Local Anomaly Detection in Smart Public Transport Vehicles

Jessie Chin Bartley Liauw A Fong
jessie.liauw-a-fong@student.uva.nl

January 6, 2023, 39 pages

Academic supervisor:	Benny Akesson, k.b.akesson@uva.nl
Daily supervisor:	Jeroen Douwes, jeroen.douwes@ximedes.com
Host organisation:	Ximedes, https://ximedes.com/

Universiteit van Amsterdam
Faculteit der Natuurwetenschappen, Wiskunde en Informatica
Master Software Engineering
http://www.software-engineering-amsterdam.nl

Abstract

This thesis delves into the realm of anomaly detection within smart public transport vehicles. A domain marked by the integration of complex, heterogeneous systems in one vehicle. Emphasizing the critical role of effective data utilization, the study explores strategies for anomaly detection aimed at enhancing the operational efficiency and reliability of public transport services. These anomalies are difficult to pinpoint due to the varied and interconnected nature of the vehicle systems, ranging from hardware components like payment terminals to software for vehicle scheduling. Effective anomaly detection in such a multifaceted environment is crucial for maintaining high service standards and ensuring passenger safety. This research strategically concentrates on local anomaly detection because of the mobile nature of a SPTV. This approach allows for more nuanced and contextually relevant data analysis, considering the specific geographical data unique to each vehicle at any given time.

This research differentiates itself by focusing on the generalization of the anomaly detection process. Anomaly detection is known to be difficult to generalize. There are three challenge categories when generalizing: Performance, Normal region, and Quality. This thesis focuses on both the normal region and the quality category.

The research tackles these challenges by creating a unified data collection framework. It does this by comparing agent-based and agent-less approaches to data collection. The study underscores the significance of an agent-based method because of its adaptability and reduced complexity in integrating new modules. Furthermore, the adoption of an industry standard for data collection is explored, highlighting its benefits in terms of simplicity and portability while ensuring minimal alterations to existing systems.

The research includes an examination of three distinct local anomaly detection algorithms across various datasets derived from a specific bus line. This empirical approach seeks to determine the most effective algorithm for handling the multifaceted data inherent to SPTV. The findings reveal nuanced insights, particularly at a 25% confidence level, where mean and median algorithms demonstrate optimal performance.

Contents

Chapter 1

Introduction

The usage of technology in various sectors has been highly impactful, and the public transportation sector has been no exception. The digital revolution has significantly changed numerous processes in the industry, ranging from the optimization of route scheduling to advancements in ticketing systems. Where once a public transport vehicle (PTV) might be just another vehicle, modern variants may contain complex technologies like Internet of Things (IoT), edge computing, and a 5G network [1]. Such an evolved transport vehicle transcends the term PTV and is more aptly dubbed a smart public transport vehicle (SPTV).

The inherent complexity and diversity of SPTV make them fascinating subjects for research. Their mobile nature and assortment of components present significant challenges in achieving comprehensive control over the vehicle. Control that is crucial for tasks such as maintenance.

Predictive maintenance aims to identify the right time to perform maintenance on a particular machine [2]. This proactive approach contrasts with traditional preventive or reactive maintenance, aiming instead to intelligently use the machine's state and data for smarter maintenance. Various methodologies can be employed to implement predictive maintenance. One such method involves using statistical models to analyze machine data [3]. Another method, which will be the focus of this research, is anomaly detection.

Central to the success of these maintenance strategies is the concept of observability, the ability to discern the internal state of a system [4]. This process results in telemetry data, which can be used for analysis such as anomaly detection. The quality of such data is detrimental to the effectiveness of anomaly detection and, therefore, also predictive maintenance.

1.1 Problem statement

Anomaly detection suffers from multiple challenges, especially when the objective is its generalization across different systems or contexts [5]. Anomalies are tightly bound to the specific system in which they occur, making generalization particularly challenging compared to other observability goals like fault analysis [6]. We can broadly categorize the challenges faced when generalizing anomaly detection into three types 1. Performance 2. Normal region 3. Quality (see Section 2.3.2).

While creating a generalized method that handles all these challenges is nearly impossible, it is worth noting that performance can be somewhat compromised without entirely negating the detection of anomalies. However, without a well-defined normal region or quality, anomaly detection becomes unfeasible.

Anomaly detection presents unique challenges in the context of a SPTV. The system comprises various components, each potentially manufactured by different entities and serving specialized functions. For example, a separate entity might develop the hardware facilitating payment while another entity maintains the software overseeing vehicle scheduling. Such heterogeneity complicates assembling a unified, comprehensive view of the system's health, which is essential for effective anomaly detection. The problem lies in the data that is being collected in different ways, resulting in different data formats, locations, and quality. In short, there is no way to collect and format the data while still maintaining its quality.

In addition, anomaly detection comes in multiple forms. There are three types of anomalies that exist: 1. Point 2. Contextual 3. Collective (see Section 2.3). This research focuses on contextual anomalies, more specifically, context about the vehicle's location. This spatial context is more interesting because of the mobile nature of a SPTV. The context of a vehicle can be used to determine if observation of the state of a SPTV is an outlier. There are two ways to classify spatial context: global and local. Global anomaly detection incorporates spatial context directly as an attribute in the detection process. In contrast, local anomaly detection focuses on the neighbors of a data point. This paper will focus on local anomaly detection. Anomaly detection is very domain-specific [6]. Therefore, an algorithm that works well in one domain does not necessarily translate to another. Analyzing different algorithms in the domain of public transport can result in outcomes different from the research that is already available.

1.1.1 Research questions

To tackle the generalization of anomaly detection for a SPTV this research will answer the following questions:

- **RQ1**: What strategies can be employed to design a unified data collection framework for effective anomaly detection in an SPTV?
- **RQ2**: Which anomaly detection algorithms are most effective and well-suited for handling the data derived from SPTV?

1.1.2 Research method

This study aims to design a generic approach to anomaly detection within a specific domain, focusing on the public transport sector. The pragmatic approach adopted in this research is rooted in examining existing literature and applying action research methodology [7]. The research will be conducted in collaboration with a company operating in the public transport domain, serving as the problem owner.

Action research, while effective, presents two primary challenges: authenticity and knowledge outcomes [7]. Authenticity means that the problem researched is a real and important one that needs solving. To address the issue of authenticity, a comprehensive review of related work will be conducted (See 3). This review will clarify the research gap the proposed study aims to fill, thereby justifying the necessity and relevance of the research questions posed.

The challenge of knowledge outcomes pertains to critically evaluating the proposed solutions. To ensure the validity and appropriacy of the chosen solution, a comparative analysis of multiple potential solutions will be conducted. This approach will facilitate a rigorous evaluation process, ensuring that the selected solution is effective and contextually appropriate for the problem at hand.

1.2 Contributions

This research makes the following contributions:

1. Development of a unified framework for anomaly detection in SPTVs.
2. Evaluation of local outlier detection algorithms in the context of SPTV.
3. Detailed case study of anomaly detection implementation.

1.3 Outline

Chapter 2 provides context to the concepts and techniques used throughout the thesis. Chapter 3 discusses the work related to anomaly detection in an SPTV. Next, Chapter 4 presents how to generalize the data collection to generalize the anomaly detection process eventually. The next chapter, Chapter 5, discusses the experiments, what anomaly detection algorithms will be used, and the results. Next, Chapter 6 discusses the effectiveness of the proposed methods. Finally, chapter 7 presents the conclusion and future work, summarizing the contributions and potential directions for further research. The outline is structured to ensure a logical progression of ideas and provide a thorough understanding of the complexities and nuances of anomaly detection in an SPTV.

Chapter 2

Background

An introduction to the basic concept necessary to understand this research will be presented as background.

2.1 System

Before diving into the background of the research, it is crucial to first identify and elaborate on the specific type of system under investigation. The characteristics of this system will directly influence the scope and focus of the background information presented.

The subject of this research is a SPTV. A system qualifies as an SPTV based on several defining attributes, of which the most important is the usage of the system for public transportation, which encompasses a broad spectrum from ferries and buses to trains. In addition, the SPTV's should also be: mobile, connected to the cloud, resource-constrained, and comprised of multiple heterogeneous components.

These components are a mix of hardware devices and software services. On the hardware front, components can vary widely, from bank card validators and ticketing systems to display screens indicating the vehicle's current stop. These hardware components must communicate with each other, a task facilitated by an underlying layer of software.

To clarify each section of the background material, a sample case will be provided as a tangible illustration. In this particular example, the focus will be on a SPTV operating in an area with high-rise buildings. Because of the surrounding buildings, the SPTV can lose connection to the cloud.

2.2 Observability

Observability is a measure of how well the internal state of the services can be observable from knowledge of external outputs [4]. The foundation of observability rests on three core pillars: logs, metrics, and traces. These can be categorized collectively as telemetry data [8].

Firstly, **Logs** serve as timestamped records that capture specific events within a system. Logs don't adhere to a single format; they can be structured as plain text, JSON, or other formats. To manage the sheer volume of logs and prevent overloading the system responsible for analysis, sampling techniques are often employed. For instance, Dapper adopts a uniform sampling strategy, which randomly selects entries without repetition, ensuring each has an equal chance of being chosen [9]. A more advanced technique is attention-based sampling, which leverages attention mechanisms to account for both temporal and structural variations in traces, thus enhancing the quality of the sample [10].

Secondly, **Metrics** are a numeric representation of data measured over intervals of time [8]. These can provide insights into various aspects of the system's performance over a given period, such as CPU utilization, average response time, or disk usage.

Lastly, **Traces** are a representation of the customer journey over multiple services [8]. Tracing can occur at different levels of abstraction; for example, in distributed systems, a trace might consist of one request going through multiple services. In contrast, within an individual application, a trace could represent the sequence of operations or "journey" a function undertakes within the software.

This research will specifically concentrate on the aspect of logging [11]. Logging itself can be split into two categories: system logs and application logs. The primary focus of this research will be application logs. Essentially, they provide a record of events occurring within an application's environment and can include errors, warnings, or other informational messages.

2.2.1 Stages

Observability can be split into five distinct stages, as seen in Figure 2.1.

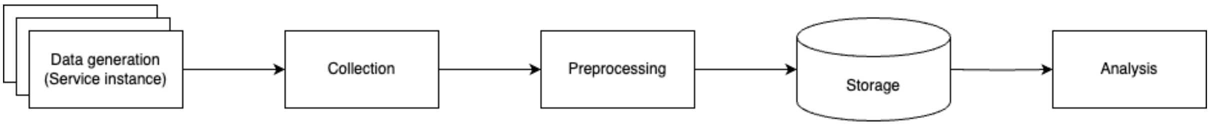


Figure 2.1: Observability stages [11]

The initial phase centers around the **generation of data** suitable for analysis. Importantly, the data produced must fall under one of the categories of telemetry data. The methodologies for generating this data can vary significantly depending on the specific type of telemetry data. When it comes to the **collection** of logs, there are two methodologies: agent-based and agent-less. An agent is a running process on the service that reads the logs files and sends them to the collector [4]. This approach allows centralized management of logs, although it introduces an additional layer to the system architecture. On the other hand, with the agent-less approach, the application sends logs directly to the collector [4]. This method tends to be simpler to manage, as it requires fewer components, but it may not provide the same level of separated concerns as the agent-based approach. After being sent to a centralized collector, the data will be **preprocessed**. The nature of this processing is determined by the intended use of the data. Preprocessing transforms the logs produced by different services into a generalized format [11]. This processed data is then sent to the **storage**, which acts as a middle-man between the data and the analysis. **Analysis** of the data can be performed as soon as the data is received in the storage. The analysis can have multiple purposes, with common applications being:

- Debugging
- Fault analysis
- Anomaly detection

2.3 Anomaly detection

One of the primary uses of telemetry data is anomaly detection. An anomaly is an observation that deviates so significantly from other observations as to arouse suspicion that it was generated by a different mechanism [6]. Anomalies can be categorized into three types [5]:

- **Point anomaly**: the simplest type of anomaly in which the anomaly detector can analyze each data point without considering any other data points or other information in the input dataset. For example, the SPTV loses cloud connection.
- **Contextual anomaly**: anomalies where the context is also analyzed to determine if a data point is an anomaly. For example, the SPTV loses cloud connection in a certain geographical area.
- **Collective anomaly**: anomalies where not only one data point is analyzed but a collection of data points. For example, the SPTV loses cloud connection whenever two passengers check out simultaneously.

Anomaly detection algorithms can process different types of input data; data can be classified as univariate or multivariate. Univariate data focuses on a single variable or attribute. For example, if one were to monitor cloud connection of a SPTV over time, the resulting dataset would be univariate, as anomalies would be detected based solely on the variations in that single parameter.

On the other hand, multivariate data encompasses multiple attributes or variables. For example, the cloud connection of a SPTV and its spatial context. Anomalies in a multivariate context can emerge from unusual combinations of these variables, even if the individual values might appear normal. Multivariate data can be represented in several forms:

- **Vector**: A one-dimensional array of data representing multiple attributes for a single instance. For example, the status of each hardware component of a SPTV.

- **Matrix**: A two-dimensional grid of data, often representing multiple attributes across multiple instances. For example, the status of each hardware component of multiple SPTV's.
- **Tensor**: A multi-dimensional array extending beyond matrices. Tensors can represent data with more than two axes.

The selection of an algorithm for anomaly detection influences the kind of data that can be used. Anomaly detection techniques primarily fall into two categories: machine-learning models and statistics-based models [5]. Within machine learning models, there are three types of data classification levels. Classification means that the data is labeled with the expected outcome of the machine-learning algorithm. The different classifications can be summarized as:

- **Supervised**: All data is labeled
- **Semi-supervised**: Some data is labeled, and others are not.
- **Unsupervised**: None of the data is labeled.

The characteristics of the data influence which types of machine learning algorithms can be used. For example, if the data is supervised, a recursive neural network can be implemented [5]. However, other algorithms come to mind if the data is unsupervised, such as K-nearest neighbour (KNN) [5].

This does not apply to statistic-based models. They can be divided into parametric models and non-parametric models. With parametric models, the data is sampled from a known distribution [5]. The training phase estimates the parameters of the model. In non-parametric models, the parameters of the models are not estimated and calculated based on the new data point presented [5].

Spatial context

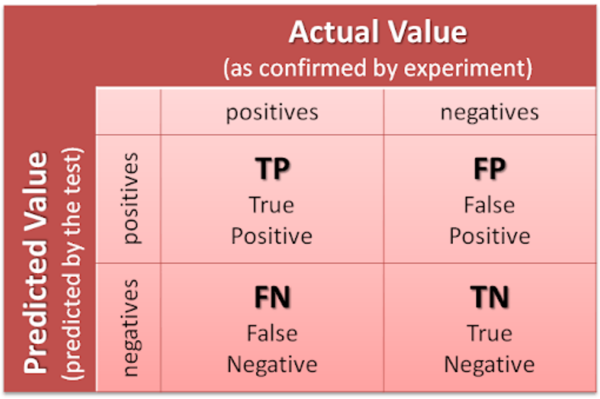
Contextual and collective anomalies factor in the context in addition to the data point itself. This context is typically classified into three types: spatial, temporal, and spatial-temporal. Although various anomaly detection methods cater to these different contexts, there's a gap. For instance, in the realm of spatial context, many methods predominantly focus on either the dataset's non-spatial attributes or strictly on spatial relationships, sidelining the correlation between the two [12]. In our example case, anomaly detection can be done without using spatial context, like identifying hardware-related issues, such as when a bankcard validator in the SPTV stops working, without considering the spatial context. In addition, anomaly detection can also be done purely based on spatial context, like when the PTV is in an off-route location.

Spatial context can be used in two ways during the anomaly detection process. Firstly, global anomaly detection incorporates spatial context directly as an attribute in the detection process. Here, anomalies are identified by comparing a new data point against the entire dataset. On the other hand, local anomaly detection focuses on the neighbors of a data point. Rather than comparing a new data point against the whole dataset, it evaluates it in relation to its immediate spatial neighbors. Spatial context doesn't strictly apply to geographical locations. It can also apply to locations based on different distances to its neighbors, for example, using the KNN algorithm, which can use any attribute to calculate the neighborhood of a data point.

Think of connectivity in a city as an example. Certain city spots may be notorious for weak connections due to tall buildings. Losing connection in such a zone wouldn't be unexpected. Yet, if you assessed that location against the entire city, it might appear as an anomaly since most areas might have strong connectivity.

2.3.1 Evaluation of anomaly detection

Anomaly detection functions as a type of binary classification [13], the performance of which can be effectively represented through a confusion matrix. A confusion matrix is a tabular summary showing how well the model performs [14]. See figure 2.2



		Actual Value (as confirmed by experiment)	
		positives	negatives
Predicted Value (predicted by the test)	positives	**TP** True Positive	**FP** False Positive
	negatives	**FN** False Negative	**TN** True Negative

Figure 2.2: Confusion matrix

Various methods exist for analyzing a confusion matrix, with the Matthews correlation coefficient (MCC) and the H-measure cited as the most reliable evaluation metrics [15].

2.3.2 Challenges in anomaly detection

An anomaly is a data point that deviates from the normal region, also called the normal behavior [5]. However, defining the normal region is a complex task. All data points outside this normal region can be characterized as anomalous. The normal region is one of the three challenge categories that can occur when generalizing anomaly detection. The other two challenge categories are performance and quality;

- **Performance**: challenges are based on the performance restrictions of a certain system.
- **Normal region**: challenges are based on defining and maintaining the normal region.
- **Quality**: challenges are based on providing a certain quality measured in accuracy.

Table 2.1: Challenges of generalizing anomaly detection grouped by category [5]

Category	**Challenge**
Performance	Availability, reliability, low latency: There must be a trade of between low latency and reliability
Performance	High throughput, parallelization, distribution, and scalability
Normal region	Defining the normal region
Normal region	Detecting malicious actions: They adapt themselves to mimic the normal behavior of a system
Normal region	Evolution of normal behavior
Quality	Different domains require different anomaly notions
Quality	Lack of labeled data
Quality	Distinguishing noise from anomalies

This research does not aim to tackle all categories because of the time restrictions. This paper will encompass a generalized solution for the quality category. In addition, defining the normal region is also at the core of the research. Anomalies are only as good as the normal region, meaning that the initial definition is essential to the generalization.

2.4 Public transport

This research's primary domain of focus is the public transport sector, where the anomaly detection algorithms will be tested and evaluated using data from this sector. The choice of public transport is strategic as the sector inherently meets the spatial context criteria essential for the anomaly detection algorithm. Additionally, the generalization of anomaly detection is based on the data that can be retrieved from the source.

2.4.1 Standardization

The public transport sector has witnessed significant technological advancements like many other domains. A common trend accompanying such progress is the move towards standardization to ensure communication across different parties. This trend is also seen in public transport, with various standards emerging to address different facets of the domain. For instance, Transmodel [16] is a reference data model for public transport, providing a standardized method to describe public transport concepts and data.

On the practical front, schedules for PTV's are also subject to standardization. Several standards aim to streamline this, with Google's General Transit Feed Specification (GTFS) [17] being a notable example. In Europe, the Committee for Standardization (CEN) [18] attempts to create standards across various sectors. Their rendition for schedule standardization is named Network Timetable Exchange (NeTEx) [19].

Moreover, there's a focus on realtime communication standards in public transport. An enhancement to GTFS, named GTFS-Realtime [20], relays realtime data such as vehicle positioning and service alerts. Parallelly, CEN has developed a realtime framework known as Service Interface for Real-time Information (SIRI) [21].

Pivoting to in-vehicle communication, as SPTV systems evolve and incorporate more features, the communication between different systems becomes more apparent. Different manufacturers might create these systems, so a universal communication standard becomes essential. Two primary contenders in this space are Information Technology for Public Transport (ITxPT) [22] and Der Verband Deutscher Verkehrsunternehmen (VDV) [23]. While VDV is created primarily for the German public transport sector, ITxPT has wider adoption across various countries.

2.4.2 ITxPT

The processed data that results from preprocessing is what gets fed into the anomaly detection algorithm. It must contain enough information for the algorithm to function effectively. To guarantee that the data meets these requirements, it needs to be of a certain quality. While a standard cannot directly ensure data quality, it does establish clear criteria for the type and format of data to be delivered. It becomes feasible to assess whether the provided data meets the information requirements.

ITxPT is a standard introduced for public transport in Europe. It enables an open architecture, data accessibility, and interoperability between IT systems [22]. ITxPT consists of architectural requirements and a set of communication specifications. It's divided into various modules. Modules represent either a software or hardware component.

The architectural requirements require that the SPTV must possess a minimal set of features to facilitate communication between different systems, such as the CAN network, DHCP, or TCP setup. While this research will utilize some of these technologies, their selection is not directly relevant to the data collection process.

Additionally, the architectural requirements define the specific modules that must be available, and these modules adhere to a standard protocol for communication which outlines the guidelines for interfacing with the modules. The modules required by ITxPT are as follows:

- Inventory
- Time
- GNSSLocation
- FMStoIP
- VEHICLEtoIP
- AVMS Automatic Vehicle Monitoring System
- APC Passenger Counting
- MADT Multi-Application Driver Terminal
- MQTTbroker

The data generated by these systems can be employed for detecting anomalies. Additionally, ITxPT offers the functionality to register custom modules, where the data can vary per module. What remains consistent across all modules is the standardized state, meaning each module specifies its possible states and the current state it is in.

Finally, the ITxPT standard outlines a method for subscribing to data updates from each individual module. This provides a consistent way to receive the latest information from the various components within the system.

2.4.3 GIVA

An example of a system that implements the ITxPT standard is Generic ICT Vehicle Architecture (GIVA). GIVA is a generic hard- and software layer for public transport vehicles, based on European standards (ITxPT / VDV 301) [24]. The system is used by gemeentevervoerbedrijf (GVB), the public transport provider for the municipality of Amsterdam. Every workday, around 800.000 people use their services [25]. Their fleet includes ferries, buses, trams, and metros, all available to the public. These vehicles are considered to be SPTV since they are composed of multiple IT systems that allow for usable and optimized public transport.

GIVA is composed of a lot of different components. Broadly, these components fall into four categories:

- **Service**: These are the software elements that run on the SPTV of GIVA. The software has a microservice architecture where each service has a distinct functionality in the system.
- **Cloud**: GIVA's cloud components facilitate communication between other vehicles in the fleet or the operational center of the GVB.
- **Vehicle Interfaces**: Supplied by the vehicle manufacturers, these interfaces offer vital insights about the vehicle, from speed to other operational metrics.
- **Third-Party Components**: These can either be software or hardware components that are created by a third party. Examples are pin terminals, chip validators, or information screens.

Figure 2.3 shows the implementation of each service.

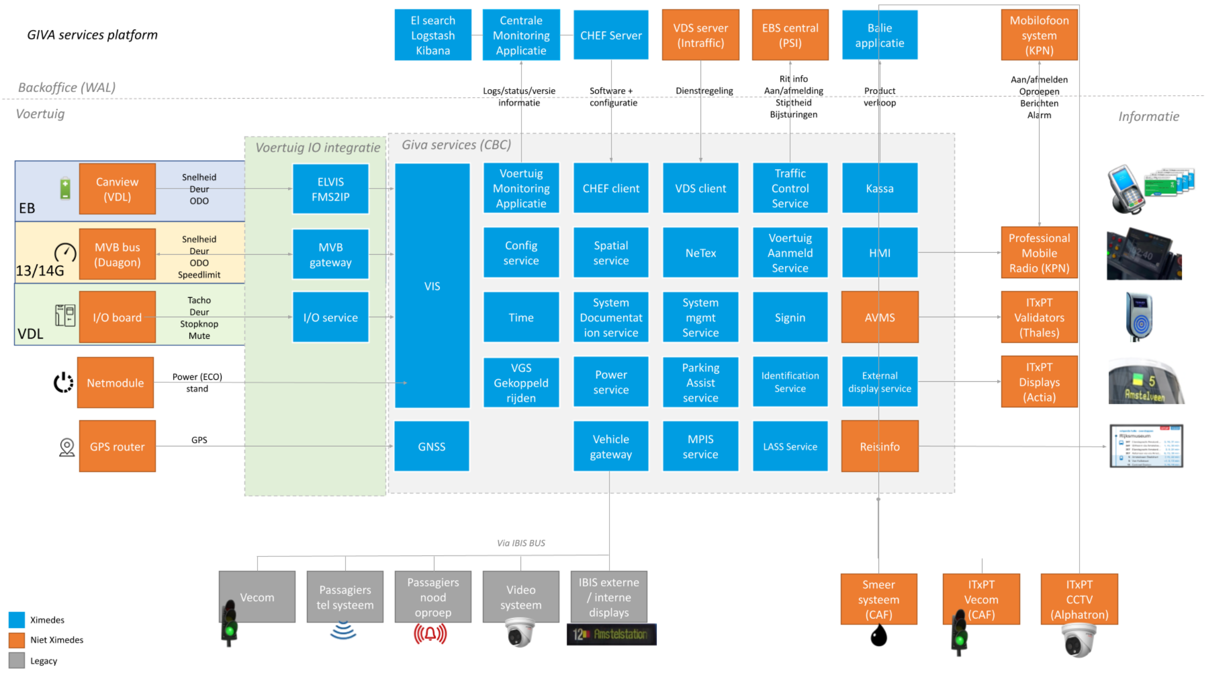


Figure 2.3: Current architecture of the GIVA system

Chapter 3

Related work

We divide the related work into observability and anomaly detection.

3.1 Observability

Observability is implemented in multiple sectors, showing the interest and usability of this practice.

3.1.1 Data collection

Observability has been recognized as a crucial practice across different domains, although its implementation varies depending on the specific requirements of each domain. Data collection is necessary for anomaly detection. While the concept of anomaly detection is agnostic to the type of data involved, different data types often result in different collection methods.

In the context of cloud microservices, Jamshidi *et al.* explores various data collection approaches, emphasizing that traces can be gathered through techniques such as Sidecar or Service Mesh [26]. These methods involve a second container that runs next to the main container through which network traffic is routed. Liu *et al.* propose an alternative approach involving Extended Berkeley Packet Filter (eBPF), which modifies the kernel environment of a Docker container, thus eliminating the need for a secondary container [27].

When it comes to log data, Li *et al.* outlines two principal methods for sending logs to a collector: manual coding and dynamic binary instrumentation [11]. Two approaches to transmitting logs are agent-less and agent-based (see 2). Zahid *et al.* proposes Security Information and Event Management System (SIEM) as an agent-based method [28]. Shengyan *et al.*, on the other hand, delves into a multi-agent approach where each agent possesses a distinct role in the log collection process [29]. Turnbull further elaborates on LogStash [31], a prominent open-source, agent-based log collection tool, outlining its implementation nuances [30].

In terms of metric collection, both Levin and Benson and Amaral *et al.* employ eBPF for different facets of microservice performance. While Levin and Benson focuses broadly on various metrics [32], Amaral *et al.* their work is explicitly tailored to performance metrics in microservice systems [33]. Brondolin and Santambrogio introduces a black-box approach focused on measuring the performance in a microservice system [34].

In contrast, this research will aim to collect data on IoT devices and send it to the cloud with an agent-based approach. This research aims to advance existing concepts and refine them to meet the specific demands of the public transport domain.

3.1.2 Smart public transport

SPTV is not a new concept. The usage of technology in public transport is getting more and more popular [35]. Technology in public transport can be used for multiple purposes. There are three main reasons technology will be used in combination with public transport: commuters, safety, and transportation of goods [35]. GIVA was created to improve all of these. Murad *et al.* created a GIVA-like system in Jakarta for the same reason [36]. Paolino has researched predictive maintenance for a Spanish public transport company [37]. Their results were very specific for the company. This contradicts our solution, which is focused on generalizing the anomaly detection process. Paolino did note that using standards, specifically ITxPT, can improve the portability of the predictive maintenance solution.

In contrast, this research will focus on anomaly detection as a form of analysis that can be done on the telemetry data. Additionally, it will integrate insights and methodologies derived from other smart public transport solutions, enriching the overall approach to the anomaly detection process.

3.2 Anomaly detection

Because anomaly detection is so domain-driven, it is hard to generalize. However, a lot can be learned from anomaly detection in different domains.

3.2.1 Anomalies in different domains

Li *et al.* explores three methodologies for analyzing telemetry data, anomaly detection being one of them [11]. Extending this discussion, Cook *et al.* conducted an in-depth survey focusing on the role of anomaly detection in IoT, specifically within the context of time-series data [6]. Their work addresses how different anomaly detection algorithms can be effectively implemented across varying IoT environments. Further broadening this discussion, Shaukat *et al.* examine a more comprehensive array of algorithms tailored for time-series anomaly detection in IoT [5].

Shifting the focus toward microservices, Du *et al.* delves into anomaly detection in metric data within Docker containers, an essential component in microservice architectures [38]. Complementing this, Pahl and Aubet offers machine learning-based solutions specifically aimed at IoT systems [39].

Network behavior is a common topic in anomaly detection research. Mehdi *et al.* investigates user home networks to identify potential anomalies [40], while Lee *et al.* introduces a scalable, generalized framework for anomaly detection within network systems [41].

In relation to the core challenges of anomaly detection outlined in 2.3.2, defining the 'normal' region remains a primary concern. To address this, Chen *et al.* has engineered a system capable of transferring the criteria for anomaly classification from one system to another [42].

3.2.2 Anomaly detection in transportation

Anomaly detection has been applied to many different domains. One characterization that is not often analyzed is the mobility of the system. Domains that share this feature all share a common theme: transportation. For example, Caetano *et al.* researched visual-spatial anomaly detection for autonomous vehicles [43]. In the same domain, Bogdoll *et al.* used data from lidar and radar to implement anomaly detection [44]. Meanwhile, Dixit *et al.* took a more generic approach to transportation and used anomaly detection techniques for identifying malicious activities [45].

Most newer means of transportation have an internal network in the vehicle. This internal network has been the topic of multiple anomaly detection implementations. Lin *et al.* has analyzed multiple in-vehicle networks for malicious activity [46]. Taylor *et al.* has taken a more generalized approach to the same problem by analyzing a CAN bus, a network architecture used in multiple vehicles [47]. Another paper by Narayanan *et al.* researched a different general architecture named ECU [48]. Finally, Narayanan *et al.* researched multiple connected vehicles for anomaly detection [48].

In addition, there are also instances of anomaly detection in SPTV's. For example, Kang *et al.* researched the break operation patterns of a metro [49]. While Maskey *et al.* uses blockchain in outlier detection in public transport [50].

In contrast, this research will focus on a more specific public transportation ecosystem domain and create a generalized approach.

3.2.3 Spatial context

Most research on spatial anomaly detection assumes that univariate data is used. Kou *et al.* research into monitoring the West Nile virus spread across U.S. states showcases that these anomalies can be caught by using univariate spatial anomaly detection [51]. Further, Shekhar *et al.* formulated specialized algorithms tailored for univariate data applications [52]. Complementing these spatial insights, Izakian and Pedrycz integrated temporal dimensions, offering a new contextual attribute to search for neighbors [53].

However, the problem gets more complicated when looking at multivariate data since the data analyzed will be more complex (see 2.2). Lu *et al.* introduced two algorithms that could be applied to multivariate data [54]. Multiple papers were created to expand upon this work. For instance, Singh and Lalitha proposed an algorithm integrating the location quotient [55]. Hayes and Capretz sharpened the focus on multivariate data for stationary sensor anomalies [56], while Alvera-Azcárate *et al.* leveraged weather and satellite data [57]. Cheng and Li discusses a hybrid approach to spatial context, including the temporal aspect [12]. Lastly, in the realm of big data, Alghushairy *et al.* unveiled an approach specifically designed for massive data streams [58].

In addition, a couple of weighted methodologies were also introduced with Kou *et al.* advocating for assigning weights to non-spatial attributes [59]. Harris *et al.* furthered this concept by comparing the weighted methods with global anomaly detection strategies [60].

In contrast, this research will aim to create a generalized anomaly detection method by focusing on the public transport domain. Drawing from the foundational work of Lu *et al.* and the advancements made by Singh and Lalitha, the study will implement these algorithms to suit the specific needs of the public transportation domain.

Chapter 4

Data collection and preprocessing

A challenge in generalizing anomaly detection is that the implemented solutions are mostly domain-specific. The data that is being collected needs to be formatted and located in the same place while maintaining its quality. This chapter aims to provide a guide in meeting these goals while still generalizing for the public transport domain.

The telemetry data generated by the observability process can serve multiple purposes. One of the primary use cases for analyzing this type of data is anomaly detection [11]. Telemetry data can be categorized into three distinct types:

- Logs
- Traces
- Metrics

Anomaly detection applies to all data types since each type carries unique information. In the effort to standardize anomaly detection, it's essential to standardize the data type, therefore this research will focus on logs. This is important because the success of the selected anomaly detection algorithm is directly tied to the structure of the data. Analysis of telemetry data can be broken down into five stages (see figure 4.1). This chapter will focus on the collection and preprocessing of the data.

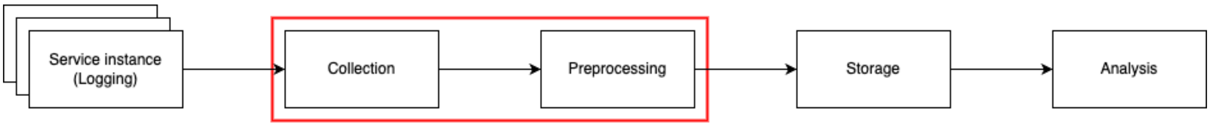


Figure 4.1: Stages of observability with focus on collection and preprocessing [11]

4.1 Collection

Data collection can be split into two approaches: agent-based and agent-less. An agent is a process running next to the application which sends the logs to the collector. In the case of a SPTV, this would be a program running on the vehicle. Another approach is the agent-less approach, which means that systems send their logs directly to the collector.

As with any two options, no solution is better than the other. However, there is a solution that will work better with the requirements of a SPTV. The collection process should adhere to the requirements specified in Table 4.1. These requirements can be mapped per approach. Table 4.2 compares the two options per requirement.

Based on the comparison in table 4.2, the best approach would be the agent-based approach since this is the only option that supports third-party integration and does not increase in complexity once more modules get added, especially if the SPTV implements a standard way of interfacing with each system, such as ITxPT.

Table 4.1: Requirements for data collection with as primary goal anomaly detection in an SPTV

Requirement	Description
Minimal integration	An SPTV can comprise numerous systems, sometimes over 100. Each system's additional development time should be minimal, ideally nonexistent.
Simple implementation	Due to the many interconnected modules, the collection's implementation should be straightforward.
No data loss during connection loss	As an SPTV is mobile, it might occasionally lose network connectivity. Such interruptions shouldn't result in data loss, which could adversely affect the anomaly detection algorithm.
No interference with primary processes	Data collection shouldn't disrupt primary processes. Anomaly detection is not considered a primary process. The data collection process should be as decoupled from main operations as much as possible.
Third-party integration	Not all systems in a SPTV have been developed by the same company. In the case of GIVA, most software is developed by Ximedes. However, none of the hardware components are developed by Ximedes.

Table 4.2: Comparison of Agent-based and Agent-less Data Collection Methods

Requirement	Agent-based	Agent-less
Minimal integration	Integration will be in infrastructure which has to be setup once per system [11].	Requires each microservice to have its integration of the data collector [11].
Simple implementation	Depends on available tools. Integration is straightforward if a common standard for interfacing with the system is present. However, if this is not the case, the configuration per system will be more complex.	Implementation is straightforward, as systems only need to send logs.
No data loss during connection loss	Agents can retry or temporarily save logs, sending them once connectivity is restored.	Each system's logging implementation must support retries and temporary log storage, possible through a language-specific library.
No interference with primary processes	Minimal impact as the agent operates separately, reading existing data.	Potential interference as systems might require modifications to support logging.
Third-party integration	Viable if logs are saved in a known, accessible location or interfacing with third-party system logs is possible.	Feasible only if third parties incorporate logging. Given multiple third-party parties, this is impractical.

4.2 Preprocessing

Preprocessing for a generalized approach means that the data input needs to be known and contain enough information to do anomaly detection. There are multiple ways in which enough information can be obtained. However, a known data structure over multiple systems means that some kind of standard needs to be implemented. There are two choices, choosing a known standard or creating a specialized standard for anomaly detection in the public transport domain. The requirements for the preprocessing are listed in Table 4.3.

The primary distinctions between the choices lie in data quality and ease of implementation. The

Table 4.3: Requirements for data preprocessing with as primary goal anomaly detection in an SPTV

Requirement	Description
Standardized input	Preprocessing demands a consistent data structure. Without standardized input, the preprocessing steps would need custom adjustments for each data type, complicating the process.
Data quality	Quality of preprocessing output is contingent upon the quality of input. Only inputs of sufficient quality can ensure reliable and meaningful preprocessing results.
Portability	To facilitate generalization in anomaly detection, preprocessing should be adaptable across various systems within the domain.
Ease of Implementation	The standard should be straightforward to integrate across various systems, ensuring broad implementation without limiting the range of systems analyzed.

Table 4.4: Comparison of domain and custom standards for data preprocessing, targeting anomaly detection in an SPTV

Requirement	Domain standard	Custom standard
Standardized input	Ensures consistent input format	Ensures consistent input format
Data quality	Quality varies between standards. Only when inspecting a standard closely can the quality be assured	Quality can be predefined and controlled
Portability	Applicable across organizations using the domain standard	Limited to organizations adopting the custom standard
Ease of Implementation	Given that the systems within a SPTV are tailor-made for the public transport domain, the likelihood of them adhering to a standard format is significant.	Implementing a custom format across all systems is impractical and poses challenges.

data quality of a domain standard varies depending on the specific one chosen, while a custom standard grapples with the challenge of needing implementation across various systems. Notably, a SPTV may comprise several third-party systems, necessitating them to adopt this custom standard. Yet, these third parties are inclined to implement an industry-recognized standard over a custom one. Consequently, a domain standard is a more practical choice.

Nonetheless, assessing the available domain standards in the public transport sector is essential. This evaluation will ensure the chosen standard meets the requisite data quality benchmarks. Numerous standards cater to public transport, each with its own merits.

As highlighted in 2.4.1, the public transport sector presents several standardization options. Focusing exclusively on in-vehicle systems, the primary contenders are ITxPT and VDV. Given that VDV predominantly serves the German market and this thesis aspires to offer solutions applicable to multiple countries, ITxPT is the logical choice. Nevertheless, simply choosing ITxPT doesn't assure the quality of the data for anomaly detection purposes. Creating a sample data format is a good indicator of potential quality. The ensuing evaluation will rely on the information drawn from this sample.

4.2.1 ITxPT example

The standard modules within the ITxPT framework include data that may be relevant to the anomaly detection process. Figure 4.2 illustrates the specific data that is of interest in this context during the process of anomaly detection:

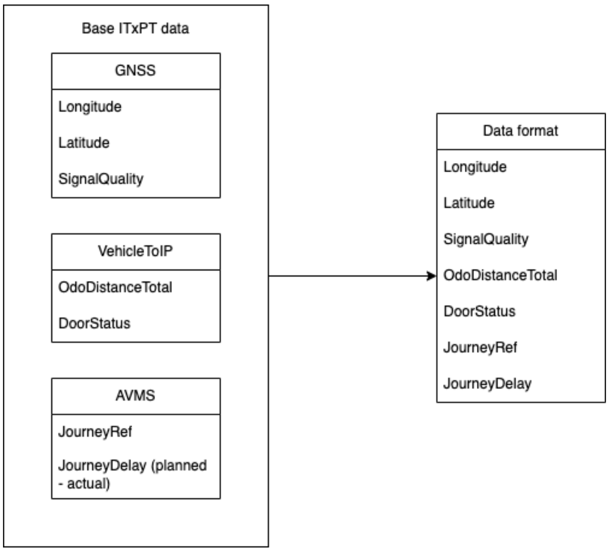


Figure 4.2: Data structure based on the standard ITxPT modules

The most important data can be retrieved from the Global Navigation Satellite System (GNSS) service. This service will provide the location of the SPTV. Unlike many other systems, a SPTV is mobile, making its location a crucial element in the anomaly detection process. Since the system can be in different states at different locations, the location information can directly influence if a new data point is anomalous.

While the data from the GNSS service is crucial, it alone is not enough for anomaly detection. In addition to the standard data, ITxPT offers the option to communicate the statuses of custom modules. These custom modules can provide additional information necessary for the anomaly detection process. The data model incorporating both standard and custom modules will be structured as follows:

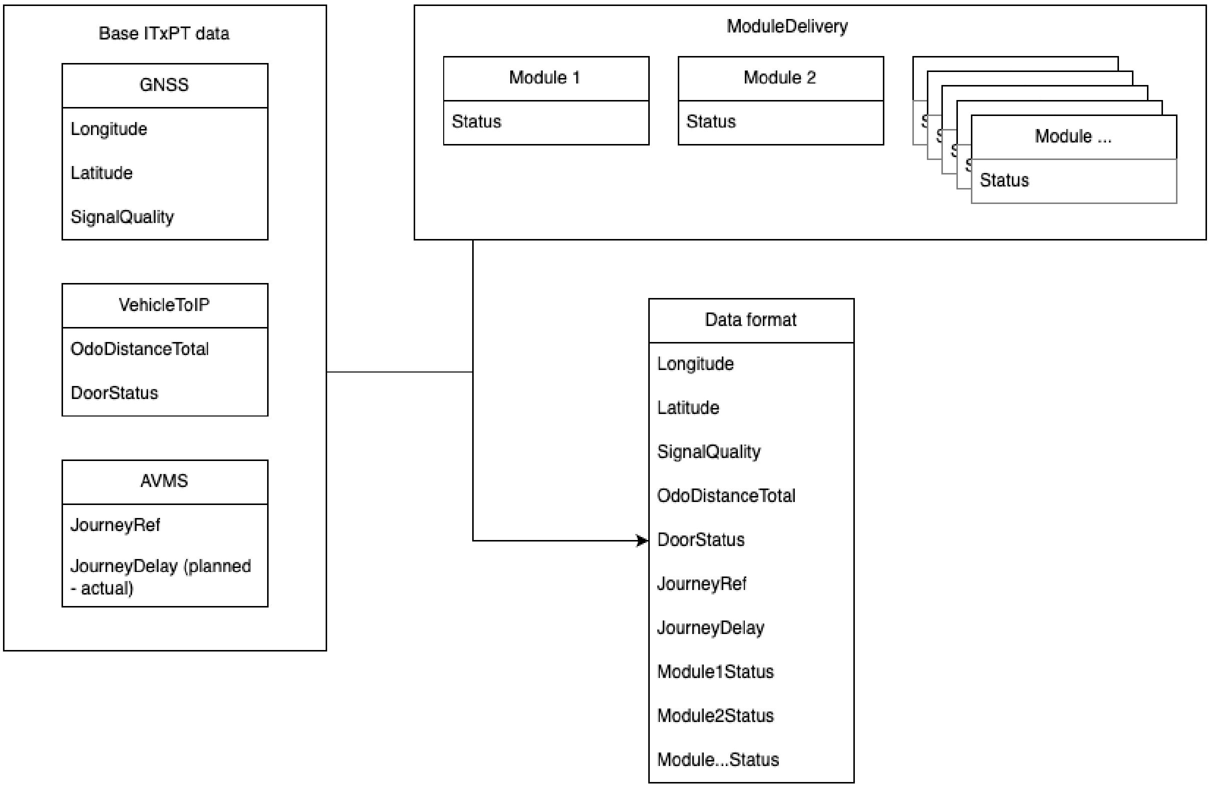


Figure 4.3: Data structure based on the standard ITxPT modules and custom modules

This will be the final data model, which will be stored and fed to the anomaly detection algorithm.

The size of the model depends on the number of custom modules that are present in the SPTV.

4.2.2 Conclusion

Standardization in anomaly detection hinges on the availability of a comprehensive and well-structured data model. Through qualitative reasoning, this study defines the data model depicted in Figure 4.3 as a robust enough data source, as it provides the entire vehicle state while still adhering to the standards provided.

Moreover, the integration of custom modules in the ITxPT standard brings flexibility to the data model. Such flexibility ensures that the model is not static but can be modified to align with new developments and requirements within the different environments of SPTV. This ability to adapt is critical, as it guarantees the continued relevance and precision of anomaly detection processes even as the SPTV evolves.

For a detailed examination of every attribute and element within this data model, readers are directed to the comprehensive list provided in the Appendix A.1.

Chapter 5

Data processing

This section focuses on the detailed examination of three anomaly detection algorithms: location quotient [55], mean [54], and median [54]. These algorithms have been selected to address the specific spatial characteristics of an SPVT. The experiments will be conducted over three datasets: synthesized GNSS Data, real GNSS Data, and general data anomaly detection. The results will be evaluated using the MCC and H-Measure [15].

5.1 Data sets

An experiment was set up using the data collection method described in chapter 4. The data was stored in a repository known as a collector, spanning the period from Monday 1^{st} May, 2023 to Saturday 1^{st} July, 2023. This data was divided into two distinct groups: a 'test set' and a 'neighbors set,' with the partitioning date being Thursday 1^{st} June, 2023. The test set was used for detecting anomalies, while insights into these anomalies were gained through the neighbors set. Before any analysis, attributes that all had the same values were removed since they did not contribute to anomaly detection. A specific bus or train line is exclusively focused on for simplicity. The specific line is chosen randomly. The same methodology could be applied to different lines. For now, the line that has been chosen is the bus line 21.

5.2 Algorithms

Three algorithms are considered for the anomaly detection process. These algorithms have been compared in previous research. However, as highlighted in chapter 2, the domain greatly influences the efficiency of the anomaly detection process [5]. The algorithms under consideration are:

- Location quotient [55]
- Mean [54]
- Median [54]

While the exact algorithms are assessed by Singh and Lalitha, the context in their research differs, as it revolves around crime data in India [55]. Moreover, their dataset emphasizes stationary data, whereas the mobility inherent to a SPTV might influence the outcome.

These particular algorithms were selected due to their similar implementation techniques, facilitating streamlined testing and generalization. Due to the time limitations imposed on this research, the exploration was restricted to these three algorithms. The explanation of each algorithm will not be part of this research since it does not aim to change the algorithms. However, it does aim to compare these algorithms in a new context.

5.2.1 Neighbors

One of the intriguing elements of anomaly detection algorithms involves the collection of neighbors. While determining the closest neighbor based on spatial context is straightforward, deciding the optimal number of neighbors is less so. There are several methods to approach this challenge:

- **Limiting by Distance**: One method involves restricting neighbors based on their spatial proximity to the data point. The benefit is that the selected neighbors are close in space. The downside is that there's no assurance that neighbors will exist for a given data point. This absence could be considered an anomaly, but defining an exact "neighborhood distance" becomes problematic.
- **Limiting by Count**: Another approach is to set a predefined number of neighbors to be collected, sorted but not limited by their distance. While this eliminates the need for arbitrary distance metrics, it may pair a data point with neighbors that are inappropriately distant, possibly resulting in irrelevant data.
- **Hybrid Method**: A combination of the two previous methods could also be employed, whereby neighbors are filtered based both on a maximum allowable distance and a maximum count to avoid overwhelming the algorithm.

Due to the computational intensity of evaluating anomalies, there's no one-size-fits-all solution to this dilemma. Given time constraints, the decision was made to limit the number of neighbors to 35 (See Section 6.2).

5.3 Experiments

Three distinct data sets will be evaluated in this study to thoroughly assess the capabilities and limitations of different anomaly detection algorithms. The first is the "All Data Set," which covers the full spectrum of available data, as elaborated in Section 5.1. The second is the "Unknown Check Data Set." This set is a subset of the "All Data Set" and focuses on a known specific anomaly message with no known cause. The aim is to use the comprehensive vehicle state data, coupled with its geographical location to find the root cause of this anomaly. Lastly, the "Synthetic Data Set," is designed to put the algorithms to the test. This synthetic data set is manually created according to the example given in Section 2.1. This set consists of 50 data points and incorporates six different types of anomalies. Given its design, the anomaly detection algorithms are expected to exhibit a high detection rate for this set.

The types of anomalies injected into the Synthetic Data Set are as follows:

1. Signal quality is 'NOK' while 'OK' is the normal
2. Signal quality is 'OK' while 'NOK' is the normal
3. The signal quality is 'NOK,' mirroring **most** of the neighboring vehicles
4. Similar to the third type, but with an additional GNSS service being unavailable.
5. The FMS service is turned off, which could potentially influence the GNSS service.
6. GNSS is turned off in most neighboring vehicles but not in all, with an additional, random service also being deactivated.

5.4 Evaluation

Two of the three datasets utilized in this study are unsupervised, making it challenging to definitively classify whether a detected anomaly is indeed an anomaly. To address this issue, this paper engages the team lead of the GIVA project, an expert familiar with the dataset, to determine the accuracy of the identified anomalies. This evaluation process is resource-intensive, requiring both the team lead and the researcher to meticulously assess each data point and decide whether it qualifies as an anomaly.

Once the data points have been evaluated and appropriately labeled, a confusion matrix will be constructed to provide an empirical basis for the algorithm's performance. This matrix will then be analyzed using specific evaluation metrics, namely the MCC and the H-measure, as outlined in Section 2.3.1. These metrics are specifically designed to evaluate binary classification algorithms and will offer valuable insights into the effectiveness of the anomaly detection process.

5.5 Results

This chapter will describe the results of the experiments.

5.5.1 Confusion matrix

This section will contain all confusion matrices for each algorithm implemented and their experiments per set as explained in Section 5.3. First, all of the confusion matrices are introduced (see Figure 2.2). These matrices are used to create the MCC and H-score as described in Section 2.3.1.

Synthetic data

Displayed here are the confusion matrices for the "Synthetic data". To the left of each row of matrices, the confidence level applied for that particular run is indicated. Above each confusion matrix, the name of the algorithm used is specified.

100%

Location quotient

		Predicted	
		Yes	No
Actual	Yes	5	0
	No	1	44

Mean

		Predicted	
		Yes	No
Actual	Yes	6	0
	No	0	44

Median

		Predicted	
		Yes	No
Actual	Yes	6	0
	No	0	44

75%

Location quotient

		Predicted	
		Yes	No
Actual	Yes	5	0
	No	1	44

Mean

		Predicted	
		Yes	No
Actual	Yes	6	0
	No	0	44

Median

		Predicted	
		Yes	No
Actual	Yes	6	0
	No	0	44

50%

Location quotient

		Predicted	
		Yes	No
Actual	Yes	5	0
	No	1	44

Mean

		Predicted	
		Yes	No
Actual	Yes	6	0
	No	0	44

Median

		Predicted	
		Yes	No
Actual	Yes	6	0
	No	0	44

25%

Location quotient

		Predicted	
		Yes	No
Actual	Yes	6	0
	No	0	44

Mean

		Predicted	
		Yes	No
Actual	Yes	6	0
	No	0	44

Median

		Predicted	
		Yes	No
Actual	Yes	6	0
	No	0	44

10%

Location quotient

		Predicted	
		Yes	No
Actual	Yes	6	0
	No	0	44

Mean

		Predicted	
		Yes	No
Actual	Yes	6	0
	No	0	44

Median

		Predicted	
		Yes	No
Actual	Yes	6	0
	No	0	44

All data

Displayed here are the confusion matrices for the "All data". To the left of each row of matrices, the confidence level applied for that particular run is indicated. Above each confusion matrix, the name of the algorithm used is specified.

100%

Location quotient

		Predicted	
		Yes	No
Actual	Yes	5	18
	No	0	27

Mean

		Predicted	
		Yes	No
Actual	Yes	1	22
	No	0	27

Median

		Predicted	
		Yes	No
Actual	Yes	1	22
	No	0	27

75%

Location quotient

		Predicted	
		Yes	No
Actual	Yes	5	0
	No	18	27

Mean

		Predicted	
		Yes	No
Actual	Yes	5	0
	No	18	27

Median

		Predicted	
		Yes	No
Actual	Yes	5	0
	No	18	27

50%

Location quotient

		Predicted	
		Yes	No
Actual	Yes	5	0
	No	18	27

Mean

		Predicted	
		Yes	No
Actual	Yes	5	1
	No	18	26

Median

		Predicted	
		Yes	No
Actual	Yes	5	1
	No	18	26

25%

Location quotient

		Predicted	
		Yes	No
Actual	Yes	5	18
	No	0	27

Mean

		Predicted	
		Yes	No
Actual	Yes	19	4
	No	3	24

Median

		Predicted	
		Yes	No
Actual	Yes	19	4
	No	3	24

10%

Location quotient

		Predicted	
		Yes	No
Actual	Yes	18	2
	No	5	25

Mean

		Predicted	
		Yes	No
Actual	Yes	19	3
	No	4	24

Median

		Predicted	
		Yes	No
Actual	Yes	19	3
	No	4	24

Unknown check data

Displayed here are the confusion matrices for the "Unknown check data". To the left of each row of matrices, the confidence level applied for that particular run is indicated. Above each confusion matrix, the name of the algorithm used is specified.

100%

Location quotient

		Predicted	
		Yes	No
Actual	Yes	10	0
	No	6	10

Mean

		Predicted	
		Yes	No
Actual	Yes	11	0
	No	5	10

Median

		Predicted	
		Yes	No
Actual	Yes	11	0
	No	5	10

75%

Location quotient

		Predicted	
		Yes	No
Actual	Yes	11	0
	No	5	10

Mean

		Predicted	
		Yes	No
Actual	Yes	11	0
	No	5	10

Median

		Predicted	
		Yes	No
Actual	Yes	11	0
	No	5	10

50%

Location quotient

		Predicted	
		Yes	No
Actual	Yes	11	0
	No	5	10

Mean

		Predicted	
		Yes	No
Actual	Yes	12	0
	No	4	10

Median

		Predicted	
		Yes	No
Actual	Yes	12	0
	No	4	10

25%

Location quotient

		Predicted	
		Yes	No
Actual	Yes	12	0
	No	4	10

Mean

		Predicted	
		Yes	No
Actual	Yes	12	0
	No	4	10

Median

		Predicted	
		Yes	No
Actual	Yes	12	0
	No	4	10

10%

Location quotient

		Predicted	
		Yes	No
Actual	Yes	12	0
	No	4	10

Mean

		Predicted	
		Yes	No
Actual	Yes	12	1
	No	4	9

Median

		Predicted	
		Yes	No
Actual	Yes	12	1
	No	4	9

5.5.2 Evaluation scores

Matthews correlation coefficient

In Table 5.1, MCC for each experimental run is presented. The results are organized according to the algorithm employed, with higher values indicating more favorable outcomes.

Table 5.1: Matthews correlation coefficient scores for all algorithms

Experiment	Description	Location quotient	Mean	Median
Synthetic data	100%	0,903	1,000	1,000
Synthetic data	75%	0,903	1,000	1,000
Synthetic data	50%	0,903	1,000	1,000
Synthetic data	25%	1,000	1,000	1,000
Synthetic data	10%	1,000	1,000	1,000
All data	100%	0,361	0,155	0,155
All data	75%	0,361	0,361	0,361
All data	50%	0,361	0,277	0,277
All data	25%	0,361	0,718	0,718
All data	10%	0,721	0,718	0,718
Unknown check data	100%	0,625	0,677	0,677
Unknown check data	75%	0,677	0,677	0,677
Unknown check data	50%	0,677	0,732	0,732
Unknown check data	25%	0,732	0,732	0,732
Unknown check data	10%	0,732	0,632	0,632

H-score

Table 5.2 displays the H-score values for each test run. The data is categorized by the algorithm applied, where a higher H-score signifies better results.

Table 5.2: H-Score scores for all algorithms

Experiment	Description	Location quotient	Mean	Median
Synthetic data	100%	0,805	1,000	1,000
Synthetic data	75%	0,805	1,000	1,000
Synthetic data	50%	0,805	1,000	1,000
Synthetic data	25%	1,000	1,000	1,000
Synthetic data	10%	1,000	1,000	1,000
All data	100%	0,131	0,024	0,024
All data	75%	0,131	0,131	0,131
All data	50%	0,131	0,085	0,085
All data	25%	0,131	0,567	0,567
All data	10%	0,566	0,567	0,567
Unknown check data	100%	0,425	0,498	0,498
Unknown check data	75%	0,498	0,498	0,498
Unknown check data	50%	0,498	0,580	0,580
Unknown check data	25%	0,580	0,580	0,580
Unknown check data	10%	0,580	0,446	0,446

Chapter 6

Discussion

6.1 Experiments

This chapter discusses the results of our experiments on anomaly detection.

6.1.1 Synthetic data

Finding 1: Location Quotient (LQ) did not find synthetically created anomaly at 100% confidence.

One key observation is that the LQ algorithm struggled to identify all the introduced anomalies, specifically missing out on Anomaly 6, as outlined in Section 5.3. This particular anomaly presented a more realistic scenario where the neighboring data points weren't uniform, thus making it challenging to detect. The LQ algorithm could only identify this anomaly when compared with a lower confidence level. The sixth anomaly was harder to catch, even when assessed manually. However, as shown in this chapter, lowering the confidence level seems to be common to get an effective anomaly detection system with the type of data analyzed.

6.1.2 All data & Unkown check data

Four major insights emerged from these sets of experiments.

Finding 2: Mean and median have equal results.

Firstly, both the mean and the median consistently produced identical confusion matrices. This occurred because the chi-squared values generated by the algorithms were almost identical. The underlying reason for this uniformity lies in the nature of our dataset. Since the dataset is binary in the sense that it only allows for 'OK' or 'NOK' status and because neighboring data points often share similar values, the mean and median tend to converge.

Finding 3: Mean and median algorithms generally do better than LQ according to both the H-score and the MCC.

Secondly, the LQ algorithm outperformed the mean and median algorithms only in the "All Data" set with a 100% confidence level. In all the other experiments the mean and median algorithms either performed the same or better than LQ. This was unexpected because the LQ algorithm was introduced as the better algorithm by Singh and Lalitha [55]. Interestingly, as the confidence level dropped, the accuracy of the LQ algorithm seemed to improve.

Finding 4: Mean and median perform best at 25% confidence.

Thirdly, the mean and median algorithms achieved their most consistent score for both the H-Score and the SPTV around a 25% confidence level. This may be attributed to the mission-critical nature of components within a SPTV. Each attribute that is being analyzed can be linked to a specific component in the SPTV. Therefore, if one component is not active, it means that only one attribute of more than a hundred attributes is affected. However, in a tightly integrated system where each component is pivotal to overall operational stability, the failure of even a single element may be indicative of an anomaly. Since one attribute can have a big impact on the whole system, it makes sense that lowering the confidence level might make the algorithms better at noticing these important changes.

Finding 5: There is one finding where the H-Score and the MCC have different results.

Lastly, the H-Score and the MCC had different results into which algorithm was better in the "All Data" set at a 10% confidence level. The H-Score insinuated that the mean and median algorithms were better, while the MCC suggested the opposite. This discrepancy may arise from a different distribution of true positives and true negatives across the confusion matrices, even though the sum remains the same.

6.2 Threats to validity

Several factors pose threats to the validity of this research, which should be considered when interpreting the results.

Firstly, the study's primary limitation is its limited scope, as it tests a generic approach intended for use across various public transport providers but validates it with only one such provider with one vehicle type on one line. While the methodology is designed to be universal, its applicability across different organizations remains untested.

Next, the number of neighbors analyzed per data point is 35. Although this number was chosen partly due to the computational intensity of spatial anomaly detection, changing it could yield different outcomes. Each data point requires its data to be calculated based on its surrounding neighbors, and this computational process has to be replicated for each algorithm. Each test must process around 250,000 data points and do computational heavy calculations three times per data point (one for each algorithm). An ideal study would include a comparative analysis using different neighbor counts to enhance validity.

Another threat to the validity of this study is the lack of comparative analysis between the generalized approach and a specialized approach specifically for the given use case. Such a comparison provides a clearer understanding of the trade-offs involved in opting for a generalized solution over one that is custom-designed to address the unique requirements of a particular scenario.

In addition, the volume of data points used to construct the confusion matrices is relatively low. The process of determining whether a data point is anomalous is labor-intensive, requiring an expert review. The effort required is considerable given that each reviewed data point comprises over 140 attributes. To improve the research, an increase in supervised data is preferred, either through additional expert analysis or a more systematic labeling process.

Lastly, the subjectivity of expert evaluation introduces another potential threat to the study's validity. While an expert's deep familiarity with the system is invaluable, human error or oversight could still occur, potentially skewing the results.

Chapter 7

Conclusion

Public transport is a lifeline for cities like Amsterdam, where every day, hundreds of thousands of people rely on buses, trams, and trains to get around. Ensuring that this vital service runs smoothly is a huge task, and one tool that can help smoothen this process is anomaly detection. It's a technique not widely studied in the context of public transport. This research takes a closer look at how to make anomaly detection work better for public transport.

What strategies can be employed to design a unified data collection framework for effective anomaly detection in an SPTV? In the pursuit of a generalized anomaly detection process, the research arrived at two pivotal decisions that would shape the approach. The first deliberation centered around the choice between agent-less and agent-based data collection methods. Considering the added complexity of adding new modules with an agent-less approach and that this complexity was not present with the agent-based approach, the latter was chosen.

The second critical decision involved the choice between adopting a domain standard already in use within the industry or developing a custom standard tailored specifically for anomaly detection. This research ultimately leaned towards utilizing an established industry standard, prioritizing the simplicity and ease of adoption it offered. This choice favored a strategy building upon and enhancing existing systems rather than overhauling them.

Which anomaly detection algorithms are most effective and well-suited for handling the data derived from SPTV? This question was answered by conducting experiments employing three distinct data sets, each analyzed by three different algorithms, all focused on the operational data from a single bus line. While these experiments did not yield definitive conclusions across all levels of confidence, notable insights emerged at the 25% confidence threshold. It was here that the mean and median algorithms consistently outperformed LQ in effectiveness. Intriguingly, due to the particular configuration of the data model used in the study, the performance of the mean and median algorithms was indistinguishable, meaning that they are essentially equivalent to this data set.

7.1 Future work

Building upon the findings of this thesis, future research could significantly expand the scope and applicability of the results. A promising direction would be replicating the experiments across different public transport providers and vehicle types using the same standards. Since this research was confined to a single provider and focused solely on buses, branching out into other modes of transportation, such as trams, ferries, and trains, could provide a more comprehensive understanding of how anomaly detection algorithms perform in different settings. Variations in vehicle design, operational patterns, and system architecture could influence the effectiveness of these algorithms, and exploring these differences would contribute to a more robust approach to anomaly detection.

Moreover, an in-depth investigation into the influence of neighbor count on anomaly detection outcomes could shed light on the optimal configuration for various contexts. Since this research has set a fixed neighbor count, experimenting with different counts could optimize performance, particularly in systems where neighbor data points have a different distribution.

In predictive maintenance, the generalized approach to anomaly detection proposed in this research could serve as a stepping stone. Future studies could explore integrating this approach with real-time data streams and machine learning models to predict potential failures and optimize maintenance schedules. Such integration could enhance operational efficiency and potentially transform the way public transport systems manage their maintenance activities.

Furthermore, this research primarily concentrated on evaluating the complete state of the vehicle, encompassing a vast array of data and numerous components. While this comprehensive approach offers an extensive overview, it also raises the possibility of information overload, potentially obscuring crucial insights. A valuable area for future research would be to conduct a comparative analysis that differentiates between examining the entire state of a vehicle and focusing on specific subsections. This targeted approach could reveal whether analyzing smaller, more defined segments of vehicle data leads to more precise anomaly detection or if a holistic view is essential for accurate assessments.

Lastly, this research opens the door to developing a standardized framework for anomaly detection across the public transport sector. By leveraging industry standards for data collection and analysis, such a framework could streamline the implementation of predictive maintenance protocols, making them more accessible and cost-effective for transport providers. This standardization could also foster collaboration between different entities, driving innovation and improving public transport services globally.

Acknowledgements

I want to express my sincere gratitude to my university supervisor, Benny Akesson, for the guidance and insight provided throughout this research. My appreciation also extends to the entire GIVA team at Ximedes and, in particular, Jeroen Douwes and Jochem Sobering. Thank you for the warm welcome to the team and the practical guidance.

A special note of thanks to my friend, Tjerko Kieft, for the technical assistance in implementing the algorithms. Your help was essential in bringing this project to fruition. I am also grateful to my girlfriend, Stella Kok, and my family for their continuous support and belief in my abilities.

Bibliography

[1] S. Mazur, *New Public Transit Technology: How Emerging Tech Is Changing Transportation*, en-US, Feb. 2022. [Online]. Available: `https://www.digi.com/blog/post/new-public-transit-technology-emerging-tech` (visited on 11/03/2023).

[2] J. Hurtado, D. Salvati, R. Semola, M. Bosio, and V. Lomonaco, "Continual learning for predictive maintenance: Overview and challenges," *Intelligent Systems with Applications*, vol. 19, p. 200 251, Sep. 2023, ISSN: 2667-3053. DOI: `10.1016/j.iswa.2023.200251`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S2667305323000765` (visited on 11/03/2023).

[3] Y. Ran, X. Zhou, P. Lin, Y. Wen, and R. Deng, *A Survey of Predictive Maintenance: Systems, Purposes and Approaches*, arXiv:1912.07383 [cs, eess], Dec. 2019. DOI: `10.48550/arXiv.1912.07383`. [Online]. Available: `http://arxiv.org/abs/1912.07383` (visited on 11/03/2023).

[4] S. R. Goniwada, "Observability," en, in *Cloud Native Architecture and Design: A Handbook for Modern Day Architecture and Design with Enterprise-Grade Examples*, S. R. Goniwada, Ed., Berkeley, CA: Apress, 2022, pp. 661–676, ISBN: 978-1-4842-7226-8. DOI: `10.1007/978-1-4842-7226-8_19`. [Online]. Available: `https://doi.org/10.1007/978-1-4842-7226-8_19` (visited on 12/21/2022).

[5] K. Shaukat *et al.*, "A Review of Time-Series Anomaly Detection Techniques: A Step to Future Perspectives," en, in *Advances in Information and Communication*, K. Arai, Ed., ser. Advances in Intelligent Systems and Computing, Cham: Springer International Publishing, 2021, pp. 865–877, ISBN: 978-3-030-73100-7. DOI: `10.1007/978-3-030-73100-7_60`.

[6] A. A. Cook, G. Mısırlı, and Z. Fan, "Anomaly Detection for IoT Time-Series Data: A Survey," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6481–6494, Jul. 2020, Conference Name: IEEE Internet of Things Journal, ISSN: 2327-4662. DOI: `10.1109/JIOT.2019.2958185`.

[7] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, "Selecting Empirical Methods for Software Engineering Research," en, in *Guide to Advanced Empirical Software Engineering*, F. Shull, J. Singer, and D. I. K. Sjøberg, Eds., London: Springer, 2008, pp. 285–311, ISBN: 978-1-84800-044-5. DOI: `10.1007/978-1-84800-044-5_11`. [Online]. Available: `https://doi.org/10.1007/978-1-84800-044-5_11` (visited on 05/25/2023).

[8] C. Sridharan, *Distributed Systems Observability.* O'Reilly Media, Inc., Jul. 2018, ISBN: 978-1-4920-3342-4.

[9] B. H. Sigelman *et al.*, "Dapper, a Large-Scale Distributed Systems Tracing Infrastructure," en,

[10] Z. Huang, P. Chen, G. Yu, H. Chen, and Z. Zheng, "Sieve: Attention-based Sampling of End-to-End Trace Data in Distributed Microservice Systems," in *2021 IEEE International Conference on Web Services (ICWS)*, Sep. 2021, pp. 436–446. DOI: `10.1109/ICWS53863.2021.00063`.

[11] B. Li *et al.*, "Enjoy your observability: An industrial survey of microservice tracing and analysis," en, *Empirical Software Engineering*, vol. 27, no. 1, p. 25, Nov. 2021, ISSN: 1573-7616. DOI: `10.1007/s10664-021-10063-9`. [Online]. Available: `https://doi.org/10.1007/s10664-021-10063-9` (visited on 12/20/2022).

[12] T. Cheng and Z. Li, "A HYBRID APPROACH TO DETECT SPATIAL-TEMPORAL OUTLIERS," en,

[13] H. Alimohammadi and S. Nancy Chen, "Performance evaluation of outlier detection techniques in production timeseries: A systematic review and meta-analysis," en, *Expert Systems with Applications*, vol. 191, p. 116 371, Apr. 2022, ISSN: 0957-4174. DOI: `10.1016/j.eswa.2021.116371`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S095741742101664X` (visited on 05/26/2023).

[14] a. Pawan_Dubey, *Confusion Matrix*, en-gb, Aug. 2019. [Online]. Available: `https://devopedia.org/confusion-matrix` (visited on 08/25/2023).

[15] C. Parker, "On measuring the performance of binary classifiers," en, *Knowledge and Information Systems*, vol. 35, no. 1, pp. 131–152, Apr. 2013, ISSN: 0219-3116. DOI: `10.1007/s10115-012-0558-x`. [Online]. Available: `https://doi.org/10.1007/s10115-012-0558-x` (visited on 05/26/2023).

[16] *Transmodel – CEN Reference Data Model for Public Transport*, en-US. [Online]. Available: `https://www.transmodel-cen.eu/` (visited on 09/18/2023).

[17] *General Transit Feed Specification*. [Online]. Available: `https://gtfs.org/` (visited on 08/16/2023).

[18] *CEN-CENELEC*. [Online]. Available: `https://www.cencenelec.eu/` (visited on 08/24/2023).

[19] *NeTEx — Network Timetable Exchange*, en-US. [Online]. Available: `https://netex-cen.eu/` (visited on 08/24/2023).

[20] *GTFS Realtime Overview — Realtime Transit*, en. [Online]. Available: `https://developers.google.com/transit/gtfs-realtime` (visited on 08/24/2023).

[21] CNC, *SIRI-CEN*, en. [Online]. Available: `https://siri-cen.eu/` (visited on 08/24/2023).

[22] *ITxPT*, en-GB. [Online]. Available: `https://itxpt.org/` (visited on 01/23/2023).

[23] *Der Verband Deutscher Verkehrsunternehmen (VDV) stellt sich vor*. [Online]. Available: `https://www.vdv.de/` (visited on 08/24/2023).

[24] *GIVA in a Day - Ximedes*, en. [Online]. Available: `https://ximedes.com/blog/2021-12-14/giva-in-a-day` (visited on 08/07/2023).

[25] *Wat we doen*. [Online]. Available: `https://over.gvb.nl/organisatie/wat-we-doen/` (visited on 08/07/2023).

[26] P. Jamshidi, C. Pahl, N. C. Mendonça, J. Lewis, and S. Tilkov, "Microservices: The Journey So Far and Challenges Ahead," *IEEE Software*, vol. 35, no. 3, pp. 24–35, May 2018, Conference Name: IEEE Software, ISSN: 1937-4194. DOI: `10.1109/MS.2018.2141039`.

[27] C. Liu, Z. Cai, B. Wang, Z. Tang, and J. Liu, "A protocol-independent container network observability analysis system based on eBPF," in *2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*, ISSN: 2690-5965, Dec. 2020, pp. 697–702. DOI: `10.1109/ICPADS51040.2020.00099`.

[28] H. Zahid, S. Hina, M. F. Hayat, and G. A. Shah, "Agentless Approach for Security Information and Event Management in Industrial IoT," en, *Electronics*, vol. 12, no. 8, p. 1831, Jan. 2023, Number: 8 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2079-9292. DOI: `10.3390/electronics12081831`. [Online]. Available: `https://www.mdpi.com/2079-9292/12/8/1831` (visited on 09/11/2023).

[29] S. Shengyan, S. Xiaoliu, Z. Jianbao, and M. Xinke, "Research on System Logs Collection and Analysis Model of the Network and Information Security System by Using Multi-agent Technology," in *2012 Fourth International Conference on Multimedia Information Networking and Security*, ISSN: 2162-8998, Nov. 2012, pp. 23–26. DOI: `10.1109/MINES.2012.181`.

[30] J. Turnbull, *The Logstash Book*, en. James Turnbull, Mar. 2013, Google-Books-ID: lhMKBAAAQBAJ, ISBN: 978-0-9888202-1-0.

[31] *Logstash: Collect, Parse, Transform Logs — Elastic*. [Online]. Available: `https://www.elastic.co/logstash` (visited on 09/11/2023).

[32] J. Levin and T. A. Benson, "ViperProbe: Rethinking Microservice Observability with eBPF," in *2020 IEEE 9th International Conference on Cloud Networking (CloudNet)*, Nov. 2020, pp. 1–8. DOI: `10.1109/CloudNet51028.2020.9335808`.

[33] M. Amaral, T. Chiba, S. Trent, T. Yoshimura, and S. Choochotkaew, "MicroLens: A Performance Analysis Framework for Microservices Using Hidden Metrics With BPF," in *2022 IEEE 15th International Conference on Cloud Computing (CLOUD)*, ISSN: 2159-6190, Jul. 2022, pp. 230–240. DOI: `10.1109/CLOUD55607.2022.00043`.

[34] R. Brondolin and M. D. Santambrogio, "A Black-box Monitoring Approach to Measure Microservices Runtime Performance," *ACM Transactions on Architecture and Code Optimization*, vol. 17, no. 4, 34:1–34:26, Nov. 2020, ISSN: 1544-3566. DOI: `10.1145/3418899`. [Online]. Available: `https://dl.acm.org/doi/10.1145/3418899` (visited on 09/11/2023).

[35] J. Jalaney and D. R. S. Ganesh, "Review on IoT Based Architecture for Smart Public Transport System," en, vol. 14, no. 2, 2019.

[36] D. F. Murad, B. S. Abbas, A. Trisetyarso, W. Suparta, and C.-H. Kang, "Development of smart public transportation system in Jakarta city based on integrated IoT platform," in *2018 International Conference on Information and Communications Technology (ICOIACT)*, Mar. 2018, pp. 872–878. DOI: 10.1109/ICOIACT.2018.8350812.

[37] D. S. Paolino, "An innovative approach to maintenance for a bus fleet," en,

[38] Q. Du, T. Xie, and Y. He, "Anomaly Detection and Diagnosis for Container-Based Microservices with Performance Monitoring," en, in *Algorithms and Architectures for Parallel Processing*, J. Vaidya and J. Li, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, pp. 560–572, ISBN: 978-3-030-05063-4. DOI: 10.1007/978-3-030-05063-4_42.

[39] M.-O. Pahl and F.-X. Aubet, "All Eyes on You: Distributed Multi-Dimensional IoT Microservice Anomaly Detection," in *2018 14th International Conference on Network and Service Management (CNSM)*, ISSN: 2165-963X, Nov. 2018, pp. 72–80.

[40] S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting Traffic Anomaly Detection Using Software Defined Networking," en, in *Recent Advances in Intrusion Detection*, R. Sommer, D. Balzarotti, and G. Maier, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2011, pp. 161–180, ISBN: 978-3-642-23644-0. DOI: 10.1007/978-3-642-23644-0_9.

[41] S. Lee, J. Kim, S. Shin, P. Porras, and V. Yegneswaran, "Athena: A Framework for Scalable Anomaly Detection in Software-Defined Networks," in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, ISSN: 2158-3927, Jun. 2017, pp. 249–260. DOI: 10.1109/DSN.2017.42.

[42] R. Chen *et al.*, "LogTransfer: Cross-System Log Anomaly Detection for Software Systems with Transfer Learning," in *2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)*, ISSN: 2332-6549, Oct. 2020, pp. 37–47. DOI: 10.1109/ISSRE5003.2020.00013.

[43] F. Caetano, P. Carvalho, and J. Cardoso, "Deep Anomaly Detection for In-Vehicle Monitoring—An Application-Oriented Review," en, *Applied Sciences*, vol. 12, no. 19, p. 10 011, Jan. 2022, Number: 19 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2076-3417. DOI: 10.3390/app121910011. [Online]. Available: https://www.mdpi.com/2076-3417/12/19/10011 (visited on 04/21/2023).

[44] D. Bogdoll, M. Nitsche, and J. M. Zöllner, "Anomaly Detection in Autonomous Driving: A Survey," en, 2022, pp. 4488–4499. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2022W/WAD/html/Bogdoll_Anomaly_Detection_in_Autonomous_Driving_A_Survey_CVPRW_2022_paper.html (visited on 04/21/2023).

[45] P. Dixit, P. Bhattacharya, S. Tanwar, and R. Gupta, "Anomaly detection in autonomous electric vehicles using AI techniques: A comprehensive survey," en, *Expert Systems*, vol. 39, no. 5, e12754, 2022, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/exsy.12754, ISSN: 1468-0394. DOI: 10.1111/exsy.12754. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.12754 (visited on 04/21/2023).

[46] Y. Lin, C. Chen, F. Xiao, O. Avatefipour, K. Alsubhi, and A. Yunianta, "An Evolutionary Deep Learning Anomaly Detection Framework for In-Vehicle Networks - CAN Bus," *IEEE Transactions on Industry Applications*, pp. 1–1, 2020, Conference Name: IEEE Transactions on Industry Applications, ISSN: 1939-9367. DOI: 10.1109/TIA.2020.3009906.

[47] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly Detection in Automobile Control Network Data with Long Short-Term Memory Networks," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Oct. 2016, pp. 130–139. DOI: 10.1109/DSAA.2016.20. [Online]. Available: https://ieeexplore.ieee.org/document/7796898 (visited on 01/06/2024).

[48] S. N. Narayanan, S. Mittal, and A. Joshi, "OBD_securealert: An Anomaly Detection System for Vehicles," in *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*, May 2016, pp. 1–6. DOI: 10.1109/SMARTCOMP.2016.7501710.

[49] J. Kang, C.-S. Kim, J. W. Kang, and J. Gwak, "Anomaly Detection of the Brake Operating Unit on Metro Vehicles Using a One-Class LSTM Autoencoder," en, *Applied Sciences*, vol. 11, no. 19, p. 9290, Jan. 2021, Number: 19 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2076-3417. DOI: 10.3390/app11199290. [Online]. Available: https://www.mdpi.com/2076-3417/11/19/9290 (visited on 04/25/2023).

[50] S. R. Maskey, S. Badsha, S. Sengupta, and I. Khalil, "BITS: Blockchain based Intelligent Transportation System with Outlier Detection for Smart City," in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Mar. 2020, pp. 1–6. DOI: `10.1109/PerComWorkshops48775.2020.9156237`.

[51] Y. Kou, C.-T. Lu, and D. Chen, "Spatial Weighted Outlier Detection," en,

[52] S. Shekhar, C.-T. Lu, and P. Zhang, "A Unified Approach to Detecting Spatial Outliers," en, *GeoInformatica*, vol. 7, no. 2, pp. 139–166, Jun. 2003, ISSN: 1573-7624. DOI: `10.1023/A:1023455925009`. [Online]. Available: `https://doi.org/10.1023/A:1023455925009` (visited on 05/22/2023).

[53] H. Izakian and W. Pedrycz, "Anomaly Detection and Characterization in Spatial Time Series Data: A Cluster-Centric Approach," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 6, pp. 1612–1624, Dec. 2014, Conference Name: IEEE Transactions on Fuzzy Systems, ISSN: 1941-0034. DOI: `10.1109/TFUZZ.2014.2302456`.

[54] C.-T. Lu, D. Chen, and Y. Kou, "Detecting spatial outliers with multiple attributes," in *Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence*, ISSN: 1082-3409, Nov. 2003, pp. 122–128. DOI: `10.1109/TAI.2003.1250179`.

[55] A. K. Singh and S. Lalitha, "A novel spatial outlier detection technique," *Communications in Statistics - Theory and Methods*, vol. 47, no. 1, pp. 247–257, Jan. 2018, Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/03610926.2017.1301477, ISSN: 0361-0926. DOI: `10.1080/03610926.2017.1301477`. [Online]. Available: `https://doi.org/10.1080/03610926.2017.1301477` (visited on 05/22/2023).

[56] M. A. Hayes and M. A. Capretz, "Contextual Anomaly Detection in Big Sensor Data," in *2014 IEEE International Congress on Big Data*, ISSN: 2379-7703, Jun. 2014, pp. 64–71. DOI: `10.1109/BigData.Congress.2014.19`.

[57] A. Alvera-Azcárate, D. Sirjacobs, A. Barth, and J. .-. Beckers, "Outlier detection in satellite data using spatial coherence," en, *Remote Sensing of Environment*, vol. 119, pp. 84–91, Apr. 2012, ISSN: 0034-4257. DOI: `10.1016/j.rse.2011.12.009`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0034425711004469` (visited on 05/22/2023).

[58] O. Alghushairy, R. Alsini, T. Soule, and X. Ma, "A Review of Local Outlier Factor Algorithms for Outlier Detection in Big Data Streams," en, *Big Data and Cognitive Computing*, vol. 5, no. 1, p. 1, Mar. 2021, Number: 1 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2504-2289. DOI: `10.3390/bdcc5010001`. [Online]. Available: `https://www.mdpi.com/2504-2289/5/1/1` (visited on 06/29/2023).

[59] Y. Kou, C.-T. Lu, and R. F. Dos Santos, "Spatial Outlier Detection: A Graph-Based Approach," eng, Book Title: 19th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007) ISSN: 1082-3409, vol. 1, IEEE, 2007, pp. 281–288, ISBN: 978-0-7695-3015-4. DOI: `10.1109/ICTAI.2007.139`.

[60] P. Harris, C. Brunsdon, M. Charlton, S. Juggins, and A. Clarke, "Multivariate Spatial Outlier Detection Using Robust Geographically Weighted Methods," en, *Mathematical Geosciences*, vol. 46, no. 1, pp. 1–31, Jan. 2014, ISSN: 1874-8953. DOI: `10.1007/s11004-013-9491-0`. [Online]. Available: `https://doi.org/10.1007/s11004-013-9491-0` (visited on 05/22/2023).

Acronyms

CEN Committee for Standardization. 10

eBPF Extended Berkeley Packet Filter. 12

GIVA Generic ICT Vehicle Architecture. 11, 12, 16, 21

GNSS Global Navigation Satellite System. 18

GTFS General Transit Feed Specification. 10

GVB gemeentevervoerbedrijf. 11

IoT Internet of Things. 4

ITxPT Information Technology for Public Transport. 2, 10, 11, 13, 15, 17–19

KNN K-nearest neighbour. 8

LQ Location Quotient. 26, 28

MCC Matthews correlation coefficient. 9, 21, 22, 24, 26, 27

NeTEx Network Timetable Exchange. 10

PTV public transport vehicle. 4, 8, 10

SIEM Security Information and Event Management System. 12

SIRI Service Interface for Real-time Information. 10

SPTV smart public transport vehicle. 1, 4–8, 10–13, 15–20, 26, 27

VDV Der Verband Deutscher Verkehrsunternehmen. 10, 17

Appendix A

Non-crucial information

A.1 Data format

- journeyDelay
- vehicleOdoDistance
- uptime
- signalQuality
- module__B1_FRONT_DISPL
- module__B1_Front_Line_Displ
- module__B1_Interior_Displ_1
- module__B1_Route_Map_Displ_A1
- module__B1_Route_Map_Displ_A2
- module__B1_Route_Map_Displ_A3
- module__B1_Route_Map_Displ_A4
- module__B1_Route_Map_Displ_B1
- module__B1_Route_Map_Displ_B3
- module__B1_TFT_A1
- module__B1_TFT_A2
- module__B1_TFT_A3
- module__B1_TFT_B1
- module__B1_TFT_B2
- module__B1_TFT_B3
- module__B2_Interior_Displ_3
- module__B2_Interior_Displ_4
- module__B2_Route_Map_Displ_A7
- module__B2_Route_Map_Displ_B6
- module__B2_Route_Map_Displ_B7
- module__B2_Side_Displ_A
- module__B2_Side_Displ_B
- module__B2_TFT_A4
- module__B2_TFT_A5
- module__B2_TFT_A6
- module__B2_TFT_B4
- module__B2_TFT_B5
- module__B2_TFT_B6
- module__B3_Interior_Displ_5
- module__B3_Interior_Displ_6
- module__B3_Route_Map_Displ_A11
- module__B3_Route_Map_Displ_A9
- module__B3_Route_Map_Displ_B10
- module__B3_Route_Map_Displ_B11

- module__B3_Route_Map_Displ_B9
- module__B3_TFT_A7
- module__B3_TFT_A8
- module__B3_TFT_A9
- module__B3_TFT_B7
- module__B3_TFT_B8
- module__B3_TFT_B9
- module__Binnen 1
- module__Binnen 2
- module__Binnen 3
- module__Binnen 4
- module__Buiten bestemming voor
- module__Buiten lijn achter
- module__Buiten lijn links
- module__Buiten lijn voor
- module__Buiten rechts
- module__Buiten rechtsmidden
- module__CMA server
- module__Dienstregeling
- module__EBS aanmelding
- module__FSZ4-1
- module__GNSS signaal
- module__Kaartlezer bestuurder
- module__Kaartlezer conducteur
- module__Kassascherm bestuurder
- module__Kassascherm conducteur
- module__Mobilofoon
- module__Mobilofoon verbinding
- module__Mvb
- module__ODO/GNSS afstand test
- module__PIN bestuurder
- module__PIN conducteur
- module__Router
- module__Switch01
- module__Switch04
- module__Tijdcontrole bestuurder
- module__VECOM-A
- module__VGS koppeling
- module__Vecom lijncheck
- module__Verkeersberichten
- module__camera8
- module__cardwr1
- module__cardwr2
- module__frontdisplay
- module__hmi1
- module__hmi2
- module__leftdisplay
- module__pin1
- module__pin2
- module__reardisplay
- module__recorder
- module__reisinfoscherm1
- module__reisinfoscherm2
- module__reisinfoscherm3

- module__reisinfoscherm4
- module__rightdisplay
- module__ticketvalidator01
- module__ticketvalidator02
- module__ticketvalidator03
- module__ticketvalidator04
- module__ticketvalidator05
- module__ticketvalidator06
- module__ticketvalidator07
- module__ticketvalidator08
- module__ticketvalidator09
- module__ticketvalidator10
- module__ticketvalidator11
- module__ticketvalidator12
- module__ticketvalidator13
- module__vecom_c1
- module__vecom_c2
- module__avms__TrainSpeedmodule
- module__avms__avms_jm
- module__avms__avms_pp
- module__avms__avms_rm
- module__ecr_conductor__Backend
- module__ecr_conductor__PinPayment
- module__ecr_driver__Backend
- module__ecr_driver__CardReader
- module__ecr_driver__Pin
- module__ecr_driver__PinPayment
- module__ecr_driver__SigninDelivery
- module__ecr_driver__activate
- module__ecr_driver__closeSession
- module__ecr_driver__paymentResult
- module__ecr_driver__paymentRetry
- module__ecr_driver__sale
- module__ecr_driver__startPayment
- module__extdisplay__/subscribe
- module__extdisplay__currentPlannedPattern
- module__extdisplay__nextPlannedPattern
- module__gnss_location_udp__MaxNMEAMessages
- module__gnss_location_udp__NmeaFramesConnection
- module__gnss_location_udp__ValidNMEAMessage
- module__hmi__FrontendCheck-conductor
- module__hmi__HmiDelivery
- module__hmi__PlannedPatternDelivery
- module__hmi__VisStaticDelivery
- module__intstopdisplay__/subscribe
- module__metro_signin__DeliveryMonitor
- module__metro_signin__PmrDelivery
- module__passenger_inf__-
- module__passenger_inf__0:0:0:0:0:0:0:1
- module__passenger_inf__129
- module__passenger_inf__130
- module__passenger_inf__131
- module__passenger_inf__132
- module__passenger_inf__Commercial transactions

- module__passenger_inf__Stop facilities
- module__passenger_inf__Transfer and disruption
- module__passenger_inf__avms_jm
- module__passenger_inf__avms_pp
- module__passenger_inf__avms_rm
- module__passenger_inf__avms_vm
- module__routedisplay__/subscribe
- module__signin__/
- module__signin__PmrDelivery
- module__signin__SigninPmrStatusmodule
- module__snmp_trap__AVMS verbinding
- module__snmp_trap__Camera 1 signaal
- module__snmp_trap__Camera 2 signaal
- module__snmp_trap__Camera 3 signaal
- module__snmp_trap__Camera 4 signaal
- module__snmp_trap__Camera 5 signaal
- module__snmp_trap__Camera 6 signaal
- module__snmp_trap__Camera 7 signaal
- module__snmp_trap__Camera 8 signaal
- module__snmp_trap__Mainstate
- module__snmp_trap__RunMonitoringDelivery
- module__snmp_trap__StartupEvent
- module__snmp_trap__Voertuiginfo status
- module__systemmgt__VisStaticDelivery
- module__tccgatewaycontrol__Connected
- module__tccgatewaycontrol__PowerOn
- module__tccgatewaycontrol__UkrActionPoint
- module__tccgatewaycontrol__UkrDeviation
- module__tccgatewaycontrol__UkrGPSPosition
- module__tccgatewaycontrol__UkrPingPosition
- module__tccgatewaycontrol__UkrSignOut
- module__tccgatewaycontrol__UkrVersionInfo
- module__vehicle_s_inf_FMS__/vehiclestaticinformation
- module__vehicle_s_inf_MVB__/vehiclestaticinformation
- module__vehicle_status__CmaStatusMessageProcessor
- module__vehicle_status__JourneyMonitoringDelivery
- module__vehicle_status__NtpClient
- module__vehicle_status__RunMonitoringDelivery
- module__vehicle_status__VisDynamicDelivery
- module__vehicle_status__VisStaticDelivery
- module__vgs__/subscribe
- module__vgs__TrainCompositionCheck
- module__vgs__VisStaticDelivery