# Urban Traffic Incident Detection for Organic Traffic Control: A Density-based Clustering Approach

Ingo Thomsen, Yannick Zapfe and Sven Tomforde
*Intelligent Systems, Christian-Albrechts-Universität zu Kiel, 24118 Kiel, Germany*

Keywords: Organic Traffic Control, Traffic Flow Analysis, Traffic Incident Detection, Traffic Management.

Abstract: The traffic demands in urban road networks can fluctuate immensely. The Organic Traffic Control (OTC) offers a resilient traffic management to control such traffic demands. An additional challenge is the detection of unforeseen traffic incidents. To enhance the capabilities of OTC accordingly, we outline a traffic incident algorithm based on DBSCAN, a density-based clustering algorithm: In a simulated urban road network, equipped with traffic light controllers at intersections, vehicle detectors are used to gather traffic flow data. The clustering of this time series data to detect simulated road blockages is expanded using various filters. This extension of the initial clustering is the result of an manual evaluation process, which shows the principal applicability of this approach.

## 1 INTRODUCTION

Increasing mobility and consequential rising traffic demands elevate traffic density on the streets, especially in urban areas. Therefore, the average travel times for individual traffic participants can rise. As traffic situations are in constant change at a spatial and temporal level, traffic management solutions require mechanisms for adaptation at runtime.

The Organic Traffic Control (OTC) system (Sommer et al., 2016) is self-organised traffic management system. Each intersection in a network is controlled by an instance of the OTC, which decides locally for each node if and how the traffic light signalisation is to be adapted. For this, the traffic conditions are constantly analysed, and the decision behaviour is improved over time. This facilitates resilient traffic management.

In addition to the common fluctuation in traffic demands, various traffic incidents – accidents, road blockades, unscheduled maintenance, or construction work – can have a severe impact on the traffic behaviour. To boost the adaptive capacities of the OTC system in the context of urban road networks, incident detection capabilities are highly beneficial for more efficient and accurate traffic management. This paper explores the possibility to detect incidents in urban road networks by employing a density-based clustering algorithm (DBSCAN) (Ester et al., 1996) for traffic incident detection. An intermediate goal is to equip the OTC with this capability. In contrast to state-of-the-art approaches from highways, the incident detection problem in urban areas is more complex due to a more heterogeneous traffic model (e.g., intersections with traffic lights, more distributed traffic demands, primary and secondary roads) and participant behaviour (e.g., unloading lorries, stopping busses). This can introduce patterns in the detector data that are very similar to those of actual incidents.

The remainder of this paper is organised as follows: The next section provides information on the Organic Traffic Control and traffic incidents detection. Section 3 then describes underlying assumptions about the traffic model under consideration. The next two sections outline the main contribution of this work: In Section 4 the proposed incident detection is presented and Section 5 describes the corresponding evaluation, which includes experimental setup, evaluation, and results. As an outlook, Section 6 proposes the next steps towards an incident-aware self-organised traffic control behaviour based on OTC. Finally, Section 7 summarises the paper.

## 2 BACKGROUND

As a basis for the presented approach, this section introduces the OTC system and briefly summarises the state-of-the-art in incident detection.



Thomsen, I., Zapfe, Y. and Tomforde, S.
Urban Traffic Incident Detection for Organic Traffic Control: A Density-based Clustering Approach.
DOI: 10.5220/0010454101520160
In *Proceedings of the 7th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2021)*, pages 152-160
ISBN: 978-989-758-513-5

## 2.1 Organic Traffic Control

The Organic Traffic Control (OTC) system is a traffic management system, developed according to the principles of Organic Computing (Müller-Schloer and Tomforde, 2017). It is a self-adaptive and self-organising (SASO) system, which distributes the complexity of decision-making processes among autonomous agents, which in turn can cooperate with each other. Such SASO systems tackle the complexity of controlling runtime behaviour by distributing the decision-making processes among autonomous agents, which in turn cooperate with each other: Goals can be reached in a large system without centralised ruling. It consists of a productive part and a control mechanism, based on a multilevel observer/controller architecture by (Tomforde et al., 2011). This architecture is outlined in Figure 1 and consists of 3 levels:

**Level 0.** Here, the the System under Observation and Control (SuOC) is situated – the (simulated) traffic system equipped with sensors (detector) and traffic light controllers (actuator). The signal settings of these controllers can be altered.

**Level 1.** This level contains an observer that processes the sensor data from the SuOC to create a model of the current traffic (flow) situation. This is handed over to the Controller, which is based on a Learning Classifier System (LCS). The controller then selects a fitting rule to alter the traffic light signalisation in the SuOC This is the "online" layer.

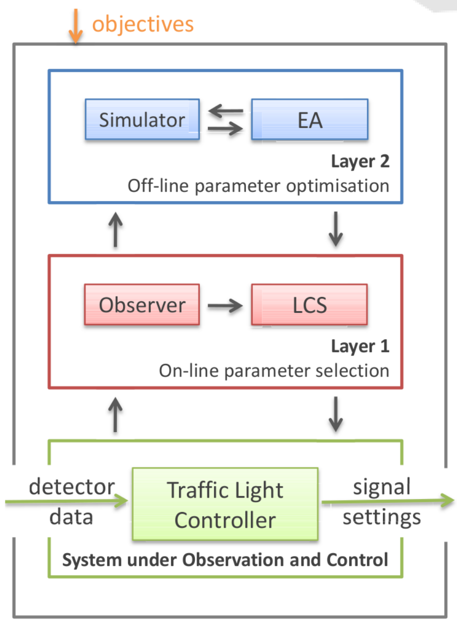



Figure 1: Overview of the multilevel OTC architecture.

**Level 2.** This is the "offline" layer, which is activated, in case no appropriate rules can be found. It uses an evolutionary algorithm to generate new rules for Layer 1. These newly found rules are evaluated using a traffic simulator (Aimsun SLU, 2020).

Each intersection is equipped with such an OTC control module and, through interaction. additional functionality can be provided, e.g., *dynamic route guidance (DRG)* or *progressive signal systems (PPS)*, also known as "green waves" in traffic.

## 2.2 Traffic Incident Detection

Traffic control has to continuously evaluate information about the current road situation, including traffic incidents. The *California Algorithm* (Payne and Tignor, 1978), the incident detection approach implemented in OTC, is a decision tree algorithm based on four states: *incident-free, incident termination, initial detection, and incident continuation*, all based on occupancy values at several locations. This method focuses mainly on highways and is not necessarily equally adequate for urban areas, as argued by (Sommer et al., 2016). For further reading, (Parkany and Xie, 2005) presents an exhaustive list of incident detection methods for highways and arterial roads.

The information on the traffic flow is usually represented as time series data. A typical part of the corresponding preprocessing is the detection of anomalies, alongside the related, but distinct novelty and outlier detection. (Pimentel et al., 2014) describe anomaly detection as recognising that test data differs significantly from training data. They also outline different approaches, based on information theory, domain, reconstruction, probability, and distance. In this context anomalies are localised traffic incidents that deviate considerably from normal traffic behaviour and have a significant influence on the overall network performance. For instance, such incidents could be road works or accidents (see Section 3).

Cluster analysis, the identification of groups of similar points in (time series) data, is a common technique for statistical data analysis, with numerous algorithms based on *connectivity, centroids, grid, distribution, and density*. Such a well density-based clustering algorithm, DBSCAN (Ester et al., 1996), is used as a base for incident detection as described in more detail in Section 4.

# 3 MODEL ASSUMPTIONS

The approach for detecting traffic incidents described in this paper is intended to be used in the context of Organic Traffic Control (see Section 2.1). Therefore, this section gives a summary of the broader underlying domain model and the simulation setup.

**Urban Road Networks.** In contrast to highways or arterial roads, urban road networks do not have a privileged "main road" with a dominating traffic load. Single or multi-lane sections with speed limits connect intersections, which are equipped with one traffic light controller (TLC) each to operate traffic lights at each incoming section. The signalisation phases of the TLCs are displayed periodically based on a control cycle. The incoming and outgoing sections also have detectors for counting vehicles (such as induction loops embedded in the street surface), thus allowing to calculate the traffic flow for each section. The network may include junctions where no roads "intersect" and which are not controlled by a TLC. To facilitate other OTC features (PSS or DRG), all incoming roads of an intersection are equipped with variable message signs (VMS) to relay information to the drivers. Additionally, the junction controllers may communicate with immediate neighbours.

**Traffic Demand.** The occurring traffic load can be defined as an origin-destination (O/D) matrix which defines how many vehicles traverse from each origin to each destination per hour. An O/D matrix may also be time-variant to model changing demands (e.g., during a simulation).

**Incidents.** At this stage, a general incident model is applied. Real-life events and their immediate effect are represented as a full roadblock. They are located within a section with fixed start and end times. Section 6 suggests a prospective, more complex model.

# 4 DETECTION APPROACH

The intention of this work is to evaluate DBSCAN with respect to identifying incidents based on preprocessed traffic flow values. As a result of the evaluation process outlined in Section 5.7, detection improvements (flattening and domain knowledge) were added to this approach and are described in this section.

## 4.1 DBSCAN

*Density Based Spatial Clustering of Applications with Noise* (DBSCAN) (Ester et al., 1996) is an established algorithm for clustering and anomaly detection. It is efficient for large amounts of data and does not require domain knowledge. Also, the number of target clusters does not have to be specified beforehand and the clusters can be of arbitrary shape.

The key idea of DBSCAN is to form clusters of core objects with a minimum neighbourhood density and further objects that are connected to these cores. The neighbourhood density is determined by the number of other objects surrounding a data object within a given radius. Other objects that are not connected with objects belonging to a cluster are considered as noise. Here, the objects are the points of the traffic flow data. With respect to a distance measure *dist* and the two algorithm parameters $\varepsilon$ and *minPts*, the following concepts are used:

**$\varepsilon$-neighbourhood.** This is the set of points $N_\varepsilon$ that is located near a point $p$ within a given radius $\varepsilon$:

$$N_\varepsilon(p) = \{p' \mid dist(p,p') \leq \varepsilon\} \tag{1}$$

**Directly Density-reachable.** All points $p'$ are directly density-reachable from a point $p$ if the following applies:

$$p' \in N_\varepsilon(p) \tag{2}$$
$$|N_\varepsilon(p)| \geq MinPts \tag{3}$$

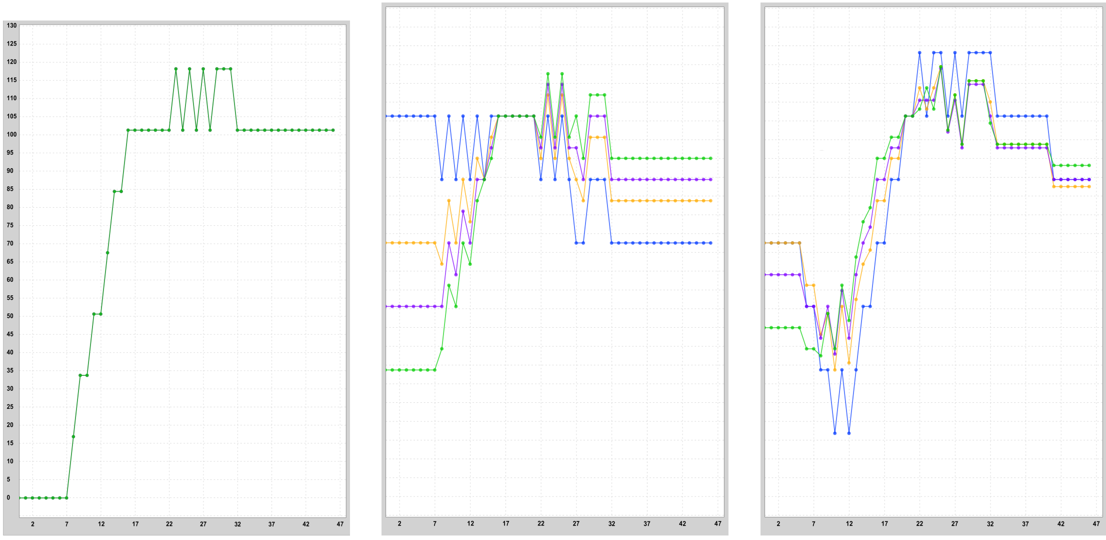
**Density-reachable.** All points $p'$ are density-reachable from a point $p$ if a chain of points $p = p_1, ..., p_n = p'$ exists such that $p_i$ and $p_{i+1}$ are directly density-reachable for each $i \in \{1, ..., n\}$.

**Density-connected.** Two points $p$ and $p'$ are density-connected if a point $q$ exists from which both are density-reachable.

**Cluster.** The set of points $C$ for which applies:

$$\forall p, p' : \text{ if } p' \text{ density-reachable from } p \in C \tag{4}$$
$$\forall p, p' \in C : p \text{ is density-connected to } p' \tag{5}$$

**Noise.** All points that do not belong to any cluster are categorised as Noise.
An important part of DBSCAN is the choice of the distance measurement for determining the $\varepsilon$-neighbourhood. Section 5.5 lists the measures used for this work.



(a) Control cycle starting at 0s. (b) Control cycle starting at 90s. (c) Control cycle starting at 180s.

Figure 2: Traffic flow at intersection labelled "2;1_1;1" (see Figure 3). Flattened and unflattened time series show the flows in *vehicles/hour* – indicated on the y-axis – for the second 45s of each of the first three control cycles at the beginning of a simulation. Blue curve: no flattening; yellow: flattening weight of 66 %; purple: 50 %; green: 33 %.

## 4.2 Flattening

The raw detector values are preprocessed to form time series of flow values – one data point per section and control cycle. In addition, the time series are *flattened* (dampened) using a weighted average. Here, previous traffic flows are taken into account with more recent values having more influence. The weight $w$ is the only parameter, denoting the percentage of a data point in the weighted average. For a flow value time series $x_t^s$ in section $s$ at the starting time $t$ with a control cycle duration $t_c$, the flattening $f$ is calculated recursively:

$$f(x_t^s) = \begin{cases} x_t^s & \text{if } t = 0 \\ x_t^s * \frac{w}{100} + (1 - \frac{w}{100}) * f(x_{t-t_c}^s) & \text{otherwise} \end{cases} \tag{6}$$

For instance, Figure 2 shows the time series of three cycles at the beginning of a simulation in the 2x2 Manhattan grid (see Section 5.1) for several weights: For the first cycle, there is no "previous" data, so the flattened times series are the same. This is then taken into calculation for the second cycle and flattens the time series accordingly. In the last chart, both previous time series are integrated into the weighted average. E.g., for the flattening weight of 50, the current time series is accountable for 50 % of the weighted average; the previous time series for 50 % of the remaining 50, resulting in a 25 % share for the first time series. It shows the intended effect of flattening the curve: Smaller and fewer spikes.

## 4.3 Domain Knowledge

The integration of knowledge about the model can help to improve the detection. It can be applied for initial detection (to increase the detection rate) or validation (to reduce the false positives). During the evaluation process described in Section 5.7, several observations about the traffic networks, demands, and incidents were utilised.

# 5 EVALUATION

To assess the performance of the incident detection conceived in Section 4, the algorithm was executed using varying parameter sets on traffic flow data obtained from traffic simulations. This section describes the experimental setup with respect to the model from Section 3 followed by the initial algorithm parameters and a description of the evaluation process.

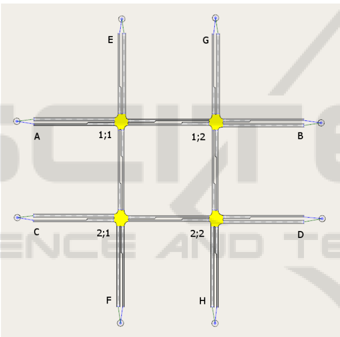
## 5.1 Road Networks

The traffic simulation was based on two regular, fully connected Manhattan grids as the initial urban road networks (see Figure 3). Only cars as vehicles were considered. All sections had a length 150m – roughly the section length of Barcelona's chequered inner-city road network – with one lane plus additional side lanes for left-turns.

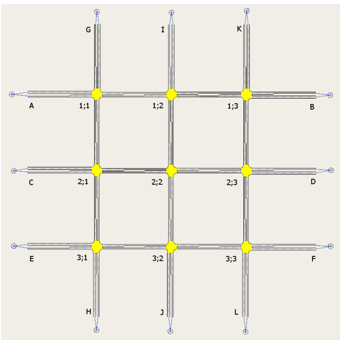
Following experiments of earlier work by (Tomforde et al., 2008), each intersection was controlled by an identical fixed-time controller (FTC) using 4 signal phases (with a transition time of 5s in between) and a control cycle time of 90s. In the phase I, cars on horizontal sections (or leaving them by turning right) are served for 25s. The remaining cars turning left are served in the phase II for 10s. The vertical traffic is handled analogously in phases III and IV.

## 5.2 Traffic Demand

The traffic demands in Table 1 were also adopted from (Tomforde et al., 2008), and can be categorised in primary, secondary, and tertiary demands along with the "straight" O/D connections. For the 2x2 grid, the most heavily used O/D pairs are A to B and D to C, with decreasing demand in their opposite directions and the vertical straight connections. In the case of the 3x3 grid, the demands as in Table 2 were similar: The horizontal directions carry the heavier traffic load and direction A to B the heaviest.




(a) 2x2 Manhattan grid




(b) 3x3 Manhattan grid

Figure 3: Manhattan network with 4-phased FTCs.

The described scenario had a total duration of 1:15 hours with the first 15 minutes being the warm-up phase, which was not considered by the incident detection algorithm.

Table 1: Traffic demands in *veh/h* for 2x2 Manhattan grid.

| O/D | demand | opposite direction |
|---|---|---|
| A – B | 400 | 200 |
| C – D | 200 | 400 |
| E – F | 150 | 150 |
| G – H | 150 | 150 |
| others | 10 | 10 |

Table 2: Traffic demands in *veh/h* for 3x3 Manhattan grid.

| O/D | demand | opposite direction |
|---|---|---|
| A – B | 400 | 200 |
| C – D | 300 | 150 |
| E – F | 300 | 150 |
| G – H | 150 | 150 |
| I – J | 150 | 150 |
| K – L | 150 | 150 |
| other | 10 | 10 |

## 5.3 Incidents

All incidents start at 00:45 and last half an hour until the simulation end. This is the second half of the simulation phase after the warm-up. In addition to the incident-free simulation run, several scenarios with one incident were created. Depending on the incident placement along the varying traffic demands described above, these scenarios fall into three categories. Due to symmetries of the topology and demands, one incident each was chosen from each category.

## 5.4 Simulation

For the simulation of road networks, incidents, and traffic demands (and the actual routing of the cars), a commercial traffic modelling and simulation software was used: AIMSUN Next (Aimsun SLU, 2020) – an obvious choice, as it is used for OTC as well (see Section 2.1). A client was programmed that used the simulator's API to extract vehicle counts from a running experiment. In this client-server setup, the incident detection, as well as the generation of plots and log files for the evaluation were implemented as part of the server.

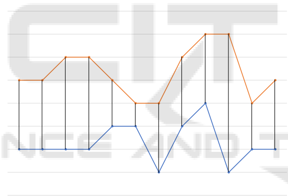
## 5.5 Distance Measures

The applied distance measures for DBSCAN are briefly explained below. A point of the input data set consists of a time series of flow values for a certain section over the course of a single control cycle – one measurement per second. DBSCAN is executed once per section and only time series of the same section are compared using the distance measure.

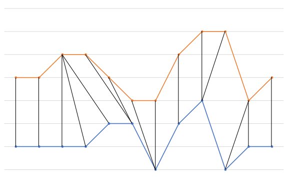
**Euclidean Distance.** The simple length of a straight line between two points in Euclidean space is widely used and as the time series have the same resolution, the exact same points in time are compared.

**Dynamic Time Warping (DTW).** This similarity measure by (Berndt and Clifford, 1994) searches a so-called warping path for two time series: A continuous and monotonous sequence of pairs of points from both series. This path has to cover both time series completely. The cumulative distance between the data points is calculated based on another measure. In this work, Euclidean distance is used and the resulting DTW distance is normalised by the path length.

Figure 4 depicts an artificial example of both distance measures: While the Euclidean distance only matches points at the same time, points from one series can be matched with multiple points from the other series in DTW.



(a) Euclidean



(b) Dynamic Time Warping

Figure 4: Artificial example for applying distance measures between the two time series – red and blue – which are connected by multiple lines that show the different data point matching for DTW (b) and Euclidean distance (a).

**Polynomial Approximation.** Two time series are approximated by polynomial functions of a certain degree – linear and cubic in this work. Then the Euclidean distance between the two lists of polynomial coefficients is calculated.

**Average Distance.** The absolute difference between the average values of two time series.

**Relative Distance.** This is a derived measure: One of the above measures is used to calculate a distance which is then normalised using the maximum of the two time series averages. Therefore, the distance is relative to the larger of two average flow values.

## 5.6 Criteria

To evaluate the choice and combinations of detection algorithm parameters, these three criteria of decreasing priority were considered:

1. Detection Rate: The most important criterion to be maximised is similar to the true positive rate in binary classification tasks. Here, a correct detection of an incident includes discerning the section where it occurs. Obviously, the incident should only be detected after the incident has actually started. As each scenario contains either one or no incident, the detection rate is the ratio of correctly identified and expected incidents for all scenarios in a model.
2. False Alarms: Number of identified incidents that do not correspond to a simulated incident (either non-existent or located in a different section). This criterion should be minimal. It is prioritised lower than the detection rate: Some false alarms may be tolerated to achieve better detection.
3. Detection Delay: This has to be considered when applying flattening (see Section 4.2), as incidents then have to persist multiple control cycles to become significant for the distance measure. Obviously, smaller time delays are preferable.

## 5.7 Evaluation Process

The concrete evaluation leading to the results in Section 5.8 was an iterative process which comprises of several phases: Initially, a rather generic set of parameter combinations was used to prove the general applicability. This was followed by a manual optimisation of the algorithm parameters. Dependencies were identified along with parameters that have little to no impact, which can then be locked at a certain value. Equally, a parameter can already have an obvious optimal value that outperforms others, so it can be locked at that optimum as well. These values were the base for finding a good combination through manual exploration based on the 2x2 Manhattan grid, which were then evaluated for the 3x3 grid as well. The goal was to find a parameter set that is optimal according to evaluation criteria above for both models.

**Initial Parameter Combinations.** In their original description of DBSCAN, (Ester et al., 1996) state that *minPts* does not have a significant influence and can be set to 4 as default. This was done for the initial validation. To find candidates for the more important ε parameter, k-distance plots for runs with and without an incident were inspected. Depending on distance measure and flattening, the ε values were chosen. For instance, the values 500, 400, 300, 200, and 100 were selected for “Euclidean distance without flattening”. For flattening, the weights 33, 50, 66 and 0 were selected. Together with 5 distance measures (all except the relative distance), this resulted in 100 initial parameter combinations.

Only moderate detection rates were achieved, coupled with high numbers of false alarms. This indicated the need for adjustments of the incident detection.

**Incident Validation.** To improve the discrimination of traffic fluctuation against real incidents, domain knowledge was applied: While decreased traffic is expected in the “incident section” and its downstream sections, the load should increase upstream as well as in “detour sections”. A validation mechanism based on this was devised, which decreased the number of false alarms significantly, while the detection rates were conserved.

**Finding Incident Indicators.** To identify the reasons for weaker detection rates, the correctly detected incidents were analysed in more detail. These were mostly associated with high traffic demand. To better detect secondary and tertiary incidents, the relative distance measure was introduced. After testing 10.000 parameter combinations, it showed to be mainly suitable for the incident indicators rather than for validators. From there on, the relative distance was only used for indicators, while the other measures were only applied for validators.

**Detection Delay.** A deeper look into the detection delay showed, that heavier flattening leads to longer delay: All optimal combinations with regard to detection rate and false alarms included flattening, which led to significant delays of at least two control cycle periods.

**Incident Filters.** Incidents with certain properties common to false alarms were identified. One finding was that most of them were situated on the edge of the road model, possibly due to the way AIMSUN Next simulates the incoming traffic. These were filtered out.

Additionally, false alarms were discovered almost exclusively in scenarios that also included incidents. It showed that some validators confirmed too many other incidents, which resulted in false alarms. Therefore, they were restricted to validate at most one incident.

**minPts.** Prior to the last optimisation step, the so-far locked parameter *minPts* was investigated once more. The values 2, 3, 4, 5, 6 and 8 were tested for all sets of distance measures, ε values and flattening weights as well as for filters, indicator and validator instances: Especially good combinations of distance measures and ε work equally well for all tested *minPts* values, but as no significant changes occurred, $minPts = 4$ was retained.

**Parameter Optimisation.** Finally, the optimal results for the 2x2 grid were evaluated in the 3x3 grid in order to optimise the parameters for unflattened data. The optimal parameter sets according to detection rate and false alarm count are chosen from those combinations that achieved an average detection delay of less than two control cycle durations (180 s).

### 5.8 Results

Table 3 presents the results for the 2x2 Manhattan grid. It shows that the incident filter reduces the number of false alarms for all distance measures. Although the results have been optimal for some combinations even before introducing the filter, more combinations of different parametrisation now show nearly optimal results. An optimal result was achieved parametrisation when using average distance for validation and relative distance for indication.

The Table 4 shows the best ε values for each combination of indicator and validator distance measure with unflattened data in the 3x3 Manhattan grid. The results indicate that in the case of unflattened data no dependency between the indicator and validator parameter sets exists. For all validator distance measures, the same ε values work best for both relative distance measures. In return, the relative average distance with ε set to 0.95 works best for all validator distance measures and especially outperforms the relative DTW with respect to false alarms. The validator distance measures do not differ significantly in their results, but the Euclidean distance measure with ε set

Table 3: Best parametrisations after introducing the incident filter from initial parameter combinations in a 2x2 Manhattan grid.

| Validation | | Indication | | | |
|---|---|---|---|---|---|
| distance measure | ε | Relative distance measure | ε | detection rate | false alarms |
| Euclidean | 200 | Average | 0.8 | 1 | 0 |
| DTW | 12 | Average | 0.8 | 1 | 0 |
| Linear | 20 | Average | 0.8 | 1 | 0 |
| Cubic | 24 | Average | 0.8 | 1 | 0 |
| Average | 8 | Average | 1 | 1 | 0 |
| Euclidean | 200 | DTW | 0.8 | 1 | 0 |
| DTW | 18 | DTW | 0.8 | 1 | 0 |
| Linear | 20 | DTW | 0.8 | 1 | 1 |
| Cubic | 12 | DTW | 0.8 | 1 | 1 |
| Average | 16 | DTW | 0.8 | 1 | 0 |

to 150 and the cubic approximation distance with 9 as ε achieve slightly better results than the other measures.

Table 4: Best parametrisations after introducing the incident filter from promising parameter combinations in a 3x3 Manhattan grid.

| Validation | | Indication | | | |
|---|---|---|---|---|---|
| distance measure | ε | Relative distance measure | ε | detection rate | false alarms |
| Euclidean | 150 | Average | 0.95 | 1 | 4 |
| DTW | 9 | Average | 0.95 | 1 | 7 |
| Linear | 10 | Average | 0.95 | 1 | 7 |
| Cubic | 9 | Average | 0.95 | 1 | 5 |
| Average | 8 | Average | 0.95 | 1 | 7 |
| Euclidean | 150 | DTW | 0.8 | 1 | 20 |
| DTW | 9 | DTW | 0.8 | 1 | 24 |
| Linear | 10 | DTW | 0.8 | 1 | 24 |
| Cubic | 9 | DTW | 0.8 | 1 | 23 |
| Average | 8 | DTW | 0.8 | 1 | 26 |

# 6 FUTURE WORK

The evaluation process in Section 5 showed that, in principle, DBSCAN combined with some enhancements can be employed for incident detection in simple urban road networks. Several assumptions are made in the context of the experiments, and future research could address the consequent limitations.

**Model Expansion.** More realistic – and complex – road networks and traffic demands have to be addressed, e.g., real-life networks with challenging traffic loads like during rush-hour. Also, the underlying incident model can be expanded dramatically. Instead of a "binary" roadblock, real-life incidents can have varying effects with respect to spatial (location, movability, extend, ...) or temporal characteristics (duration, regularity, ...). Finally, an incident may occur isolated: Different types of traffic incidents may appear simultaneously or with relatively small delay.

**Algorithm Enhancements.** The presented incident detection can be extended: Additional distance measures or even a different clustering algorithm are conceivable, e.g., the *Local Outlier Factor (LOF)* by (Breunig et al., 2000), which shares similarities with DBSCAN. Also, an expansion of the model might provide domain knowledge, which could be beneficial for incident indication and validation.

**Parameter Optimisation.** The evaluation in Section 5.7 for a restricted selection of algorithm parameters already generated thousands of test runs. Together with the extensions proposed above, a manual inspection of parameters sets is no longer feasible. Hyperparameter optimisation from machine learning could provide suitable techniques, for instance, *grid search, evolutionary algorithms, Bayesian or gradient-based optimisation*

**Integration with OTC.** The structured exploration of parameters on potentially much more complex models calls for an automated system for setting up and conducting experiments as well as the subsequent evaluation. Consequently, the integration into the Organic Traffic Control could improve the self-adaptive and self-organising capabilities of the OTC.

# 7 CONCLUSION

In this work, a traffic incident detection approach in the context of urban road networks is presented, which employs the density-based clustering algorithm DBSCAN together with some enhancements. These include incident validation using surrounding sections, a combination of different algorithm parameter sets for the detection and validation of incidents, and filtering out incidents with characteristics common to false alarms. These enhancements are results of the evaluation process, which eventually show the principal applicability of the proposed approach.

The experimental setup is limited compared to real-life networks, traffic demands and incidents. Together with the numerous and often manual evaluation steps, further research steps can be identified, which were also presented as future work.

## REFERENCES

Aimsun SLU (2020). *Aimsun Next Professional, Version 20*. Barcelona, Spain.

Berndt, D. J. and Clifford, J., editors (1994). *Using dynamic time warping to find patterns in time series*, volume 10. Seattle, WA, USA.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof. In Dunham, M., Naughton, J. F., Chen, W., and Koudas, N., editors, *Proceedings of the 2000 ACM SIGMOD international conference on Management of data - SIGMOD '00*, pages 93–104, New York, New York, USA. ACM Press.

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

Müller-Schloer, C. and Tomforde, S. (2017). *Organic Computing – Technical Systems for Survival in the Real World*. Autonomic Systems. Birkhäuser. ISBN: 978-3-319-68476-5.

Parkany, E. and Xie, C. (2005). A complete review of incident detection algorithms & their deployment: What works and what doesn't.

Payne, H. J. and Tignor, S. C. (1978). Freeway incident-detection algorithms based on decision trees with states. In *Urban system operation and freeways*, Transportation research record. National Academy of Sciences, Washington, DC.

Pimentel, M. A., Clifton, D. A., Clifton, L., and Tarassenko, L. (2014). A review of novelty detection. *Signal Processing*, 99:215–249.

Sommer, M., Tomforde, S., and Hähner, J. (2016). An organic computing approach to resilient traffic management. In McCluskey, T. L., Kotsialos, A., Müller, J. P., Klügl, F., Rana, O., and Schumann, R., editors, *Autonomic Road Transport Support Systems*, pages 113–130. Birkhäuser, Basel.

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., Prothmann, H., Rochner, F., Branke, J., Hähner, J., Müller-Schloer, C., and Schmeck, H. (2008). Decentralised progressive signal systems for organic traffic control. In *2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, pages 413–422. IEEE.