theta_a$ are combined as $p_c = \frac{1}{|I|}\sum_i c_i$ and the result is either $(true, p_c)$ if $p_c > \theta_b$ or $(true, 1-p_c)$.

4.2 Algorithm 2

This extension of the Algorithm 1 can validate an incident at KL 3 by collecting all associated incidents as set I and draw conclusions based on the confidence

Require: incident i to be validated
Require: $I =$ set of N received incidents
Require: associated incident confidences c
if $\exists j \in I | j_{pos} == i_{pos}$ **then**
  $p_a = \frac{c_i + c_j}{2}$
  if $p_a > \theta_a$ **then**
    return $(true, p_a, pos, type)$
  else
    return $(false, 1 - p_a)$
  end if
else
  $p = \frac{1}{N}\sum_I c$
  if $N = i \wedge p < \theta_i$ **then**
    return $(true, p, pos, type)$
  else
    return $(false, 1 - p)$
  end if
end if

Figure 5: Pseudocode for Algorithm 2.

values. Based on the number of incidents $N = |I|$, the thresholds are: $\Theta = [\theta_a = 0.8, \theta_0 = \theta_1, \ldots, = \theta_N = 0.65]$ For Manhattan-like networks (Fig. 2) with 4 neighbouring junctions, this results in 6 thresholds. Figure 5 outlines the concept: If a neighbour reports sufficient incidence confidence compared to θ_a, a local incident is validated. Otherwise, the combined confidence of all N neighbouring incidents is compared to the respective threshold θ_n.

4.3 Algorithm 3

The third algorithm can classify an incident at knowledge levels 1 and 2. The basic idea of this (as well as the following) algorithm is to create a conclusion, based on all possible responses from the neighbouring controllers. The potential number of responses is very high, but it is possible to limit these to incidents (a) within a junction on a turning to or from the connected section or (b) on any incoming or outgoing (including the connected) section.

This reduction is possible because the consequences for the algorithm described below are always the same. Moreover, only associated incidents are taken into account, causing a large portion of responses to drop out. Three different positions (see Fig. 6) relative to the local controller are considered when drawing “conclusions”: the section between the local and the responding controller, in a turning in the associated intersections, and in one of the sections which is not between the local and the responding controller.

Table 1: Weighting of the conclusions used in Algorithm 3.

Conclusion	Effect e
Confirmation	1
Weak Confirmation	0.5
No Consequence	0
Possible defect	0

Table 2: Results for Algorithm 3based on the reported confidence c and the cumulative effect E.Note that *type* is "unknown" here, as this algorithm does not work at KL 3.

E	Validation Result
0	$(false, 1-c)$
$]0,1]$	$(true, \min(1, c \times 1.25), pos, type)$
>1	$(true, \min(1, c \times 1.5), pos, type)$

Conclusions and Procedure

The potential conclusions drawn from these cases and their effects are listed in Table 1. For now, "Possible Defect" is handled as "No Consequence", but will be investigated as part of future work. Again, the consensus controller asks its direct neighbours about possible incidents. The number of answers N is at most 4. Based on the incident positions, the effects e_n of these conclusions are summarised as effect $E = \sum_N e_n$. This can be at most $\theta_{max} = 4$ due to the maximum number of junction controllers. Using E and the case distinction in Table 2, the final validation is determined.

Positions

In the case of position 1, the incident is located on the incoming or outgoing section between the local and the responding consensus controller 2. Table 3 outlines the respective conclusions.

Position 2 is an incident within the junction. This corresponds to a turning closure (see Figs. 6c and 6d). For both directions, the answer from controller 2 is the same: *An incident on section A detected by 2 always yields "Weak Confirmation"*.

The position 3 corresponds to an incident on one of the sections that are not between the local and the responding consensus controller as with position 1. Again, the incident can be on the incoming (Fig. 6e) section A or on the outgoing (Fig. 6f) section A. *In both cases, the reaction of controller 2 is negligible, yielding the conclusion "No Consequence"*.

4.4 Algorithm 4

This extension of Algorithm 3 can also validate incidents at KL 3 and take the incident type into account. This multiplies the possible consequences. The conclusions are drawn from reported incidents of neighbouring controllers (the weighting effect in Table 4).

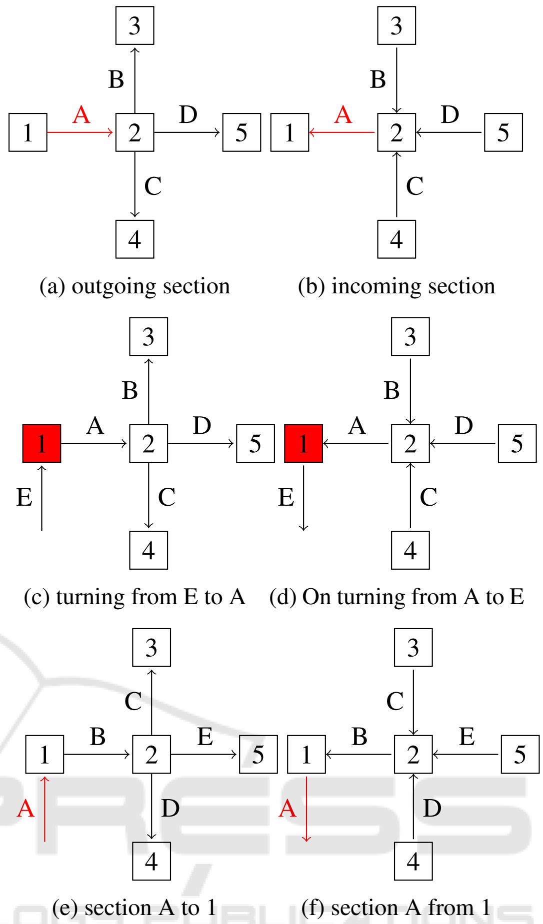

Figure 6: Three incident positions reported by 1 in relation to questioned neighbour 2: an (1) incident on the outgoing or the incoming section between both controllers, a (2) closure of the turning between E and A in either direction and a (3) incident which is *not* the section in case (1).

Table 3: Conclusions for Algorithm 3 at position 1 with an reported incident at consensus controller 1. It can be either on the incoming (Fig. 6a) or the outgoing section A (Fig. 6b). The conclusions are based on the answer from controller 2 – for instance, a turning closure (TC) or incidents of unknown type on section A.

Location	Answer from 2	Conclusion
incoming	Incident on outgoing section	Weak Confirmation
outgoing	Incident on incoming section	
	TC from A to an outgoing section	
incoming	TC from an outgoing section to A	Confirmation
both	Incident on section A	
both	Nothing	Possible defect at 1

Table 4: Weighting of the conclusions used in Algorithm 4.

Conclusion	Effect e
Confirmation	1
Weak Confirmation	0.5
No Consequence	0
Contradiction	-1.0
Possible defect	0

Table 5: Resulting output of Algorithm 4 based on the reported confidence and on the cumulative effect E of the conclusions which are also considered the incident type.

E	Validation Result
< 0	$(false, 1)$
$= 0$	$(true, c, pos, type)$
$= 0.5$	$(true, c \times 1.25, pos, type)$
$= 1$	$(true, \min(1, c \times 1.5), pos, type)$
$]1, 2]$	$(true, \min(1, c \times 1.6), pos, type)$
$]2, 3]$	$(true, \min(1, c \times 1.7), pos, type)$
> 3	$(true, \min(1, c \times 1.8), pos, type)$

A new conclusion "Contradiction" can occur. Again, a combined effect $E = \sum_N e_n$ of the consequences is used for the final validation in Table 5.

The conclusions are based on the incident position reported in relation with the consensus controller being questioned (see Fig. 6). For the first case, Tables 6 and 7 show the conclusions for an incident (incoming or outgoing section). For the second case of a turning in the local controller, the possible conclusions are outlined in Table 8. Lastly, the incident can be in the sections which are not between consensus controllers 1 and 2 (Figs. 6e and 6f). Table 9 shows the possible conclusions.

5 EVALUATION

To evaluate the basic substantially of the validation approaches outlined in Section 4, the functionality of the consensus controller was implemented as part of the OTC system. It was then evaluated using an "emulation service" which executed all algorithms on artificially created test and validation cases of detected incidents. In particular, we introduce artificial incidents in OTC/Aimsun and use the knowledge regarding the type, characteristics, etc. to assess the correctness and success of the validation algorithms.

Table 6: Conclusions for Algorithm 4 at position 1 (see Figs. 6a and 6b) The incident abbreviations correspond to the incidents described in Section 3.2.

Incident	Answer	Conclusion
SC, LC, PLC	TC in 2 to outgoing section	Confirmation
	SC on outgoing section	
	LC on A	
SC	LC, PLC on outgoing section	Weak Confirmation
	LC, PLC on A	
LC	SC, PLC on A or Nothing	Possible defect at 2
PLC	PLC on A	
	SC on A or Nothing	

Table 7: Conclusions for Algorithm 4 at position 1 (see Fig. 6a). An incident is reported by consensus controller 1 on an *outgoing* section towards controller 2.

Incident	Answer	Conclusion
SC, LC	SC on A	Confirmation
	LC, PLC on outgoing section	
	TC in 2 to outgoing section	
PLC	SC, LC, PLC on A	
PLC	TC in 2 to outgoing section	Weak Confirmation
	LC, PLC on outgoing section	
LC	PLC on A	
SC	LC, PLC on A	Contradiction
LC	SC on A	
SC, LC	Nothing	Possible defect at 1

Table 8: Conclusions for Algorithm 4 at position 2 (see Fig. 6b). A turn closure towards or from section A is found at consensus controller 1 .

Direction	Answer	Conclusion
from	SC, LC, PLC on A	Weak Confirmation
towards	LC, PLC on A	Confirmation
	SC on A	Contradiction

Table 9: Conclusions for Algorithm 4 at position 3 (see Figs. 6e and 6f). An incident is reported by consensus controller 1 on an incoming or outgoing section.

Incident	Location	Answer	Conclusion
SC, LC	incoming	LC, PLC on B	Confirmation
LC	outgoing	TC to B	
PLC	incoming	LC, PLC on B	Weak Confirmation
LC, PLC	outgoing	SC, LC, PLC	
PLC	outgoing	TC to B	

5.1 Evaluation Data

The artificial evaluation data are created with two goals in mind: The system should correctly detect false positives as well as all cases from the case distinctions of Algorithm 4 (Section 4.4). Six sets of data (see Table 10) were created: Two sets are based on the

Table 10: The 6 evaluation datasets with a short description and the number of test cases. Sets 4 to 6 represent the “assumed” normal or randomly varying confidences. Section 5.2 describes the sets in more detail.

Set	Content	Tests
1	No responses from neighbouring services	140
2	Responses from second level neighbouring services	420
3	Normal confidence values	80
4	Confidence values varying by 10%	80
5	Confidence values varying by 15%	80
6	Confidence values varying by 20%	80

Table 11: Exemplary datasets for occurring incidents: 2 sections closures which are to be (a) validated or (b) not, as well as a turning closure (c) which should not be validated.

	(a)	(b)	(c)
type	SC	SC	TC
time	1	2	3
location	$3;3>4$	$3;3>4$	$2;5\gg 1;4$
confidence	1	0.1	1
controller	3;4	3;4	2;4
	Expected Output		
decision	true	false	true
location	$3;3>4$		$2;5\gg 1;4$
type	SC		TC

false positives, and four sets with varying confidence values cover the assumptions of Algorithm 4. The confidence values were chosen under the assumption that the detection of a complete closure is straightforward: section closures (0.9), lane closures (0.6), partial lane closures (0.3), and turn closures (0.8).

The sets were created using the emulation service and the 7x7 Manhattan grid (Fig. 2) as a basis. Table 11 shows some example evaluation cases for the data of the incident. Depending on the knowledge level, the “expected location” or “expected type” is redundant.

5.2 Test Sets

For every case in Section 4 where responses from neighbours are expected, each test set contains at least one corresponding dataset. This response covers the incident type, time, location and confidence, as well as other aspects of the algorithms:

1. 20 junctions were randomly selected and three incidents were generated for each. An incoming and an outgoing section are randomly selected and a section closure, a lane closure and a partial lane closure are created respectively. A closure is generated for a randomly selected turning movement.
2. Again, 20 junctions were selected including second-level neighbours for each. Each selected neighbour responds to either a section closure, a lane closure, or a partial lane closure incident. For each, a corresponding (from the point of view of the neighbour) incident is generated.
3. Based on the case distinctions of Algorithm 4 (Section 4.4), occurring incidents and corresponding responses were created.
4. This set is similar to the one above, with the confidence values randomly varied by $\pm$ 10%.
5. This set is similar to the one above, with the confidence values randomly varied by $\pm$ 15%.
6. This set is similar to the one above, with the confidence values randomly varied by $\pm$ 20%.

Table 12: Results for running each algorithm on all 6 test sets. The columns of the far right list the correctly validated cases, when the Algorithm 1 to 4 was applied.

Set	Cases	1	2	3	4
1	140	100	40	119	0
2	280	288	108	351	0
3	80	42	45	32	80
4	80	43	47	37	80
5	80	42	48	39	80
6	80	42	46	30	80

5.3 Results and Discussion

The results in Table 12 show the unpredictable behaviour of Algorithms 1, 2 and 3, when used on datasets which are based on case distinctions: The confidence fluctuation in the last four sets should affect the predictability of the outcome. For example, the confidence of set 4 for a section closure could drop to 70%, yet the expected result is still a validated incident (while it is more likely a false positive). Still, Algorithm 4 reaches the expected 100% for sets 3 to 6, while the others validate 30 to 50 cases per set.

As the tests are artificially created and the thresholds are just estimated, the validity of this evaluation is not high. Also, the conclusions are based on assumptions regarding the simulation behaviour. However, the results demonstrate the potential to work: All four algorithms provide outcomes characterised by the different knowledge levels. Furthermore, a perfect classification might not be achievable in real-world environments, due to the trade-off between reaction or classification time and accuracy. This is especially due to the fact that the longer a traffic pattern is observed, the less is the impact of short-time fluctuations. With this in mind, the current approach aims at an as-fast-as-possible reaction. It is a basis for subsequent consideration in the traffic control strategies.

6 CONCLUSIONS

This work outlined a generic approach for a collaborative validation of locally detected incidents in simple urban road networks. The network topology is taken into account as well varying degrees of knowledge about the detected incidents. Integrated into the OTC, the approaches showed potential to improve the accuracy and resulting success of local incident detection. As a limitation, the evaluation is based only on assumptions about the underlying AID.

This is a first attempt at an improved detection in OTC and next steps can be outlined: An evaluation with real (simulated) traffic and incident situations. Also, the AID must be carried out by the OTC as outlined in Section 2. Finally, the various threshold must be optimised, e. g., by using machine learning techniques. All this will multiply the test cases and possible conclusions dramatically which make a complete test scenario unfeasible. Finally, the reinforcement learning capabilities of the OTC should be applied, to ensure the SASO capabilities of the system.

ACKNOWLEDGEMENTS

This research was supported by the Deutsche Forschungsgemeinschaft, DFG, in the context of the project “Zwischenfall-bewusstes resilientes Verkehrsmanagement für urbane Straßennetze (InTURN)” under grant TO 843/5-1. We acknowledge this support.

REFERENCES

Ahmed, S. and Cook, A. (1980). Time series models for freeway incident detection. *Transp. Eng. J of the Am. Soc. of Civ. Eng.*, 106(6):731–745.

Aimsun (2021). *Aimsun Next 20 User’s Manual*, Aimsunm Next 20.0.3 edition. Accessed on: October, 22, 2021.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *AAAI Press*, pages 226—-231.

Gokulan, B. and Srinivasan, D. (2010). Distributed geometric fuzzy multiagent urban traffic signal control. *IEEE Trans. on Int. Transportation Sys.*, 11(3):714–727.

Jenelius, E. and Koutsopoulos, H. (2013). Travel time estimation for urban road networks using low frequency probe vehicle data. *Transp. Res. Part B: Methodological*, 53:64–81.

Lin, W. and Daganzo, C. (1997). A simple detection scheme for delay-inducing freeway incidents. *Transp. Res. Part A: Policy and Practice*, 31(2):141–155.

Mauro, V. and Taranto, C. D. (1990). Utopia. *Control, computers, communications in transportation*.

Müller-Schloer, C. and Tomforde, S. (2017). *Organic Computing-Technical Systems for Survival in the Real World*. Springer.

Oliveira, L. D. and Camponogara, E. (2010). Multi-agent model predictive control of signaling split in urban traffic networks. *Transp. Res. Part C: Emerging Tech.*, 18(1):120–139.

Payne, H. and Tignor, S. (1978). Freeway incident-detection algorithms based on decision trees with states. *Transportation Research Record*.

Payne, H. J. (1975). Freeway incident detection based upon pattern classification. In *Proc. of IEEE Conf. on Decision and Control*, volume 14, pages 688–692. IEEE.

Prothmann, H., Branke, J., Schmeck, H., Tomforde, S., Rochner, F., Hähner, J., and Müller-Schloer, C. (2009). Organic traffic light control for urban road networks. *Int. J. Auton. Adapt. Commun. Syst.*, 2(3):203–225.

Robertson, D. and Bretherton, D. (1991). Optimizing networks of traffic signals in real time – the SCOOT method. *IEEE Trans. on Veh. Tech.*, 40(1):11–15.

Sims, A. and Dobinson, K. (1980). The Sydney coordinated adaptive traffic (SCAT) system – Philosophy and benefits. *IEEE Trans. on Veh. Tech.*, 29(2):130–137.

Sommer, M., Tomforde, S., and Hähner, J. (2016). Forecast-augmented route guidance in urban traffic networks based on infrastructure observations. In *Proceedings of the International Conference on Vehicle Technology and Intelligent Transport Systems, VEHITS 2016, Rome, Italy, April 23-24, 2016*, pages 177–186.

Stephanedes, Y. and Chassiakos, A. (1993). Freeway incident detection through filtering. *Transp. Res. Part C: Emerging Technologies*, 1(3):219–233.

Thomsen, I., Zapfe, Y., and Tomforde, S. (2021a). Urban traffic incident detection for organic traffic control: A density-based clustering approach. In *Vehits*, pages 152–160.

Thomsen, I., Zapfe, Y., and Tomforde, S. (2021b). Urban traffic incident detection for organic traffic control: A density-based clustering approach. In *Proceedings of the 7th International Conference on Vehicle Technology and Intelligent Transport Systems, VEHITS 2021, Online Streaming, April 28-30, 2021*, pages 152–160.

Tomforde, S., Prothmann, H., Branke, J., Hähner, J., Mnif, M., Müller-Schloer, C., Richter, U., and Schmeck, H. (2011). Observation and control of organic systems. In *Organic Computing—A Paradigm Shift for Complex Systems*, pages 325–338. Springer.

Tomforde, S. and Thomsen, I. (2022). A concept for collaborative incident validation in a self-organised traffic management system. In *Proceedings of the 8th International Conference on Vehicle Technology and Intelligent Transport Systems, VEHITS 2022, Online Streaming, April 27-29, 2022*, pages 316–323.

Vincent, R., Peirce, J., and Webb, P. (1990). Mova traffic control manual. *MOVA reports*.

Wilson, S. W. (1995). Classifier Fitness Based on Accuracy. *Evolutionary Computation*, 3(2):149–175.