Streaming-Based Anomaly Detection in ITS Messages

Juliet Chebet Moso, Stéphane Cormier, Cyril de Runz, Hacène Fouchal, John Mwangi Wandeto

applied sciences

MDPI

Article

Streaming-Based Anomaly Detection in ITS Messages

Juliet Chebet Moso [1,2,*], Stéphane Cormier [1], Cyril de Runz [3], Hacène Fouchal [1] and John Mwangi Wandeto [2]

[1] CReSTIC EA 3804, Université de Reims Champagne-Ardenne, 51097 Reims, France; stephane.cormier@univ-reims.fr (S.C.); hacene.fouchal@univ-reims.fr (H.F.)
[2] Computer Science, Dedan Kimathi University of Technology, Private Bag, Dedan Kimathi, Nyeri 10143, Kenya; john.wandeto@dkut.ac.ke
[3] BDTLN, LIFAT, University of Tours, Place Jean Jaurès, 41000 Blois, France; cyril.derunz@univ-tours.fr
* Correspondence: juliet.moso@dkut.ac.ke

Abstract: Intelligent transportation systems (ITS) enhance safety, comfort, transport efficiency, and environmental conservation by allowing vehicles to communicate wirelessly with other vehicles and road infrastructure. Cooperative awareness messages (CAMs) contain information about vehicles status, which can reveal road anomalies. Knowing the location, time, and frequency of these anomalies is valuable to road users and road authorities, and timely detection is critical for emergency response teams, resulting in improved efficiency in rescue operations. An enhanced locally selective combination in parallel outlier ensembles (ELSCP) technique is proposed for data stream anomaly detection. A data-driven approach is considered with the objective of detecting anomalies on the fly from CAMs using unsupervised detection approaches. Based on the experiments carried out, we note that ELSCP outperforms other techniques, with 3.64 % and 9.83 % better performance than the second-best technique, LSCP, on AUC-ROC and AUCPR, respectively. Based on our findings, ELSCP can effectively detect anomalies in CAMs.

Keywords: anomaly detection; data streams; intelligent transportation systems; traffic incident detection

Citation: Moso, J.C.; Cormier, S.; de Runz, C.; Fouchal, H.; Wandeto, J.M. Streaming-Based Anomaly Detection in ITS Messages. *Appl. Sci.* **2023**, *13*, 7313. https://doi.org/10.3390/app13127313

Academic Editor: Juan-Carlos Cano

Received: 15 May 2023
Revised: 14 June 2023
Accepted: 17 June 2023
Published: 20 June 2023

1. Introduction

The advancement of sensor monitoring technologies and low-cost solutions, together with the introduction of the Internet of Things (IoT) in everyday life, has resulted in the capture of huge volumes of data [1]. Data streams are huge, continuous, unbounded sequences of data that are generated at a rapid rate and have a dynamic distribution. Data stream mining is an ongoing study subject that recently emerged in order to extract knowledge from enormous amounts of continuously created data. Cooperative intelligent transport systems (C-ITSs) with networked vehicles are poised to transform mobility's future. The flow of messages between vehicles via vehicle-to-vehicle communication (V2V) and between vehicles and transportation infrastructure via vehicle-to-infrastructure communication (V2I) facilitates this transformation. Cooperative awareness messages (CAMs) provide real-time information about individual vehicles. Nonetheless, due to the novelty of the idea, the impact of C-ITS services on road networks has yet to be fully felt and analysed [2].

Anomalies are "patterns in data that do not conform to a well-defined notion of normal behaviour" [3]. They are classified into three types: point anomalies, contextual anomalies, and collective anomalies [3]. Point anomalies, or "outliers", are individual data components that are inconsistent or anomalous in relation to all other data elements [4]. Contextual anomalies are data elements that are considered unusual in a certain context. Collective anomalies are groups or sequences of connected data components that are out of sync with the rest of the dataset. For example, excessive traffic on a highway during business hours is usual, yet it is contextually anomalous traffic behaviour after midnight [1]. Contextual attributes (such as time of day, season, and location) and behaviour attributes define each

data piece when seen in context. The early identification of anomalies can decrease event risks, such as accidents and traffic jams. The majority of these occurrences may be attributed to driver error or poor road conditions. Road users and authorities benefit from identifying the location, time, and frequency of these road abnormalities.

Traffic incidents are non-recurring events that may cause traffic congestion and travel time delays. An incident is "an unexpected event that temporarily disrupts the flow of traffic on a segment of a roadway" [5]. To lessen the impact and duration of incidents, it is critical to understand the frequency of occurrences by spotting variations from usual traffic patterns. Road occurrences/anomalies include car wrecks, vehicle breakdowns, debris on the road, and vehicle(s) stalled in the middle of the road. Two forms of traffic irregularities include traffic jams and road management [6]. Short-term traffic disturbances may persist for a span of minutes or several hours, inducing a decline in traffic velocity or an upsurge in traffic density. Resolving long-term traffic management anomalies is a challenging task that may require considerable time and effort. The examination of deviations in traffic can be conducted by examining either local traffic anomalies or group traffic anomalies. The road network is divided into separate segments for local traffic anomalies, and each segment is analysed for individual abnormalities. For group traffic anomalies, any irregularity detected in one portion of a road network will influence and be assessed by analysing the causal connections between adjacent segments.

The detection of incidents in [5] relies on actual Global Positioning System (GPS) data collected from vehicle tracks. The road network is segmented by road type, date, time, and the predominant weather conditions. Segments that exhibit a significantly lower average speed than the designated normal speed are regarded as abnormal and are extracted. The problem with this technique is that the segmentation process is impacted by the precision of polygonal line coordinates, and the accuracy range of GPS influences the differentiation of incidence from typical traffic congestion. To identify long-term abnormal traffic zones in big centres, ref. [7] proposes long-term traffic anomaly detection (LoTAD). The method divides the road network into sections by utilising bus line data and an actual bus trajectory dataset, which results in temporal and spatial segments known as TS segments. Anomalies in bus lines are detected through the computation of an anomaly index, utilising the average velocity and average stop time as trajectory features. Utilising the data obtained from the atypical areas can provide valuable input for future urban traffic planning. These kinds of anomalies can be detected with the tool proposed in [8].

The filter–discovery–match (FDM) method [9] is a suggestion for determining accident locations. It involves dividing a roadway network into sections and creating speed vectors using the average speed. Actual incident records are used to determine the specific sections of road where the incident took place. Subsequently, the speed vectors of vehicles passing through those sections during the incident time are extracted. The regular velocity direction of the road sections is determined by computing the average velocity of the automobiles that crossed those sections within a specific time period and were not impacted by any traffic disruptions. The velocity disparities between the incident speed vectors and regular speed vectors for each segment are utilised to determine the candidate speed patterns. Through thorough experimentation using both real taxi data and simulated data, it was discovered that FDM resulted in a lower mean time-to-detect (MTTD) when compared to other existing techniques.

A comprehensive body of research was conducted to create diverse anomaly detection algorithms that encompass several categories, namely classification, nearest neighbour, clustering, statistical, information theoretic, spectral, and graph-based approaches [3,10]. Histogram-based outlier score (HBOS) [11] operates on the premise of feature independence and computes outlier scores via the construction of histograms for individual features. Swift computation time is facilitated without the need for data labelling. Time is essential in computing, especially in C-ITS, where an enormous volume of data must be analysed to identify irregularities. Deviances from typical road traffic data are perceived by analysing the intricate attributes of constructed histograms to spot anomalies [12]. Two categories of

histograms are possible to construct: static bin-width and dynamic bin-width histograms. To achieve the uniform weighting of every feature, the bin's maximum height is standardised to one by normalising the histograms and flipping the quantified results, resulting in abnormal occurrences receiving a higher score and normal instances receiving a lower score. This action aims to minimise the impact of floating-point precision errors that can lead to imbalanced distributions and elevated scores. For each instance x, the HBOS is determined by the height of the bin in which the instance is placed:

$$HBOS(x) = \sum_{i=0}^{d} \log\left(\frac{1}{hist_i(x)}\right) \quad (1)$$

where d denotes the number of features, x is the vector of features, and $hist_i(x)$ is the density estimation of each feature instance.

HBOS scoring produces numerical values that indicate the degree of "outlierness" of each data point in relation to the rest of the dataset. The last stage involves thresholding, where a decision label is assigned to each element, indicating whether it is a regular instance or an anomaly, depending on the threshold parameter Th. Different statistical deviation measures, such as standard deviation, median absolute deviation (MAD), quantiles, and streaming analysis with a defined window can be utilised to establish the value of the Th parameter. If a score exceeds three times the standard deviation, it can be deemed an anomaly. Another method is to order the scores so that a top_k algorithm provides the k most anomalous observations.

Anomalies can be identified by assuming that the data follow a specific probability distribution and categorising data points with a low probability density as anomalous. In an elliptical distribution, the Mahalanobis distance between each point and the mean is calculated, with points exceeding a predetermined threshold being categorised as anomalies. Due to its ability to resist outlier observations, the minimum covariance determinant (MCD) [13] serves as a highly dependable means for identifying anomalies in multivariate contexts. Given a dataset presented as an $n \times p$ matrix, where n refers to the number of occurrences and p relates to the number of features. The initial stage in obtaining the MCD estimator involves calculating the covariance matrix's determinant. A smaller set of observations (consisting of h data points, wherein $n/2 \leq h \leq n$) is selected from a larger sample of n data points. This selection is made in a way that minimises the generalised variance of the subset h. The MCD estimator defines the following location and scatter estimates [14]:

1. $\hat{\mu}_0$, the mean of the h observations with the least possible determinant of the sample covariance matrix.
2. $\hat{\Sigma}_0$ is the associated covariance matrix multiplied by a consistency factor c_0.

The mean and the covariance matrix are used to calculate the robust distance for a point x defined as [14]

$$RD(x) = d(x, \hat{\mu}_{\mathrm{MCD}}, \hat{\Sigma}_{\mathrm{MCD}}) \quad (2)$$

where $\hat{\mu}_{\mathrm{MCD}}$ is the MCD estimate of location, and $\hat{\Sigma}_{\mathrm{MCD}}$ is the MCD covariance estimate. MCD selects the section of the data with the closest distribution to eliminate anomalies, as they tend to be far from the bulk of the data. This minimises the masking effect caused by atypical observations [15].

The isolation forest (IForest) [16] algorithm is utilised to uncover anomalies in data that have a high number of dimensions.It is a non-parametric method that demonstrates favourable results when applied to normally distributed, unbiased data that contain minimal noise [4]. Its suitability for anomaly detection in C-ITS data lies in the fact that the data lack prior distribution and remain unlabeled. The IForest model is composed of a collection of unique, random isolation trees *itrees* that are divided into nodes through recursive partitioning. IForest's scoring stage computes an anomaly score for each data observation within the dataset. The outlier score is calculated based on the distance between the leaf

and the root. The ultimate outcome is obtained by taking the mean of the distances from the individual data points to the different *itrees* within the isolation forest. Given an instance x, the anomaly score is defined as

$$c(n) = 2H(n-1) - (2(n-1)/n) \tag{3}$$

$$E(h(x)) = \sum_{i=1}^{t} h_i(x) \tag{4}$$

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}} \tag{5}$$

where $E(h(x))$ is the average path length of sample x over t *itrees*, $c(n)$ is the average path length of the unsuccessful search in the binary search tree, and $H(i)$= ln(i) + γ (γ is Euler's constant). Based on the anomaly score s, the following conclusions can be made [17]:

1. If instances return $s(x, n)$ extremely close to 1, then they are anomalies;
2. If instances have an $s(x, n)$ less than 0.5, then they are deemed normal instances;
3. If all the instances return an $s(x, n)$ of 0.5, then there is no differentiation between normal and anomalous instances.

Robust random cut forest (RRCF) [18], a variation of isolation forest designed for streaming data, incorporates concept drift and tree evolution to generate a measure of the isolation score. The tree structure is impacted by the degree to which a new point alters the anomaly score. Consequently, the sensitivity of RRCF is reduced when the sample size is decreased. A robust random cut data structure is utilised as a summary or representation of the input stream. While detecting anomalies, RRCF maintains the original distances between all pairs of data points. The LSCP (locally selective combination in parallel outlier ensembles) [19] detector builds a small area surrounding a test instance, utilising the consensus of its nearest neighbours in randomly selected feature subspaces. It employs an average of maximum technique, in which a homogenous set of base detectors is fitted to the training data before generating a pseudo ground truth for each occurrence by picking the maximum outlier score. It locates and combines the best detectors in the area and investigates both global and local data linkages. Its strength is that it can quantify the magnitude of local outliers.

The local outlier factor (LOF) [20] measures how much a sample's density deviates from its neighbours on a localised level. The score for the anomaly is determined based on the object's isolation from its surroundings, giving it a localised significance. The distance between the k-nearest neighbours determines the locality, which is used to estimate the local density. The initial step is to compute the *k-distance* between a point p and its k-th neighbour. Measurement of the distance can be accomplished by various methods, though the Euclidean distance is frequently utilised (Equation (6)):

$$d(p,o) = \sqrt{\sum_{i=1}^{n} (p_i - o_i)^2} \tag{6}$$

Given a dataset D and a positive integer k, the k-nearest neighbours of p is any data point q, whose distance to p is not greater than *k-distance (p)* (Equation (7)):

$$N_{k\text{-}distance(p)}(p) = \{q \in D \setminus \{p\} \mid d(p,q) \leq k\text{-}distance(p)\} \tag{7}$$

The reachability distance of data point p with respect to data point o is defined using Equation (8):

$$reach\text{-}dist_k(p, o) = max\{k\text{-}distance(o), d(p,o)\} \tag{8}$$

The next step is the estimation of the local reachability density (*lrd*), which is inversely proportional to the average reachability distance of p to its nearest k neighbours (Equation (9)):

$$lrd_{\mathrm{MinPts}}(\mathrm{p}) = 1 \Big/ \left[\frac{\sum_{\mathrm{o} \in N_{MinPts}(\mathrm{p})} reach\text{-}dist_{MinPts}(p, o)}{\left| N_{MinPts}(p) \right|} \right] \tag{9}$$

The LOF is then calculated, which is the mean ratio of the *lrd* of point p to the *lrds* of its neighbouring points (Equation (10)). If a point is considerably distant from its surrounding points in relation to their proximity to one another, it is deemed an anomalous point:

$$LOF_{MinPts}(p) = \frac{\sum_{\mathrm{o} \in N_{MinPts}(p)} \frac{lrd_{MinPts}(\mathrm{o})}{lrd_{MinPts}(p)}}{\left| N_{MinPts}(p) \right|} \tag{10}$$

LOF primarily excels at identifying outliers within a local context. If a point is proximal to a cluster with an extremely high density, it is classified as an anomaly. The interpretation of LOF is challenging, as it is presented in the form of a ratio. There is no specific threshold at which a point is considered an outlier. The identification of an anomaly is influenced by both the issue at hand and the individual analysing it. Streaming data refers to an ongoing influx of information that has the potential to be unending and can be regarded as a time series featuring multiple variables. The limitless influx of incoming data generates circumstances in which the data can transform over time, culminating in a scenario where modelling behaviour with more recent data holds greater relevance than using older data [21]. The stream data model may be described as follows:

$$Z \equiv \{z(1), z(2), \ldots, z(t), z(t+1), \ldots\} \tag{11}$$

where $z(t) \in \mathbb{R}^N$ for $t \geq 1$

Algorithms specifically created for handling data streams are capable of managing enormous volumes of data. The fundamental concept of processing data streams is that instances are assessed just once upon arrival and eliminated to make room for succeeding instances. The algorithm analysing the stream lacks the ability to dictate the order of encountered instances, thereby necessitating that its model be adjusted in a stepwise fashion for each inspection. The *"anytime property"* is another desirable characteristic that entails the model being readily available for usage at any given time interval during training. There are three primary challenges to identifying anomalies in data streams: limited memory capacity, imbalanced datasets, and concept drift [22]. Adapting streaming anomaly detection techniques to real-world applications is a straightforward task, owing to their high speed and limited memory constraints [23]. However, cutting-edge stream detection techniques are frequently geared towards detecting a certain sort of anomaly.

To prioritise fast processing and efficient storage in streaming situations, anomaly detection algorithms must possess the capability to swiftly and adeptly detect anomalies. In ref. [21], the stream outlier miner (STORM) algorithm was proposed for detecting outliers based on distance. Two versions of STORM, namely exact-STORM and approx-STORM, have been suggested to address outlier queries in accordance with the sliding window model. If the memory can hold the complete window, then the outliers are determined by utilising exact-STORM. If memory is scarce and the window cannot be accommodated, approx-STORM is employed to estimate the anomalies using efficient approximations that offer statistical assurance. STORM considers the time-based characteristics of an individual data point in a data stream. Every datum remains within the sliding window for a specific duration.

Detecting outliers is a subjective task that heavily relies on the problem domain, data traits, and the kinds of anomalies present; hence, the effectiveness of detection algorithms varies widely [1,24]. There is a chance that particular subspaces may be successfully identified by certain anomaly detection algorithms, whereas some may exhibit low detection

capabilities [25]. To minimise errors comprehensively, merging the knowledge domains of every algorithm is crucial [26]. Data points that fall outside the usual range for the whole dataset are known as global outliers, whereas local outliers can exist within the normal range of the entire dataset but exceed the normal range for nearby data points [27].

The idea of data locality was initially introduced by the authors of [28], and subsequently improved in [29] to facilitate dynamic classifier selection in local spaces of the training points. Dynamic strategies for selecting and merging base classifiers have yielded superior outcomes as opposed to static approaches that merely aggregate base classifier outputs through voting. The ensemble methods for learning involve combining the forecasts of multiple fundamental models to produce results that are more stable and reliable. For a reliable anomaly detection ensemble that produces consistent and impartial overall accuracy, it is preferable to incorporate a variety of base detectors and methodically integrate their results to create a robust detector.

Anomaly detection ensembles use parallel or sequential combination structures to improve accuracy by combining outcomes from multiple detectors. Parallel combination structures aim to minimise variance, while serial combination structures aim to mitigate bias [30]. Including all base detector outcomes in an ensemble may diminish its effectiveness, as different detectors may not identify specific anomalies, especially in unsupervised learning scenarios. Unsupervised algorithms for detecting anomalies aim to identify deviations in unlabelled datasets automatically, based on certain assumptions. The performance of a model can be evaluated based on the various features that exist within a dataset, and detection rates differ due to specialised models accommodating diverse observational characteristics. Using a collection of unique skills within an ensemble yields a stronger outcome than solely relying on one detector [31]. Some other studies have been considered to detect anomalies as in [32–34].

This study is focused on contextual anomalies. The notion of context originates from the structure of datasets, wherein two distinct sets of attributes characterise each data instance [3]:

- *Contextual attributes*: These are utilised for establishing the context (or neighbourhood) of a particular instance. Contextual features of a location in spatial datasets include its longitude and latitude. In time series data, time serves as a contextual characteristic that determines the position of an instance within the entirety of the sequence.
- *Behavioural/indicator attributes*: These attributes have a direct correlation with the anomaly detection process, as they establish the anomalous behaviour. Within a spatial data collection outlining the mean precipitation levels within a specific nation, the proportion of rainfall observed at a given site shall be deemed a behavioural attribute.

Our approach involves utilising data to actively detect anomalies through unsupervised methods that target local contextual anomalies. We propose an enhancement of an ensemble anomaly detector called enhanced locally selective combination in parallel outlier ensembles (ELSCP). ELSCP is tailored for streaming scenarios by leveraging a pipeline framework that transforms data into a stream and passes it to ELSCP using a reference window model that implements a sliding window approach. The updated version facilitates the handling of information in a continuous flow, thereby allowing us to assess how effective our algorithm is in a streaming environment. Our approach involves the use of hypothesis testing to identify any unusual patterns in vehicle movement on the road. The primary assumption of our analysis is that *"normal instances are far more frequent than anomalies"*. The central hypothesis is that *"If vehicles change their speed abruptly at a specific point, then it implies an incident has occurred"*. We seek to investigate the following questions:

(a) What is the significance of data associations in anomaly detection, especially in a constrained road network?
(b) How can a balance between variance and bias be achieved in ensemble learning?
(c) How can we improve the detection rate of anomalies in CAM data streams?
(d) Can enhancing the LSCP algorithm improve the identification of anomalies in CAM data streams?

(e) How can the adapted technique be applied to real-world problems?

We propose the following contributions:

1. We define and investigate the issue of completely unsupervised anomaly ensemble construction;
2. We propose a robust ensemble-based methodology for the detection of anomalies from data streams in the C-ITS context;
3. We evaluate the proposed technique using a dataset of CAM messages generated in the C-ITS environment and compare its performance with state-of-the-art techniques in the streaming context.

This paper is structured as follows: Section 2 presents the data generation and preprocessing steps. They correspond to the used materials. It also introduces our anomaly-detection approach, called enhanced LSCP (ELSCP), and the performance indicators that we used. Section 3 presents the experimental results. Section 4 is dedicated to discussion and limitations, with Section 5 giving the conclusion and future work.

2. Materials and Methods

The study concentrates on unsupervised contextual anomaly detection on C-ITS CAMs.

2.1. Data Generation, Pre-Processing and Transformation

A real-world dataset of CAMs that was collected from 80 cars in France between September 2018 and August 2019 as part of a C-ITS project was used as seed data to generate simulated data for CAMs. Simulation of vehicular ad hoc network (VANET) applications requires simulating both vehicle-to-vehicle wireless data transmission and vehicle mobility. Simulations were carried out using the objective modular network testbed (OMNET++) network simulator [35], and simulation of urban mobility (SUMO) road traffic simulator [36]. The artery vehicle-to-everything (V2X) simulation framework [37] was used to combine the network simulator with the road traffic simulator, allowing for efficient communication. This connection is especially crucial in C-ITS, which provides applications for traffic safety and efficiency.

2.1.1. Data Generation

The simulation depicted the movement of C-ITS equipped vehicles in the French city of Reims. The simulation settings were configured as shown in Table 1. The initial stage was to configure the SUMO simulator, after which the map of Reims, France, was retrieved from Openstreetmap (https://www.openstreetmap.org/, accessed on 12 February 2023) and loaded into the simulator. The second stage was the generation of random trips for a hundred vehicles. The SUMO simulator periodically transmitted the locations and details of each vehicle to the OMNeT++ simulator. The Veins framework then collected the sent data from the OMNeT++ simulator, and the Artery framework generated the CAM messages. The frequency of CAM message generation varied from 10 Hertz to 1 Hertz (100 milliseconds to 1000 milliseconds).

The secure message structure was used to construct the security headers and certificates for the CAM messages in the Vanetza framework public key infrastructure (PKI) system as stated in the European Telecommunications Standards Institute Technical Specifications, ETSI TS 103 097 [38]. The CAM signature and validation mechanisms were constructed using the PKI system's pseudonyms. Each generated CAM was signed with a pseudonym by appending the signature at the end of the message. This signature is used by the receiving vehicle's validation system to authenticate the received message. The mobility and communications between the vehicles were designed to simulate actual movement in the C-ITS environment. This was done to ensure that the outcomes were as realistic as possible. The simulation ran for two hours, with each vehicle taking a random route. The CAM messages for each vehicle for the entire simulation duration were saved in separate files.

Table 1. Simulation parameters.

Parameter	Value
Network simulator	OMNeT++
Road traffic simulator	SUMO
Framework	Artery (Vanetza, INET, Veins)
Number of nodes	100
Simulation time	7200 s
Road map of Reims	10,000 m × 10,000 m
CAM message interval	0.1 s
Carrier Frequency	5.9 GHz
Number of channels	180
Transmitter power	20 mW

2.1.2. Data Pre-Processing and Transformation

One typical strategy for traffic incident identification is to learn traffic patterns from previously observed accumulated traffic data and identify instances when the real-time traffic data differ significantly from the learned patterns [5,39]. A important component in this respect is the vehicle's speed, which has a direct influence on the safety, productivity, and the degree of environmental implications for the traffic ecosystem.

It is presumed that a vehicle's speed data recorded at successive timestamps would demonstrate temporal continuity with minor, calculated deviations. As a result, the speed standard deviation of a collection of messages belonging to a small sub-trajectory T_i^k of a trajectory T_k should be low in normal conditions but may be significant under anomalous conditions. Speeds measured at spatially close locations should also show spatial continuity with minimum fluctuation. Anomaly detection using CAM messages is a multivariate task, in which sub-trajectory sizes must be considered in order to be able to learn typical conditions and recognise abnormal ones. Anomalies are defined as points that do not appear to fit in with the rest of the dataset, based on the assumption that the vast majority of occurrences in the dataset are normal.

We present strategies for predicting the occurrence of an incident on a road section based on hypothesis testing. Our research is predicated on the premise that "if there is a traffic incident on the road, the vehicle speed, as captured in the sent messages, will significantly differ from the typical or expected speed at that section". The central hypothesis is that *"If vehicles change their speed abruptly at a specific point, then it implies an incident has occurred"*. We target non-recurring traffic disruptions, the occurrence of which is usually unexpected and random. In the event that an incident closes the whole lane, we anticipate a significant change in speed as well as a change in heading as the cars approach the incident area (as shown in Figure 1). We focus on local contextual anomalies, where an occurrence may affect a sub-trajectory, and the contextual anomaly can be detected from the behaviour features, speed, and heading.

Trajectory mining was utilised to extract the sample data for the study using the PostgreSQL database system, with the spatial extension PostGIS employed for processing spatial data. The CAMs created by vehicles passing through Boulevard Dauphinot (route N51) in Reims, France, were the subject of the study. The initial step was to transform each message's latitude and longitude information into geometry data types using Spatial Reference Systems SRID 4326 (WGS 84), Europe's projection. All messages within a 150-metre radius of an established reference location were retrieved using the *ST_DWithin* function in PostGIS to build a bounding box. This produced messages for a 300-metre section of road that served as our research area. Since C-ITS vehicles have a communication

range of 150 m [40], a roadside unit (RSU) can be stationed at the centre of the designated route. The extracted dataset contains trajectories for 40 vehicles.

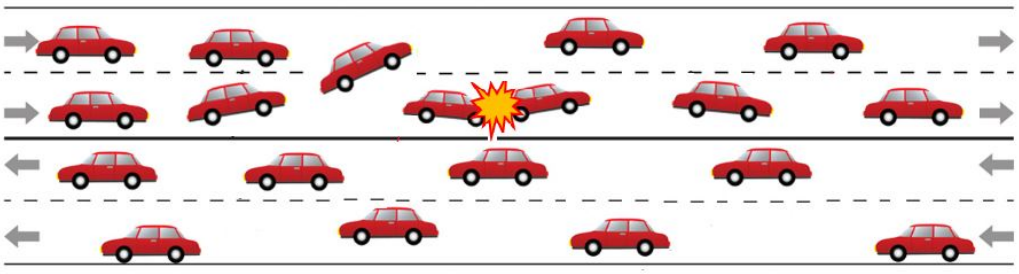

Figure 1. Illustration of an accident scenario on the road.

Anomalies were added to the extracted dataset to construct an evaluation dataset by placing an "obstacle" in the centre of the road segment [41]. The driver is deemed to have noticed the obstacle when a message comes within range of it (based on distance computation using latitude and longitude), and the data in the message are changed. The speed is lowered by a number chosen at random from a defined range of values. The subsequent messages will be updated as well, depending on the position and length of the obstacle. The data anomalies reflected incidents on the road when cars had to lower their speed as they approached the obstruction, maintain the reduced speed for the length of the obstacle, and then restore regular speed after passing the obstacle. The presented obstacles simulate road accidents (such as a stalled car, an accident, or road debris), allowing us to label the messages of the vehicles affected by this incident as anomalous. The remaining messages were not altered. Figure 2 presents the trajectories (in blue) on Boulevard Dauphinot (route N51) and the obstacle section, showing normal points in green with anomalies in red.

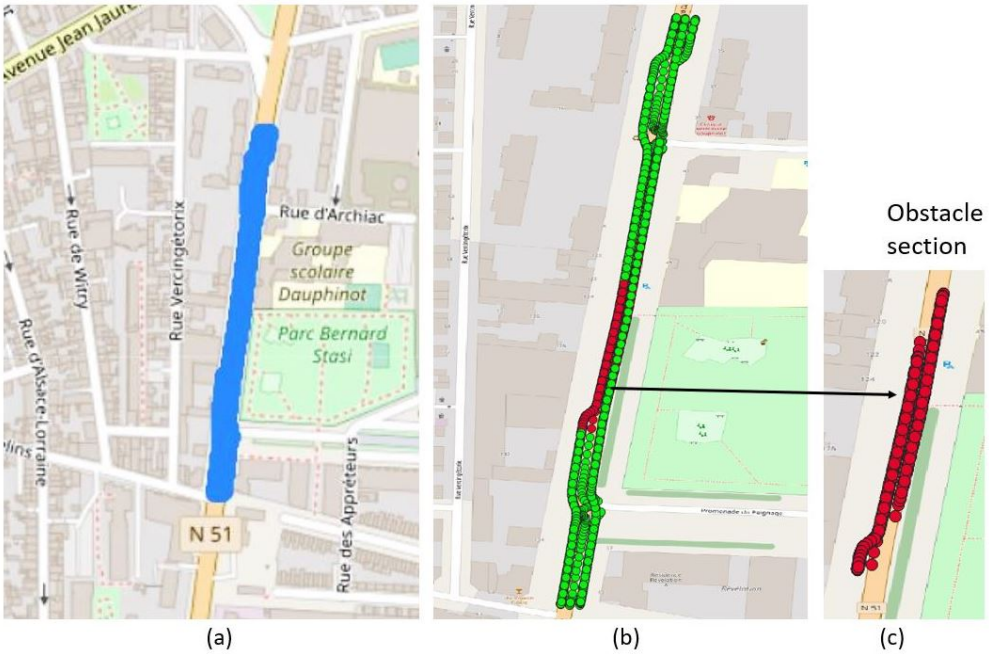

Figure 2. Area of interest; (**a**) trajectories on Boulevard Dauphinot (route N51); (**b**) obstacle section: normal instances in green and anomalies in red; (**c**) obstacle section showing only the anomalies.

The vehicle ID, timestamp, latitude, longitude, vehicle speed, and heading are used to detect multivariate anomalies. The data are encoded as a data stream and fed into an ensemble-based technique for event detection that makes use of the windowing concept. Given that the focus is on non-recurring events, the anomalies of interest give rise to an imbalanced data set.

Therefore, let $\mathcal{A}_m$ be the set of CAM message anomalies. Then,

$$\forall m, Pr(m \in \mathcal{A}_m) << 1 - Pr(m \in \mathcal{A}_m)$$

Further, let $\mathcal{A}_T$, the set of (sub-) trajectory anomalies, then

$$\forall T_k, Pr(T_k \in \mathcal{A}_T) << 1 - Pr(T_k \in \mathcal{A}_T)$$

2.2. Problem Statement

Definition 1. *Message: Each CAM message* $\mathrm{m_i}$ *is defined by* $< v_{id}, t, x, y, s, h >$*, where we have the following:*

- v_{id} *is the vehicle identifier;*
- t *is the timestamp of the message;*
- x *is the longitude of the vehicle* v_k *at time* t*;*
- y *is the latitude of* v_k *at time* t*;*
- s *is the speed of* v_k *at time* t *in metres/second;*
- h *is the heading of* v_k *at time* t *in degrees.*

Definition 2. *Vehicle Trajectory: A trajectory is a time-ordered sequence of n messages belonging to a given vehicle such that* $T_k =< m_1, m_2, \ldots, m_n >$

In this work, a trajectory is defined as the collection of all messages that are uniquely identifiable by a single identifier. A sub-trajectory is a series of sequential messages that are part of a trajectory.

Definition 3. *Sub-Trajectory* T_k^i *is a sub-trajectory of* T_k *if and only if the following hold:*

- T_k^i *is a trajectory:* $T_k^i = \langle m_{i_1}, \ldots, m_{i_p} \rangle$.
- $\forall m_{i_j} \in T_k^i$, $m_{i_j} \in T_k$.
- *All consecutive messages in* T_k^i *are also consecutive in* T_k*, i.e., there does not exist a message of* T_k *situated between messages of* T_k^i *that does not belong to* T_k^i*:*

$$\nexists m_j \in T_k \text{ and } m_j \notin T_k^i \text{ with}$$

$$Rank(T_k, m_{i_1}) \leq Rank(T_k, \mathrm{m_j}) \leq Rank(T_k, m_{i_p})$$

Suppose we have a dataset $Z = \{\mathrm{z_1}, \mathrm{z_2}, \ldots, \mathrm{z_n}\}$, where n is the total number of instances. Each instance i of Z consists of both contextual and behavioural attributes. Additionally, within Z, we have a set O of instances that are outliers or anomalies. Our goal is to assign each instance i with an outlierness score S_i such that outliers in O have much higher values than other instances. The outlierness of an instance results from the abnormal behavioural attributes in its context. Given the contextual attributes, there is an underlying pattern that limits the behavioural attributes to some expected values, beyond which an instance is considered an outlier.

Definition 4. *Contextual outlier: Based on its contextual features, this is an instance whose behavioural attributes contradict the dependent pattern.*

Definition 5. *Contextual neighbours: Contextual neighbours of instance i are the instances that are comparable to it based on its contextual attributes. In principle, the collection of contextual neighbours of instance i is*

$$CN_i = \{j : j \in D \wedge j \neq i \wedge \text{sim}(x_i, x_j) \geq \varphi\} \quad (12)$$

where x_i and x_j are contextual attribute vectors, $D = \{1, 2, \ldots, N\}$ denotes the set of instances indexes, $sim(\cdot)$ is a similarity function of two vectors, which in our case is a correlation measure, and φ is a predefined similarity threshold.

The study concentrates on unsupervised contextual anomaly detection. It is preferable to develop a robust model capable of efficiently and reliably predicting an observation as an anomaly when the behavioural attributes are anomalous in the context. Moreover, such a model should be sensitive to abnormalities in the contextual attributes and make meaningful predictions using the best available relevant context. With the goal of improving detection performance in anomaly detection, an ensemble-based anomaly detection technique with heterogeneous base detectors is developed.

2.3. Proposed Enhanced LSCP Algorithm (ELSCP)

Unsupervised outlier ensemble implementations lack labels for "outliers" and "inliners". Hence, it is difficult to develop a reliable technique for selecting competent base detectors and maintaining model stability. Conventional unsupervised combination approaches in parallel ensembles are usually general and global (e.g., averaging, maximising, and weighted averaging) but do not consider locality. LSCP [19] uses the concept of nearest neighbours in randomly chosen feature subspaces to build a local region around a datapoint. The best base detectors in this neighbourhood are picked and combined to create the final model. The LSCP algorithm was created to address two problems: a lack of ground truth and a lack of a dependable technique for choosing the best base detectors. We devised ELSCP, a novel method that improves on LSCP by improving how the local region definition is retrieved and the selection of competent detectors.

The management of variance and bias, especially in ensemble approaches, is an inherent difficulty in anomaly detection systems. An inherent challenge in anomaly detection algorithms is how to handle variance and bias, especially in ensemble techniques. According to [42], variance is decreased by integrating heterogeneous base detectors using techniques such as averaging, maximum of average, and average of maximum. In contrast, a composite of all base detectors may contain errors, resulting in increased bias. As outlined in Aggarwal's bias–variance framework, ELSCP combines variance and bias reduction. It improves variance reduction by introducing diversity through the initialisation of heterogeneous base detectors with different hyperparameters. ELSCP also focuses on detector selection based on local competency, which aids in identifying base detectors with conditionally low model bias.

To compute the local region, LSCP employs the K-dimensional tree (KD-Tree) k-nearest neighbours (kNN) method with Euclidean distance. The space partitioning strategy utilised determines the effectiveness of the search trees, which can be binary or multidimensional [43]. KD-tree involves binary splitting, in which each split considers just one dimension. In a two-dimensional space, for example, the binary splitting hyperplane is parallel to either the X or Y dimension. Ball tree employs a multidimensional method, in which the split criterion is more flexible and can take values from multiple or all dimensions into consideration. The splitting criterion requirements also dictate the geometry of the resultant partitions, with KD-trees having rectangular partitions and ball trees having spherical partitions.

When dealing with skewed datasets, binary slits generate elongated skinny rectangles, increasing the number of backtracking levels during search, or making the tree highly imbalanced. Additionally, rectangles and squares are not the best shapes for splitting. This is because if the target point is at the corner of a rectangle, tracing many nodes around the

corner to determine the nearest neighbour would complicate the search algorithm [44]. As a result, the most efficient approach is to use a metric tree, such as the ball tree, where the hypersphere spatial partitioning is explicitly adjusted to the distance function [43,45].

Considering we are working with spatial–temporal data, our aim in anomaly detection is to discover local contextual anomalies where our contextual variables are longitude and latitude. The Harvesine distance is the most exact measure for determining spherical distances on the Earth's surface. As a result, we suggest combining the ball tree KNN method with the Harvesine distance metric to enhance local region definition. The computation of the Harvesine distance is presented in Equation (13):

$$d = 2\,r\,sin^{-1}\left(\sqrt{sin^2\left(\frac{\varphi_2-\varphi_1}{2}\right)+cos(\varphi_1)\,cos(\varphi_2)\,sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right) \tag{13}$$

where r is the radius of the Earth (6371 km), d is the distance between two points, φ_1 and φ_2 are the latitudes of the two points, and λ_1 and λ_2 are the longitudes of the two points, respectively.

Suppose we have a dataset $\mathbb{R}$ that is divided into training and test datasets: let $X_{train} \in \mathbb{R}^{m\times a}$ represent the set of training data with m points and a attributes, and $X_{test} \in \mathbb{R}^{y\times a}$ be the set of test data with y points and a attributes. The ELSCP technique begins with a heterogeneous list of base detectors D being fitted to the training data. The outcome of this training is a predicted set of outlier scores O_{train}, which is presented in Equation (14):

$$O(X_{train}) = [D_1(X_{train}), D_2(X_{train}), \ldots, D_k(X_{train})] \in \mathbb{R}^{m\times k} \tag{14}$$

The second step is to generate the training pseudo ground truth (*target*) by picking the highest score across all detectors from O_{train}. The third step is to define a local (ξ) for each test instance, X_{test_i}. This is defined by calculating each instance's k nearest neighbours using the ball tree KNN algorithm with the Harvesine distance. This is formalised as follows:

$$\xi = x_j \,|\, x_j \in X_{train}),\ x_j \in L_{ens} \tag{15}$$

where L_{ens} is the set of a test instance's nearest neighbours according to ball tree ensemble criteria.

Feature spaces are created by randomly picking t groups of $[d/2, d]$ features to form the local region L_{ens}. The k training objects closest to $X_{test\,i}$ in each group are found using the Harvesine distance. Training objects that occur more than $t/2$ times are added to the L_{ens} array. After defining the local region, a local pseudo ground truth ($target_s \in \mathbb{R}^{s\times l}$) is constructed by extracting the points in ξ from *target*. Using Equation (16), the local training outlier score for the test instance is obtained from the pre-computed training score matrix O_{train}:

$$O(X_{train_s}) = [D_1(X_{train_s}), D_2(X_{train_s}), \ldots, D_k(X_{train_s})] \in \mathbb{R}^{s\times k} \tag{16}$$

The fourth step is to choose the optimal detector. This is done by calculating the similarity between the base detector scores and pseudo target using correlation measures. The absence of direct and consistent access to binary labels in unsupervised outlier identification motivates similarity computation. Although it is feasible to convert pseudo outlier scores to binary labels, obtaining an accurate conversion threshold is challenging [46]. Furthermore, because imbalanced datasets are common in outlier identification tasks, utilising similarity measures is more reliable [19]. We propose an implementation of ELSCP that computes the final score using Pearson correlation and weights.

Definition 6. *Pearson Correlation: Let x and y be two vectors of length n, where $\overline{x}$ and $\overline{y}$ are the means of the vectors. The Pearson correlation coefficient is defined as the ratio of the co-variance of the two vectors to the product of their respective standard deviations (presented in Equation (17)). It assesses the linear relationship between two numerical variables. It also applies to features that are normally distributed:*

$$r_{xy} = \frac{\sum_{i=1}^{n}((x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{17}$$

where n is the sample size; $\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n}$ *denotes the mean of x; and* $\overline{y} = \frac{\sum_{i=1}^{n} y_i}{n}$ *denotes the mean of y.*

Using Equation (17), ELSCP computes the correlation between the local pseudo ground truth ($target_s$) and the local detector scores $D_i(X_{train_s})$ as $r(target_s, D_i(X_{train_s}))$. This calculation loops through all the k base detectors. A histogram is constructed with b equal intervals out of Pearson correlation scores, and detectors in the largest bin are chosen as competent base detectors for the given test instance. The Pearson correlation values are then ranked to compute weights. Finally, the selected detector scores are merged using the weighted average of the maximum strategy as $weighted\_avg(D_t^*(X_{test_i}))$ to obtain the final detection score. The implementation of ELSCP is summarised in Algorithm 1, with Figure 3 showing the flow chart.

Algorithm 1: Enhanced LSCP.

Input : the pool of heterogeneous detectors D, training data X_{train}, test data X_{test}, the local region size k
Output: Outlier scores for each instance in X_{test}

Train all base detectors in D on X_{train};
Generate training outlier scores O_{train} with Equation (14) ;
Generate pseudo ground truth: $target := max(O(X_{train}))$;
for *each test instance* X_{test_i} *in* X_{test} **do**
  Define local region **(ξ)** by Ball tree kNN ensemble ;
  Extract local pseudo ground truth $target_s$ by selecting k neighbours in **(ξ)** from $target$;
  for *each base detector* D_i *in* D **do**
    Get the outlier scores associated with training data in the local region $D_i(\xi)$;
    Evaluate the local competency of D_i by computing the similarity between $target_s$ and $D_i(\xi)$ using Pearson correlation with Equation (17) ;
  Select a group of t most similar detectors and add to the empty set D_t^*;
  Compute weights by ranking the Pearson correlation scores;
  return $weighted\_avg(D_t^*(X_{test_i}))$;
return *scores*;

ELSCP implementation applies two base detectors, HBOS and LOF. The local outlier factor (LOF) [20] determines how far a sample's density deviates from its neighbours on a local scale. It is local in the sense that the anomaly score is determined by the object's isolation from the surrounding area. The locality is determined by the distance between the k-nearest neighbours, which is used to estimate the local density. One can discover outliers (samples that have a much lower density than their neighbours) by comparing the local density of a sample to the local densities of its neighbours.

The procedure for ELSCP starts with HBOS and LOF base detectors being fitted to the training data. A pseudo ground truth for each train instance is generated by taking the maximum outlier score from all of the base detectors. We implement Pearson correlation in the model selection and combination phases. In the computation of the final ensemble score, we implement a weighted average, where the weight is computed by ranking the Pearson correlation scores. For each test instance, the following are performed:

1. Using the ball tree nearest neighbour algorithm with the Haversine distance metric, the local region is defined to be the set of the nearest training points in randomly sampled feature subspaces that occur more frequently using a defined threshold over multiple iterations.
2. Using the local region, a local pseudo ground truth is defined, and the Pearson correlation is calculated between each base detector's training outlier scores and the pseudo ground truth.
3. Weights are computed for each detector by ranking the Pearson correlation scores such that the detector with the best score obtains the highest weight.
4. Using the correlation scores, the best detector is selected. The final score for the test instance is computed using a weighted average of the best detector's local region scores.

ELSCP is adapted to the streaming context by implementing it through a pipeline. Given a dataset of CAMs, the stream simulator converts the data into a data stream. The stream is passed to ELSCP through a reference window model, which implements the windowing concept. Within the specified window size, a sliding window is implemented such that partial anomaly scores are generated for each model. The partial scores are then evaluated, and final scores are generated for each instance.

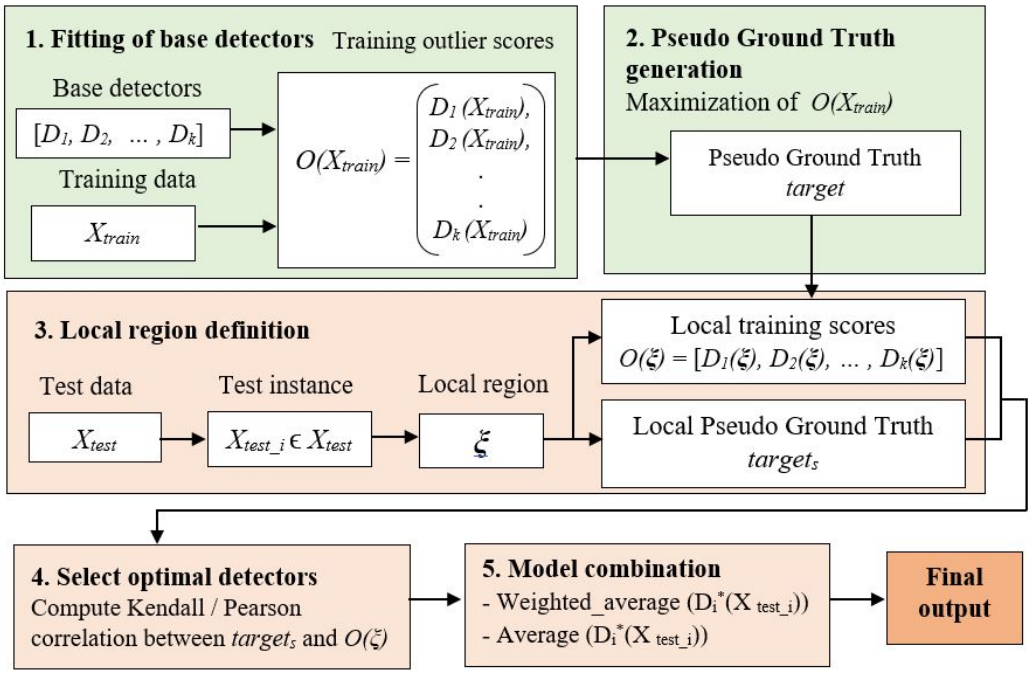

Figure 3. ELSCP flow chart: The results for steps 1 and 2 are cached; steps 3–5 are re-computed for each test instance.

A sliding window approach is employed in streaming anomaly detection, in which data samples inside a window are sorted by an outlier score, with highly ranked data samples being labelled anomalies. For ELSCP adapted to streaming context, a pipeline framework is adopted, where each incoming new instance x_t is passed through a pre-processor (unit norm scaler), which transforms x_t into a scaled feature vector without changing its dimensions. The scaled feature vector is then processed by the streaming anomaly detection model, which predicts the label y_t for the instance. This predicted label is then passed to the running average post processor, which converts the score to the average of all previous scores in the current window. Figure 4 depicts the proposed anomaly detection framework.

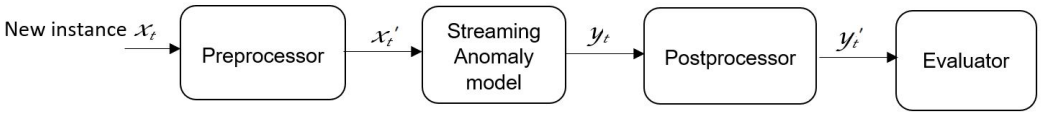

Figure 4. Anomaly detection framework.

The key advantage of using a sliding window is that with the arrival of a new data instance, a sliding window may be modified, resulting in an online and incremental updating process. The update mechanism necessitates the deletion of an old data instance and the storage of a new one, making it computationally efficient. The sliding window contains the latest subset of the dataset at any given time. As a result, a sliding window approach finds outliers based on the most recent subset and addresses the temporal property of data streams.

2.4. Performance Indicators

In the unsupervised outlier detection setting, it is often problematic to judge the effectiveness of the algorithms in a rigorous way. The majority of the outlier-detection algorithms output an outlier score, which is converted to a label based on a threshold. If the threshold selection is too restrictive (to minimise the number of declared outliers), then the algorithm will miss true outlier points (false negatives). On the other hand, too many false positives will be generated if the algorithm declares too many data points as outliers. This trade-off can be measured in terms of precision and recall, common measures of effectiveness.

In order to evaluate the performance of the different approaches, we use the area under the curve of the receiver operating characteristic (AUC-ROC) [47] and the area under the curve of precision–recall (AUCPR) [48]. Both indicators are based on the following concepts:

- True positive (TP): True positives are correctly identified anomalies.
- False positive (FP): False positives are incorrectly identified normal data.
- True negative (TN): True negatives are correctly identified normal data.
- False negative (FN): False negatives are incorrectly rejected anomalies.

The true positive rate (TPR), or recall, is

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}.$$

The false positive rate (FPR) is

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}.$$

The AUC-ROC receiver operating characteristics are TPR and FPR. The higher the AUC-ROC, the better the detection. AUC-ROC is the most popular evaluation measure for unsupervised outlier detection methods [49].

AUCPR uses precision and recall. Precision is the fraction of retrieved instances that are relevant [50]. Recall or sensitivity is the ability of a model to find all the relevant cases within a dataset:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

The AUCPR baseline is equivalent to the fraction of positives [51]:

$$\mathrm{AUCPR-baseline} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}}$$

AUCPR is a highly effective assessment metric that performs well for a variety of classification tasks. It is especially beneficial when dealing with imbalanced data when the minority class is more significant, like in anomaly detection. As a result, we used these metrics to assess the efficacy of anomaly detection algorithms in this study.

3. Results

This section presents the results obtained from experimenting with anomaly detection approaches on CAMs. The algorithms were written in Python using the Python Streaming Anomaly Detection (PySAD) framework [52]. This allows us to integrate batch processing algorithms from the Python outlier detection (PyOD) framework [53] and apply them to streaming data using a sliding window. The first series of studies was designed to see if the proposed ELSCP employing Pearson correlation outperformed LSCP. We wanted to know which model best estimated anomalous cases and what effect variations in window size had on model performance. Several experiments were performed with varying window sizes. Both algorithms employed two base detectors, HBOS and LOF. Table 2 summarises the parameters used in the experiments.

Table 2. Experimental parameters.

Parameters	Values
Window sizes	50, 100, 200, 300, 400, 500, 600
Sliding window size	50
Initial window (training set)	1000

Table 3 summarises the experimental results. Based on the data, it is evident that the improvements to ELSCP resulted in better AUC-ROC and AUCPR performance. Figures 5 and 6 indicate that the performance of both algorithms for AUC-ROC and AUCPR improves gradually with increasing the window size.

Table 3. ELSCP AUC-ROC and AUCPR performance results for window size variation.

	AUC-ROC		AUCPR	
Window Size	**LSCP**	**ELSCP**	**LSCP**	**ELSCP**
50	0.7408	0.7794	0.1793	0.2157
100	0.7763	0.8065	0.2126	0.2496
200	0.8203	0.8453	0.2525	0.3125
300	0.8706	0.8977	0.2989	0.3841
400	0.8833	0.9138	0.3145	0.4208
500	0.8910	0.9247	0.3206	0.4331
600	0.8917	0.9277	0.3206	0.4373
Average	0.8392	0.8707	0.2713	0.3504

The second series of tests was designed to assess the ELSCP performance by examining the true positive and false positive rates using a receiver operating characteristic (ROC) curve. The performance was also examined in terms of precision and recall. The baseline AUC-ROC is usually set at a value of 0.5, which suggests no discrimination; 0.7 to 0.8 is considered acceptable; 0.8 to 0.9 is considered excellent; and more than 0.9 is considered outstanding [54]. ELSCP was trained with 67% of the data, and the prediction performance was tested with 33% of the data. ELSP achieved 0.97 area under the ROC curve in both the true normal data (class 0) and true anomalies (class 1), which is outstanding performance

(as shown in Figure 7). The baseline AUCPR for the dataset was 0.087 (based on a sample size of 6341 with 552 true positive anomalies). Based on the results obtained from the precision recall curve (as shown in Figure 8), ELSCP achieved 0.66 for the positive class, which is good performance.

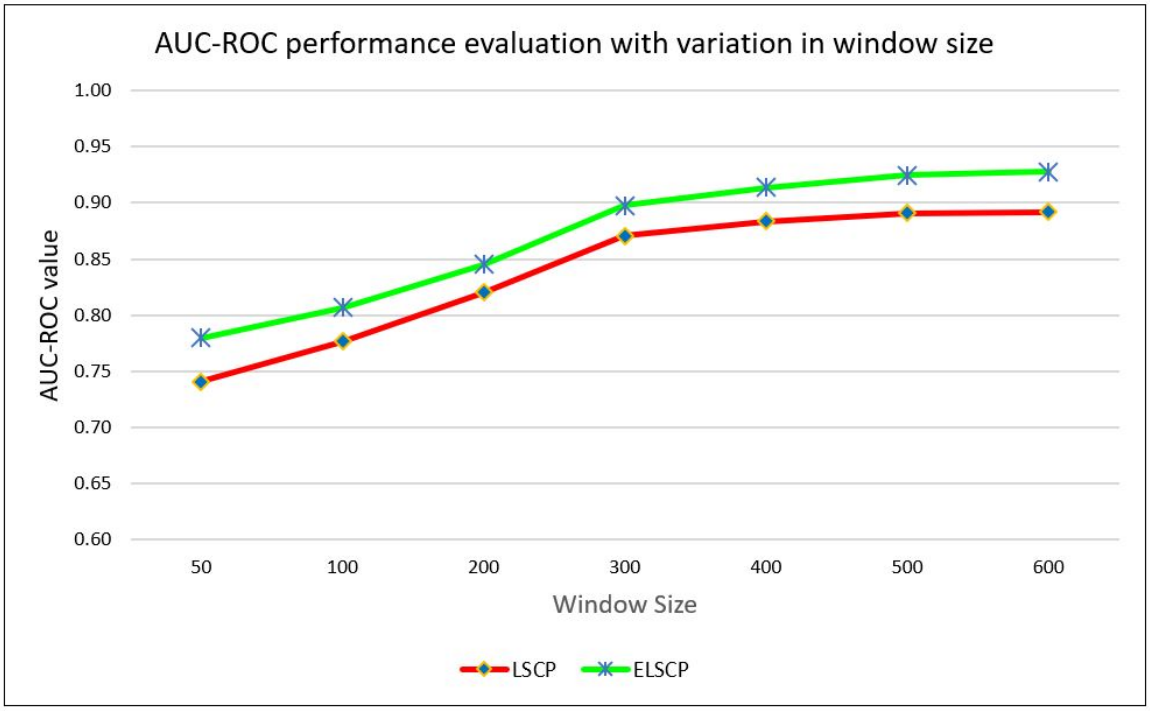

Figure 5. Comparison of models' AUC-ROC performance.

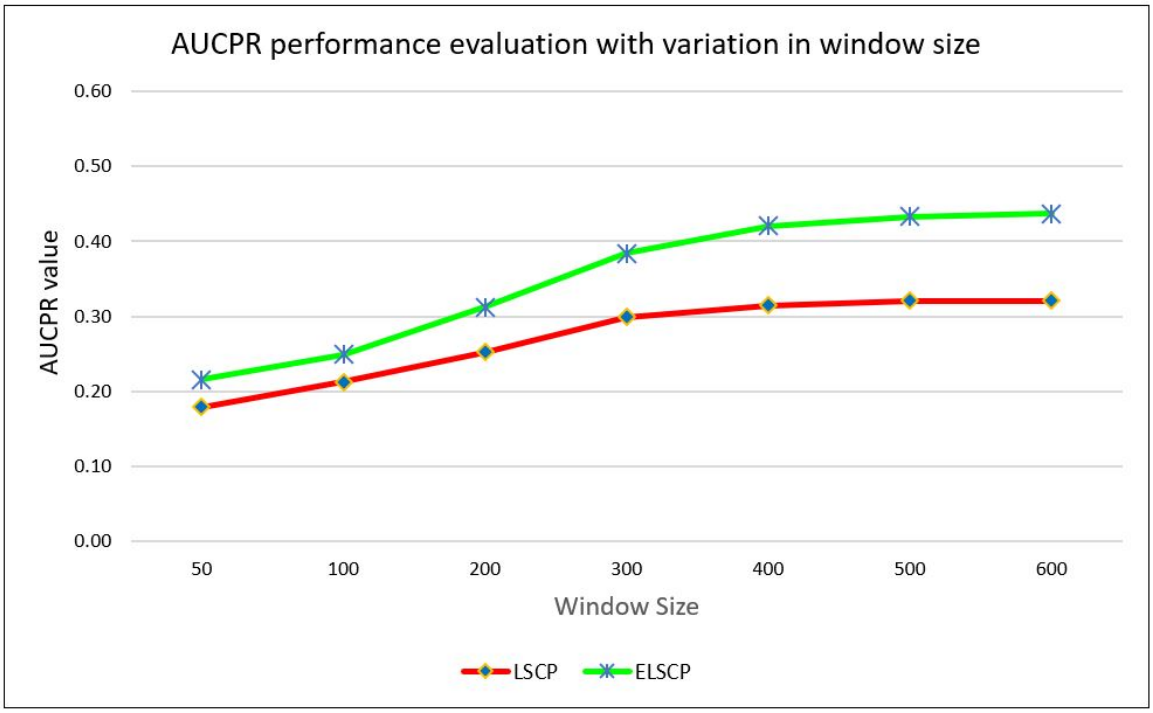

Figure 6. Comparison of models' AUCPR performance.

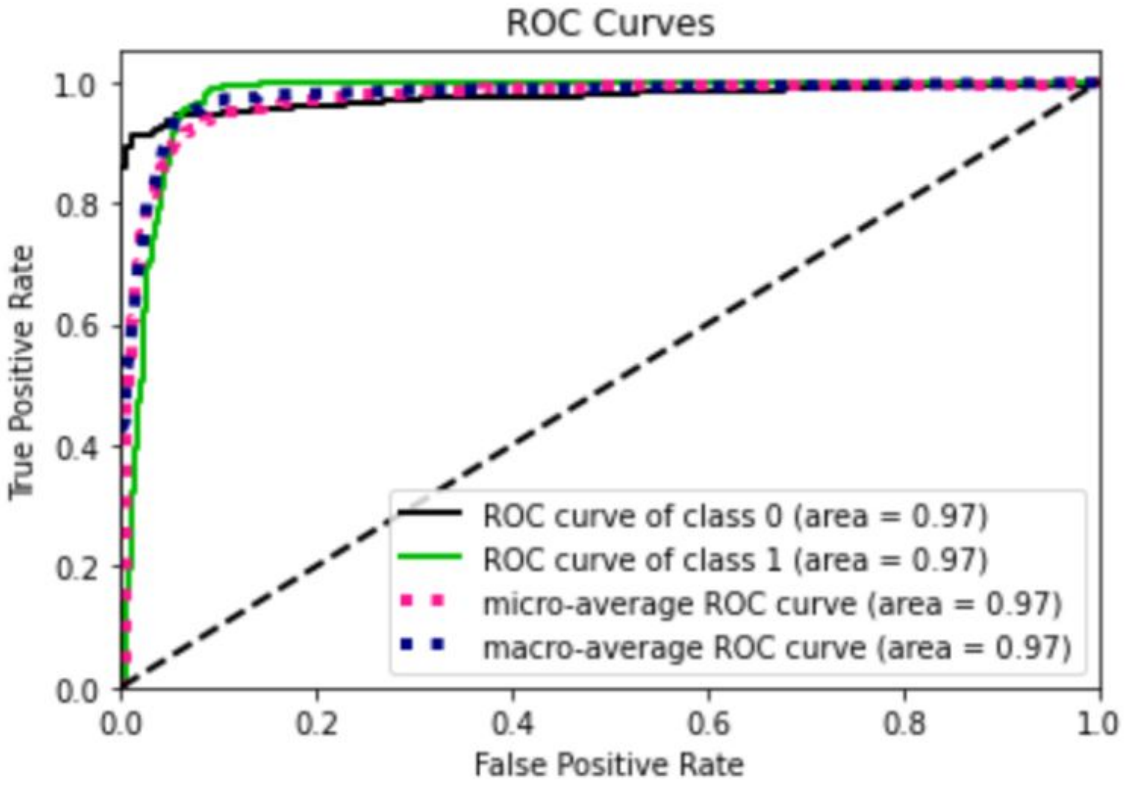

Figure 7. ELSCP ROC curves performance evaluation.

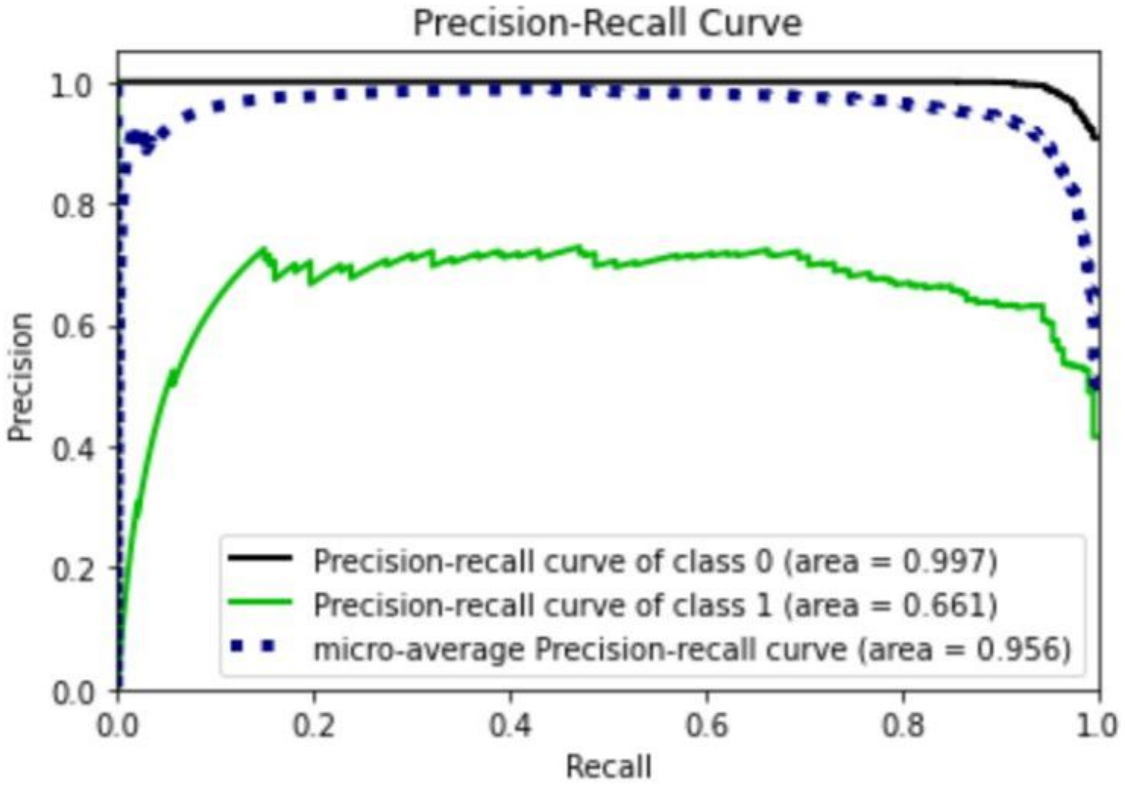

Figure 8. ELSCP precision–recall performance evaluation.

In the third set of tests, we endeavoured to determine the anomaly detection method that best estimates the number of anomalous events in C-ITS data. We used the data to analyse the performance of MCD, IForest, RRCF, exact-STORM, LSCP, and ELSCP algorithms in terms of AUC-ROC and AUCPR. The findings of the experiments are summarised in Table 4. According to the results, the batch processing models (MCD, IForest, LSCP, and ELSCP) that were adapted to the streaming context utilising the reference window model outperformed the streaming models (RRCF and exact-STORM). In terms of AUC-ROC, both LSCP and ELSCP performed admirably, with ELSCP surpassing LSCP by 3.64 percent. MCD and IForest both yielded satisfactory results. Exact-STORM was at the baseline and did not appear to differentiate between the positive and negative classes. In terms of AUCPR, all models except Exact-STORM can distinguish between the positive and negative classes since their average AUCPR value is higher than the baseline of 0.087.

Table 4. Average AUC-ROC and AUCPR performance results for various anomaly detection models on C-ITS data.

Model	AUC-ROC	AUCPR
ELSCP	0.8945	0.3841
LSCP	0.8581	0.2858
IForest	0.7825	0.2210
MCD	0.7587	0.1665
RRCF	0.6042	0.1146
Exact-STORM	0.5000	0.0885

4. Discussion

Advances in battery technology and the availability of low-cost storage devices permitted the collection of densely sampled trajectory data over a long period of time. With more data, it is now feasible to identify more intriguing patterns. A great deal of progress has been made in the field of anomaly detection systems, with several strategies developed to handle the problem of anomaly identification. In the field of connected autonomous vehicles, Xia et al. [55] developed a data collection and analytics framework for vehicle trajectory extraction, reconstruction, and assessment. To minimise noise and remove outliers in the trajectories, a Kalman filter and the Chi-square test were used. They also introduced a trajectory discontinuity detection approach that can detect and reconstruct discontinuous trajectories using a forward–backward prediction smoothing method.

Given that there is a lot of data without labels, unsupervised learning is widely favoured for real-life applications, particularly anomaly detection. In this work, we performed unsupervised anomaly detection in CAM data streams acquired from the C-ITS environment. We evaluated anomalies that might have implications, such as an accident or incident that requires motorists on that segment of the road to significantly lower their speed as they approach the event location. We propose an ensemble anomaly detector, enhanced locally selective combination in parallel outlier ensembles (ELSCP). ELSCP is tailored for streaming scenarios by leveraging a pipeline framework that transforms data into a stream using a reference window model that implements a sliding window approach. The approaches in the ensemble are combined in such a way that the total complexity does not surpass that of the individual model with the highest complexity. Based on our findings, ELSCP can detect anomalies in CAMs.

4.1. Use Cases

We selected streaming techniques because in real-world C-ITS setups, anomaly detection would be performed by roadside devices that gather a large number of signals from cars within range. This implies that it would have to process the messages on the fly for a variety of reasons (memory limitation, response time, etc.). The windowing concept makes it easier to discover abnormalities on the fly. This research can be applied to driver behaviour analysis, especially lane-changing behaviour analysis and obstacle detection. The

detection capabilities of ELSCP might be incorporated into road operators' decision-making processes in order to improve safety and traffic flow. The timely detection of anomalies is crucial, especially for emergency response teams, resulting in increased efficiency in rescue operations.

4.2. Limitations

The first drawback of ELSCP is the extraction of adjacent data points forming the local region of the test instance using a distance metric applied to the kNN ball tree algorithm. This strategy has two challenges:

1. It takes much time to find the test instance's nearest neighbours;
2. When many features or attributes are irrelevant, performance in a multidimensional space can be degraded.

To remedy this problem, the local-region definition can be solved by the use of fast approximate methods [56] or by prototyping [57]. Since these strategies do not require all data points, they can greatly minimise the time necessary to build up a local domain. The second drawback is that we used a basic maximisation approach to generate the pseudo-ground truth. This might be enhanced by considering exact techniques, such as active base detector pruning [58].

5. Conclusions

Detecting anomalies is a subjective task that relies on the problem domain, data traits, and the kinds of anomalies present. In this work, we studied the anomaly detection issue and applied it to messages generated by cars in C-ITS to detect anomalies characterised by obstacles on roads. Our approach involves utilising data to actively detect anomalies through unsupervised methods that target local contextual anomalies. A robust ensemble for anomaly detection is proposed that improves variance reduction by using heterogeneous base detectors with different hyperparameters. Detector selection based on local competency helps identify base detectors with a conditionally low model bias.

Our future work will concentrate on updating the calibration of outlier scores by incorporating dependent loss functions, as a false negative in the C-ITS scenario might cause some troubling difficulties. We also recommend introducing automated parameter calibration in order to increase the algorithm's chances of being used by road infrastructure operators. Another essential component will be process optimisation in order to increase complexity while fine tuning the ensemble learning decision rules for efficiency enhancement. It will also be fascinating to investigate various scenarios based on actual traffic incident situations. We also suggest transforming the ELSCP approach into a tool for the real-time identification and analysis of data streams.

Author Contributions: Conceptualisation, J.C.M., S.C., C.d.R., H.F. and J.M.W.; methodology, J.C.M., S.C., C.d.R. and H.F.; software, J.C.M.; validation, J.C.M., C.d.R. and H.F.; formal analysis, J.C.M.; investigation, J.C.M.; resources, J.C.M.; data curation, J.C.M.; writing—original draft preparation, J.C.M.; writing—review and editing, S.C., C.d.R., H.F. and J.M.W.; visualisation, J.C.M.; supervision, S.C., C.d.R., H.F. and J.M.W.; project administration, S.C., H.F. and J.M.W.; funding acquisition, J.C.M., S.C., C.d.R., H.F. and J.M.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the French Embassy in Kenya.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AUC-ROC	Area under the curve of the receiver operating characteristic
AUCPR	Area under the curve of precision–recall
CAM	Cooperative awareness message
C-ITS	Cooperative intelligent transport systems
ELSCP	Enhanced locally selective combination in parallel outlier ensembles
ETSI TS	European Telecommunications Standards Institute Technical Specifications
FDM	Filter–discovery–match
FP	False positive
FPR	False positive rate
FN	False negative
GPS	Global positioning system
HBOS	Histogram-based outlier score
IForest	Isolation forest
ITS	Intelligent transportation systems
IoT	Internet of Things
KD tree	K-dimensional tree
kNN	k-nearest neighbours detector
LOF	Local outlier factor
LoTAD	Long-term traffic anomaly detection
LSCP	Locally selective combination in parallel outlier ensembles
MAD	Median absolute deviation
MCD	Minimum covariance determinant
MTTD	Mean time-to-detect
OMNET	Objective modular network testbed
P	Precision
PKI	Public key infrastructure
PyOD	Python outlier detection
PySAD	Python streaming anomaly detection
R	Recall
ROC	Receiver operating characteristic
RRCF	Robust random cut forest
RSU	Roadside unit
STORM	Stream outlier miner
SUMO	Simulation of urban mobility
TP	True positive
TPR	True positive rate
TN	True negative
VANET	Vehicular ad hoc network
V2I	Vehicle-to-infrastructure
V2V	Vehicle-to-vehicle
V2X	Vehicle-to-everything

References

1. Fahim, M.; Sillitti, A. Anomaly detection, analysis and prediction techniques in iot environment: A systematic literature review. *IEEE Access* **2019**, *7*, 81664–81681. [CrossRef]
2. Lu, M.; Türetken, O.; Adali, O.E.; Castells, J.; Blokpoel, R.; Grefen, P. C-ITS (cooperative intelligent transport systems) deployment in Europe: challenges and key findings. In Proceedings of the 25th ITS World Congress, Copenhagen, Denmark, 17–21 September 2018; p. EU-TP1076.
3. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. *ACM Comput. Surv. (CSUR)* **2009**, *41*, 1–58. [CrossRef]
4. Aggarwal, C.C. Outlier analysis. In *Data Mining: The Textbook*; Springer: Cham, Switzerland, 2015; Volume 1, Chapter 8, pp. 237–263.
5. Kamran, S.; Haas, O. A multilevel traffic incidents detection approach: Identifying traffic patterns and vehicle behaviours using real-time gps data. In Proceedings of the 2007 IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, 13–15 June 2007; pp. 912–917.

6. Zhang, M.; Li, T.; Yu, Y.; Li, Y.; Hui, P.; Zheng, Y. Urban Anomaly Analytics: Description, Detection and Prediction. *IEEE Trans. Big Data* **2020**, *8*, 809–826. [CrossRef]
7. Kong, X.; Song, X.; Xia, F.; Guo, H.; Wang, J.; Tolba, A. LoTAD: Long-term traffic anomaly detection based on crowdsourced bus trajectory data. *World Wide Web* **2018**, *21*, 825–847. [CrossRef]
8. Fouchal, H.; Bourdy, E.; Wilhelm, G.; Ayaida, M. A validation tool for cooperative intelligent transport systems. *J. Comput. Sci.* **2017**, *22*, 283–288. [CrossRef]
9. Han, X.; Grubenmann, T.; Cheng, R.; Wong, S.C.; Li, X.; Sun, W. Traffic incident detection: A trajectory-based approach. In Proceedings of the 2020 IEEE 36th International Conference on Data Engineering (ICDE), Dallas, TX, USA, 20–24 April 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1866–1869.
10. Toshniwal, A.; Mahesh, K.; Jayashree, R. Overview of Anomaly Detection techniques in Machine Learning. In Proceedings of the 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 7–9 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 808–815.
11. Goldstein, M.; Dengel, A. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012 Poster Demo Track* **2012**, *1*, 59–63.
12. Kind, A.; Stoecklin, M.P.; Dimitropoulos, X. Histogram-based traffic anomaly detection. *IEEE Trans. Netw. Serv. Manag.* **2009**, *6*, 110–121. [CrossRef]
13. Rousseeuw, P.J.; Driessen, K.V. A fast algorithm for the minimum covariance determinant estimator. *Technometrics* **1999**, *41*, 212–223. [CrossRef]
14. Hubert, M.; Debruyne, M.; Rousseeuw, P.J. Minimum covariance determinant and extensions. *Wiley Interdiscip. Rev. Comput. Stat.* **2018**, *10*, e1421. [CrossRef]
15. Rousseeuw, P.J.; Hubert, M. Anomaly detection by robust statistics. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1236. [CrossRef]
16. Liu, F.T.; Ting, K.M.; Zhou, Z.H. Isolation forest. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 413–422.
17. Chen, W.R.; Yun, Y.H.; Wen, M.; Lu, H.M.; Zhang, Z.M.; Liang, Y.Z. Representative subset selection and outlier detection via isolation forest. *Anal. Methods* **2016**, *8*, 7225–7231. [CrossRef]
18. Guha, S.; Mishra, N.; Roy, G.; Schrijvers, O. Robust random cut forest based anomaly detection on streams. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 2712–2721.
19. Zhao, Y.; Nasrullah, Z.; Hryniewicki, M.K.; Li, Z. LSCP: Locally selective combination in parallel outlier ensembles. In Proceedings of the 2019 SIAM International Conference on Data Mining, Calgary, AB, Canada, 2–4 May 2019; pp. 585–593.
20. Breunig, M.M.; Kriegel, H.P.; Ng, R.T.; Sander, J. LOF: Identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, TX, USA, 16–18 May 2000; pp. 93–104.
21. Angiulli, F.; Fassetti, F. Detecting distance-based outliers in streams of data. In Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, Lisbon, Portugal, 6–10 November 2007; pp. 811–820.
22. Tan, S.C.; Ting, K.M.; Liu, T.F. Fast anomaly detection for streaming data. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Spain, 16–22 July 2011; pp. 1511–1516.
23. Gaber, M.M. Advances in data stream mining. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2012**, *2*, 79–85. [CrossRef]
24. Calikus, E.; Nowaczyk, S.; Sant'Anna, A.; Dikmen, O. No free lunch but a cheaper supper: A general framework for streaming anomaly detection. *Expert Syst. Appl.* **2020**, *155*, 113453. [CrossRef]
25. Britto, A.S., Jr.; Sabourin, R.; Oliveira, L.E. Dynamic selection of classifiers—A comprehensive review. *Pattern Recognit.* **2014**, *47*, 3665–3680. [CrossRef]
26. Polikar, R. Ensemble based systems in decision making. *IEEE Circuits Syst. Mag.* **2006**, *6*, 21–45. [CrossRef]
27. Alghushairy, O.; Alsini, R.; Soule, T.; Ma, X. A Review of Local Outlier Factor Algorithms for Outlier Detection in Big Data Streams. *Big Data Cogn. Comput.* **2021**, *5*, 1. [CrossRef]
28. Ho, T.K.; Hull, J.J.; Srihari, S.N. Decision combination in multiple classifier systems. *IEEE Trans. Pattern Anal. Mach. Intell.* **1994**, *16*, 66–75.
29. Woods, K.; Kegelmeyer, W.P.; Bowyer, K. Combination of multiple classifiers using local accuracy estimates. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 405–410. [CrossRef]
30. Rayana, S.; Akoglu, L. An ensemble approach for event detection and characterization in dynamic graphs. In Proceedings of the ACM SIGKDD ODD Workshop, New York, NY, USA, 24–27 August 2014.
31. Zimek, A.; Campello, R.J.; Sander, J. Ensembles for unsupervised outlier detection: challenges and research questions a position paper. *ACM Sigkdd Explor. Newsl.* **2014**, *15*, 11–22. [CrossRef]
32. Fouchal, H.; Habbas, Z. Distributed backtracking algorithm based on tree decomposition over wireless sensor networks. *Concurr. Comput. Pract. Exp.* **2013**, *25*, 728–742. [CrossRef]
33. Fouchal, H.; Francillette, Y.; Hunel, P.; Vidot, N. A distributed power management optimisation in wireless sensors networks. In Proceedings of the 34th Annual IEEE Conference on Local Computer Networks, LCN, Zurich, Switzerland, 20–23 October 2009; IEEE Computer Society: Piscataway, NJ, USA, 2009; pp. 763–769. [CrossRef]
34. Salva, S.; Petitjean, E.; Fouchal, H. A simple approach to testing timed systems. In Proceedings of the FATES01 (Formal Approaches for Testing Software), a Satellite Workshop of CONCUR, Aalborg, Denmark, 25 August 2001.

35. Varga, A. The OMNeT++ discrete event simulation system. In Proceedings of the European Simulation Multiconference, Prague, Czech Republic, 6–9 June 2001; pp. 319–324.
36. Krajzewicz, D.; Hertkorn, G.; Rössel, C.; Wagner, P. SUMO (Simulation of Urban MObility)—An open-source traffic simulation. In Proceedings of the 4th Middle East Symposium on Simulation and Modelling, Berlin-Adlershof, Germany, 1–30 September 2002; pp. 183–187.
37. Riebl, R.; Obermaier, C.; Günther, H.J. Artery: Large scale simulation environment for its applications. In *Recent Advances in Network Simulation*; Springer: Cham, Switzerland, 2019; pp. 365–406.
38. *103 097 V1. 4.1*; Intelligent Transport Systems (ITS); Security; Security Header and Certificate Formats. ETSI: Valbonne, France, 2020.
39. Zhang, Z.; He, Q.; Tong, H.; Gou, J.; Li, X. Spatial-temporal traffic flow pattern identification and anomaly detection with dictionary-based compression theory in a large-scale urban network. *Transp. Res. Part C Emerg. Technol.* **2016**, *71*, 284–302. [CrossRef]
40. Leblanc, B.; Fouchal, H.; De Runz, C. Obstacle Detection based on Cooperative-Intelligent Transport System Data. In Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 7–10 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–6.
41. Moso, J.C.; Boutahala, R.; Leblanc, B.; Fouchal, H.; de Runz, C.; Cormier, S.; Wandeto, J. Anomaly Detection on Roads Using C-ITS Messages. In Proceedings of the International Workshop on Communication Technologies for Vehicles, Bordeaux, France, 16–17 November 2020; Springer: Cham, Switzerland, 2020; pp. 25–38.
42. Aggarwal, C.C.; Sathe, S. Theoretical foundations and algorithms for outlier ensembles. *ACM Sigkdd Explor. Newsl.* **2015**, *17*, 24–47. [CrossRef]
43. Dolatshah, M.; Hadian, A.; Minaei-Bidgoli, B. Ball*-tree: Efficient spatial indexing for constrained nearest-neighbor search in metric spaces. *arXiv* **2015**, arXiv:1511.00628.
44. Witten, I.H.; Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques*; Morgan Kaufmann: San Francisco, CA, USA, 2005; Chapter 4.
45. Kumar, N.; Zhang, L.; Nayar, S. What is a good nearest neighbors algorithm for finding similar patches in images? In Proceedings of the European Conference on Computer Vision, Marseille, France, 12–18 October 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 364–378.
46. Zhao, Y.; Hryniewicki, M.K. DCSO: Dynamic combination of detector scores for outlier ensembles. *arXiv* **2019**, arXiv:1911.10418.
47. Hanley, J.A.; McNeil, B.J. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* **1982**, *143*, 29–36. [CrossRef] [PubMed]
48. Boyd, K.; Eng, K.H.; Page, C.D. Area under the precision-recall curve: point estimates and confidence intervals. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Prague, Czech Republic, 23–27 September 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 451–466.
49. Campos, G.O.; Zimek, A.; Sander, J.; Campello, R.J.; Micenková, B.; Schubert, E.; Assent, I.; Houle, M.E. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Min. Knowl. Discov.* **2016**, *30*, 891–927. [CrossRef]
50. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [CrossRef]
51. Saito, T.; Rehmsmeier, M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE* **2015**, *10*, e0118432. [CrossRef]
52. Yilmaz, S.F.; Kozat, S.S. PySAD: A Streaming Anomaly Detection Framework in Python. *arXiv* **2020**, arXiv:2009.02572.
53. Zhao, Y.; Nasrullah, Z.; Li, Z. PyOD: A Python Toolbox for Scalable Outlier Detection. *J. Mach. Learn. Res.* **2019**, *20*, 1–7.
54. Mandrekar, J.N. Receiver operating characteristic curve in diagnostic test assessment. *J. Thorac. Oncol.* **2010**, *5*, 1315–1316. [CrossRef]
55. Xia, X.; Meng, Z.; Han, X.; Li, H.; Tsukiji, T.; Xu, R.; Zheng, Z.; Ma, J. An automated driving systems data acquisition and analytics platform. *Transp. Res. Part Emerg. Technol.* **2023**, *151*, 104120. [CrossRef]
56. Hajebi, K.; Abbasi-Yadkori, Y.; Shahbazi, H.; Zhang, H. Fast approximate nearest-neighbor search with k-nearest neighbor graph. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Spain, 16–22 July 2011.
57. Cruz, R.M.; Sabourin, R.; Cavalcanti, G.D. Dynamic classifier selection: Recent advances and perspectives. *Inf. Fusion* **2018**, *41*, 195–216. [CrossRef]
58. Rayana, S.; Akoglu, L. Less is more: Building selective anomaly ensembles. *ACM Trans. Knowl. Discov. Data (TKDD)* **2016**, *10*, 1–33. [CrossRef]