Fleet and traffic management systems
for conducting future cooperative mobility

D3.3 Specification and initial version of anomaly detection routines

Document Type	Deliverable
Document Number	D3.3
Primary Author(s)	Margarita Tsarmopoulou | FRIC
Document Version / Status	V1.0 | Final
Distribution Level	PU (public)
Project Acronym	CONDUCTOR
Project Title	Fleet and traffic management system for conducting cooperative mobility
Project Website	https://conductor-project.eu/
Project Coordinator	Netcompany Intrasoft SA | www.netcompany-intrasoft.com
Grant Agreement Number	101077049

CONDUCTOR project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101077049.

CONTRIBUTORS

Name	Organization	Name	Organization
Margarita Tsarmpopoulou	FRIC	Oliva García Cantú	Nommon
Raquel Sánchez	Nommon		

FORMAL REVIEWERS

Name	Organization	Date
Athina Tympakianaki	Aimsun	2024-01-18
Oscar Eikenbroek	University of Twente	2024-01-12

DOCUMENT HISTORY

Revision	Date	Author / Organization	Description
0.1	2023-10-12	Margarita Tsarmpopoulou (FRIC)	Definitions of terms of content
0.2	2023-10-12	Margarita Tsarmpopoulou (FRIC)	Review of terms of content
0.3	2023-12-22	Margarita Tsarmpopoulou (FRIC)	Frist Draft
0.4	2024-01-18	Margarita Tsarmpopoulou (FRIC)	Draft under review
0.5	2024-01-25	Margarita Tsarmpopoulou (FRIC)	Final draft
1.0	2024-01-29	Flavien Massi (INTRA)	Final version

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

The CONDUCTOR project's main goal is to revolutionize the landscape of transportation by spearheading advanced traffic and fleet management solutions for the seamless and globally optimal movement of passengers and goods. At its core, CONDUCTOR aims to establish an innovative paradigm by integrating dynamic balancing, priority-based vehicle management, and cutting-edge technologies into the Cooperative, Connected, and Automated Mobility (CCAM) ecosystem.

The document "CONDUCTOR D3.3 Specification and initial version of anomaly detection routines" primarily addresses the development of anomaly detection methods in traffic patterns and transport demand for the CONDUCTOR project. It provides an exhaustive literature review, outlining various methodologies in the field, and discusses the project's focus on employing machine learning and statistical approaches for anomaly detection. The document details the technical aspects of the project, including data sources, methodology, model selection, implementation, and preliminary results. It emphasizes the importance of identifying and understanding traffic anomalies for effective traffic management and includes evaluations of different models used for anomaly detection.

The deliverable D3.3 focuses on the specification and initial implementation of these detection routines. It leverages a blend of machine learning and statistical methods to identify irregular traffic patterns and transport demand anomalies. This document presents comprehensive literature reviews, detailed methodology, and evaluation of various models, laying a foundation for future iterations and improvements in anomaly detection within the CONDUCTOR framework. The project is a significant step towards revolutionizing transportation management with intelligent, data-driven solutions.

Keywords**:** Cooperative, Connected, and Automated Mobility (CCAM), Traffic and Fleet Management, Dynamic Balancing, Machine Learning and Data Fusion, Interoperability, Autonomous Vehicles, Urban Traffic Reduction, Demand-Response Transport, Open Platform Integration, Quality of Life Improvement.

1 INTRODUCTION

1.1 Background

The landscape of modern transportation is evolving rapidly, driven by technological advancements and the integration of autonomous and connected vehicles. In response to these changes, the CONDUCTOR project is actively engaged in pioneering efforts to shape the future of transportation through resilient multimodal autonomous mobility. One of the critical components of this initiative is the development and implementation of anomaly detection routines, a fundamental aspect addressed in this deliverable (deliverable D3.3).

CONDUCTOR leverages state-of-the-art fleet and traffic management solutions, elevating them through the incorporation of machine learning and data fusion. The project's focal point is to empower transport authorities and operators to act as "conductors" of future mobility networks, thereby orchestrating efficient, responsive, and centralized control over traffic and fleets. The initiative seeks to position autonomous vehicles as central to future city landscapes, enhancing safety measures and offering flexible, responsive traffic management.

The project's anticipated outcomes encompass a reduction in urban traffic and congestion, diminished pollution levels, and an overall improvement in the quality of life for city dwellers. The project's innovations will be consolidated into a common, open platform, fostering interoperability of traffic management systems. Validation of these advancements will occur through three distinct use cases:

1. Use Case 1 (UC1): Integration of traffic management with intermodality.
2. Use Case 2 (UC2): Testing demand-response transport solutions.
3. Use Case 3 (UC3): Addressing urban logistics challenges.

Each use case will undergo rigorous testing and validation, combining simulations with real-life data to ensure the practicality and effectiveness of the proposed solutions. By creating a comprehensive, open platform, CONDUCTOR aims to set new standards in traffic and fleet management, ushering in a future marked by reduced urban congestion, lower pollution levels, and an elevated urban living experience.

1.2 Objectives and contribution

Deliverable D3.3, titled "Specification and initial version of anomaly detection routines," is a pivotal document within the CONDUCTOR project. The primary objective is to provide a comprehensive specification of anomaly detection techniques that will empower the identification of specific traffic patterns. These patterns, ranging from routine traffic flows to emerging critical network anomalies, require identification in case remedial actions and response plans need to be implemented as countermeasures.

This deliverable is intricately linked to Task 3.5, which is spearheaded by Frontier Innovations as the lead beneficiary, with the valuable contributions of Nommon for the Anomaly Detection in Transport Demand. Task 3.5, spanning from Month 3 to Month 30, focuses on the development of advanced situation detection capabilities. These capabilities are crucial for identifying traffic patterns and emerging critical network and traffic anomalies, facilitating adaptive measures such as network adaptation, traffic redirection, and coordination.

This deliverable's aim to present the types of incidents that will be handled by CONDUCTOR, both in the supply and the demand side. In the first iteration of the deliverable, since the final datasets

that will be used in the project are not yet finalized, a more methodological approach will be followed, with an initial implementation of anomaly detection and forecasting techniques aiming to showcase the capabilities of the CONDUCTOR components.

As the implementation of CONDUCTOR is in line with the agile methodology, each individual component is implemented in parallel, and the findings and key assumptions will be validated during the first iteration. After the first iteration and the first demonstration of results every output will be adjusted according to the feedback and will be enriched in the next iteration of the deliverable (D3.4).

1.3 Outline

Anomaly detection in traffic patterns and transport demand is a critical aspect of traffic management, operations and control, especially event detection. In this deliverable our aim is to investigate methods for achieving detection, coming from various sources. Those sources include, amongst others, Origin-Destination (OD) matrices and traffic flow data from embedded sensors that monitor the infrastructure of a network.

Our objective is to conduct a comprehensive analysis, leveraging state-of-the-art machine learning algorithms and data fusion techniques, to detect supply and demand anomalies. The identification of such anomalies serves as a pre-emptive mechanism, alerting "conductors" to potential problems in real-time.

A key focus of this deliverable is to provide a data-driven definition of anomalies. In the context of traffic, anomalies signify substantial deviations from the expected normal traffic behaviour, encompassing a spectrum of events that can disrupt the regular flow of vehicles within a network, distinguishing them from traditional incident detection as often claimed in existing research papers. Recognizing and characterizing these anomalies is indispensable for proactive traffic management, facilitating timely responses to emergent situations.

In the context of transport demand, anomalies are considered as every abnormal (or non-recurrent) demand behaviour that would require the application of special measures on the supply side (beyond normal service functioning) to address the needs of the travellers.

It is essential to emphasize that anomaly detection and incident detection are not always the same thing. For example, an incapacitated vehicle during a period of low traffic flow may not impact the overall network, hence does not pose an anomaly to be detected. This document unfolds with an Executive Summary, encapsulating the essence of the anomaly detection routines and their significance within the broader CONDUCTOR project. The subsequent sections delve into the background, objectives, and contributions of Deliverable D3.3. Following the introduction, detailed specifications and the initial version of the anomaly detection routines will be presented, aligning with the goals set forth by Task 3.5.

2 LITERATURE REVIEW

2.1 Anomaly Detection in Traffic Patterns

Various methodologies have been proposed for detecting anomalies in traffic patterns, broadly categorized into two classes. The first-class leverages data from individual vehicles, utilizing automatic vehicle identification systems [1], cameras for individual vehicle identification [2], or GPS and social media data from navigation apps like Waze [3]. The second class employs time series data from embedded loop sensors, focusing on aggregated measures of vehicle counts, occupancy, and speed.

Our approach falls, at least in the scope of the project, within the second class, and in this section, we provide a summary of pertinent literature. One prevalent method is basic pattern matching, exemplified by the California algorithms and their variants [4] [5]. These algorithms compare occupancy values at adjacent sensors, employing pair-wise metrics to detect deviations from established thresholds.

A second notable approach is the McMaster algorithm, rooted in catastrophe theory [6] [7]. It constructs a lower bound of occupancy-flow data based on uncongested regimes, defining critical occupancies and flows to categorize system states. Calibration of thresholds and fitting parametric forms pose challenges but have been addressed in subsequent works, such as using a particle-swarm approach [8].

Addressing the difficulty in calibrating incident detection models due to data quality issues [9], recent methodologies focus on clustering data into typical and anomalous states. [10], distinguish non-recurrent congestions using journey time scaling, while Piciarelli and Foresti [11] cluster vehicle trajectories to identify anomalous events. Gaussian Mixture Hidden Markov Models [12] and approaches based on change-point detection [13] also contribute to anomaly detection.

A third approach involves the standard normal deviate (SND) methodology [14], constructing mean and variation values for traffic variables. Calibration challenges are addressed by pre-filtering datasets [15] or using robust summary statistics [16]. Spatial information is incorporated in SND-based algorithms [17], while Chakraborty proposes noise threshold construction with spatial-temporal correlations [18].

Machine learning and deep learning methodologies have gained traction, such as Yuan et al. [19] incorporating traffic data, weather information, and spatial structure into a convolutional-LSTM model. Computational intensity and data quality issues remain challenges for these methods.

Our proposed approach draws inspiration from this diverse literature. Like the McMaster algorithm, we segment the density-flow diagram into distinct regions. However, unlike McMaster, our segmentation is data-driven and parameter-free, akin to the SND algorithm. This method combines segmentation principles with robust statistical approaches, avoiding the challenges of obtaining labelled data and calibrating an event detection system.

2.2 Anomaly Detection in Transport Demand

The knowledge of OD flows is crucial for traffic operations and control. The characterisation of mobility patterns and demand and the understanding of anomalies in those patterns allows us to anticipate and give response to the needs of the travellers. As mentioned in Section 1.3, from the demand perspective, we understand as anomaly every abnormal (or non-recurrent) value that would require the application of special measures (beyond normal service functioning). Usually, anomalies in the transport demand can be explained by special events (such as holidays, strikes, or sport

events), incidents, or weather conditions, each of which requires different remedial actions or response plans. These anomalies usually translate into traffic anomalies (and the same happens the other way around: traffic anomalies usually generate anomalies in the demand, as travellers tend to use other routes and modes to avoid the anomaly). So, usually, anomalies in traffic and anomalies in the demand go hand in hand.

One common approach when detecting anomalies in transport demand is to consider historical data as a time series, use it to fit a model and compare the prediction with the actual demand volumes [20] [21] [22]. Based on this comparison, the actual demand is classified as a normal value or as an anomaly. Additionally, other works analyse the factors that may have caused them (weather conditions, especial events, etc.) [23] [24] [25]. Based on that, the anomalies can be classified, and response plans can be associated to each class.

Other works focus on predicting mobility demand patterns (including abnormal or non-recurrent ones), looking for models that are robust against anomalies [26] [27] [28]. These works do not detect anomalies strictly speaking; however, the design of time series forecasting algorithms that are robust to large fluctuations in demand is highly related to the correct identification of anomalies and the approaches used are also of interest for this literature review.

When detecting anomalies, a great challenge is to properly define what an anomaly is. For that, some works define and construct a “normal” baseline scenario and look for deviation from that baseline [25] [23] [24]. Other works use some kind of confidence intervals to detect and classify the anomalies, establishing different levels of anomalies [21]. These intervals can also be used to identify when a model recalibration is needed.

The techniques found in the literature for anomaly detection in transport demand are mainly generic methods for time series prediction and anomaly detection. These techniques include traditional parametric and non-parametric statistical methods, machine learning models, deep learning models, and hybrid methods. Next, we briefly discuss the most relevant ones.

Among the statistical methods, the more relevant are Kalman filters [26] [28] [21] and Gaussian kernel functions [23] [24]. In particular, in those two lasts works the demand distribution of a “normal” baseline scenario is estimated using a Gaussian kernel function and anomalies are found with a density criterion looking for deviation from the average using a Z-score measure. This allows not only the identification of abnormal days but also the location. This information is used to search for special events that may cause the anomaly.

Statistical methods are mainly used with small datasets (small OD matrices or individual OD pairs, and short time periods), and have the limitation that they are not very scalable to large datasets. As the volume of data increase, machine learning, and specially, deep learning techniques, have become more popular.

Machine learning models include both supervised learning models (linear models [29], support vector machines (SVMs) [22], random forest [22], K-nearest neighbour models (K-NN) [26] [28]) and unsupervised learning models (principal component analysis (PCA) [30] [26] [28]). The PCA technique is mainly used as a data pre-processing step to reduce the dimension of the OD matrices.

Deep learning models mainly include long short-term memory (LSTM) network-based models [22] [29] [20], although regular neural networks are also found [20].

Finally, hybrid methods are also proposed. Among these, we highlight four works. Pasiniproposes a LSTM encoder-predictor model, which combines the capabilities of LSTM models with a recurrent neural network (RNN) encoder-decoder structure to predict the short-term evolution of trainload [20]. Davis proposes an anomaly detection-based model that combines a LSTM model with and Extreme Value Theory (EVT)-based approach [22]. This model uses an LSTM model to predict the demand of the next time step of a time series and uses an EVT-based rule to develop anomaly thresholds on

prediction errors. This rule is based on a EVT result that states that, under a weak condition, the extreme events have a known distribution, called Extreme Value Distribution.

Following the approach of Zheng [28], Liu proposes a dynamic traffic demand prediction framework that combines a parametric with a non-parametric model [26]. For that, PCA is used to extract main demand patterns from historical data. If those patterns are considered as “normal”, then a parametric model (Kalman model) is used for prediction. Otherwise, a non-parametric model (K-NN model) is used. This distinction is based on the conclusions reached in [28], in which the Kalman filter model (parametric model) performs better for regular OD flows, while K-NN methods (non-parametric model) have a better performance for abnormal patterns.

Finally, [27] generate a Gaussian conditional random field (GCRF) model trained with a boosting approach, using the adaptive boosting (AdaBoost) technique, to enhance the predictive capabilities of GCRFs. This method proves to be robust under demand anomalies.

Regarding the evaluation metrics and the performance measures, the most common and relevant metrics are the mean absolute error [26] [28], root mean absolute error [27] [28], mean absolute percentage error [27], cumulative variance explanation [26], and temporal relative L2-norm [21]. Most of the works used more than one metric in order to assess the results, mainly combining an absolute error and a relative error metric. This way complementary information is provided, which allows a deeper and more complete performance analysis.

In almost all the works reviewed, the performance of difference methods is compared, mainly to assess the predictive performance of the approach proposed in the work. In general, LSTM networks and hybrid models outperform the rest of the models considered, showing the great capability of complex deep learning techniques. In particular, Davis et al. compare the predictive performance of their hybrid LSTM-EVT model with that of regular LSTMs, SVMs and generalized auto regressive conditional heteroskedasticity models [22]. Their method outperforms the rest of the algorithms in almost of the cases. Finally, Qian et al., compare the performance of the GCRF model trained with an AdaBoost technique with that of a bagging-GCRF (a GCRF model trained with the bagging technique), a multilayer perceptron, a convolutional neural network, and a LSTM network. The results show that the proposed model is able to better forecast the distribution of short-term OD flows [27].

It is important to note that both comparisons were performed with small OD matrices, so it is to be expected that when the size of the OD pairs increase, the statistical method component of these hybrid methods may not scale well.

2.3 Summary

In the evolving area of traffic management, identifying and understanding anomalies in traffic patterns and transport demand has become essential. Our review covers a range of methods, each contributing in its own way to our grasp and control of traffic systems.

Emerging Technologies and Methodologies

The methods proposed are part of the approach that focuses on time series data from OD matrices and sensors, concentrating, in this last case, on collective measures like vehicle counts, occupancy, and speed. In the case of anomaly detection in traffic patterns, this approach, rooted in data, marks a significant shift from earlier pattern-matching methods like the California and McMaster algorithms. These traditional methods, foundational in their time, often struggled with calibration issues and inflexible threshold settings.

Recent developments have seen clustering techniques and Gaussian Mixture Hidden Markov Models gain traction in differentiating irregular congestion and pinpointing traffic anomalies. These

methods take a more detailed perspective of traffic behaviour, moving beyond the constraints of older models. Also, the combination of machine learning techniques and statistical methods, such as the boosting-GCRF, have proved to be robust under demand anomalies.

Deep learning techniques, especially LSTM-based models, have also emerged as a powerful approach. Models such as convolutional-LSTM models allow blending traffic data with additional factors like weather conditions, and LSTM encoder-predictor models combine prediction and reconstruction tasks, being able to better capture the dynamics of the time series and achieving accurate multi-step forecasting. Yet, these sophisticated models sometimes face challenges related to computational demands and the quality of data, highlighting the need for robust and scalable solutions.

The proposed anomaly detection methods integrate these advancements, adopting a data-driven and parameter-free approach similar to the SND methodology. This strategy combines robust statistical analysis with modern techniques, avoiding the difficulties associated with a lack of labelled data and system calibration.

Future Challenges and Opportunities

Looking ahead, several challenges and opportunities stand out. A major challenge is effectively managing large and complex datasets. Machine learning and deep learning models, particularly LSTM networks and hybrid models, have excelled in predictive accuracy. Nonetheless, making these models scalable and adaptable to diverse and larger datasets is an area that needs more work.

Another challenge is in setting a clear and consistent definition of what an anomaly is. Anomalies are deviations from normal behaviour or expectations. However, in complex systems like traffic networks, what is considered normal can vary significantly based on factors like time of day, location, and external events. The ambiguity in defining anomalies arises because these factors create a wide range of "normal" conditions, making it challenging to pinpoint what is truly abnormal.

To address this, anomaly detection algorithms in traffic management, operations, and control must be adaptive and context aware. They need to consider varying conditions and patterns, adapting their definitions of normal and abnormal accordingly. This requires sophisticated techniques like machine learning and deep learning, which can learn from data and adjust to new patterns over time. The quality of anomaly detection can be quantified by how accurately these algorithms identify real-world issues without producing too many false positives or negatives. The quality of anomaly detection in traffic management systems hinges on accurately defining and recognizing anomalies, which requires understanding the context-specific nature of traffic and mobility patterns and employing advanced, adaptive algorithms capable of learning from diverse data sets.

Present methods vary in their baseline definitions, using different statistical and machine learning tools to spot deviations. A more unified approach could improve how anomaly detection is compared and trusted across various systems and locations.

Opportunities are present in the adoption of new technologies like real-time data processing. These technologies could greatly improve the accuracy and speed of anomaly detection systems. Additionally, integrating various data sources could offer a more complete picture of traffic patterns and transport demand, providing richer information for analysis and decision-making.

The field of anomaly detection in traffic and transport demand is at an exciting turning point. With the rise of smarter cities and autonomous vehicles, sophisticated traffic management systems will become increasingly important. Embracing new technologies and methods while addressing current challenges is key to creating a future with better traffic efficiency and safety.

3 ANOMALY DETECTION IN TRAFFIC PATTERNS

3.1 Understanding the anomalies

The general purpose of anomaly detection in general is widely accepted as finding data that does not conform with the notions of normal behaviour. The main outcome of this problem is to identify such anomalies and provide an appropriate response when such anomalies occur.

Understanding the nature of anomalies is fundamental in their detection, leading to the classification of outliers into three main categories [31]:

1. **Point Anomalies**

   An outlier in this category is a data point lying outside the boundary of the normal region of observations, distinguishing it from regular points.

2. **Contextual Anomalies**

   Anomalies in this category are context-specific, classifiable only within a particular context, hence there are 2 attributes that each data point should contain:

   - ***Contextual Attributes***

     Indicate the context for a given instance. For instance, in time series, time itself serves as a contextual attribute, determining the position of each observation in the sequence.

   - ***Behavioral Attributes***

     Refer to the characteristics of observations not bound by context. For example, in a spatial dataset describing the average consumption of a product worldwide, the amount at any specific location represents a behavior.

3. **Collective Anomalies**

   Outliers of this kind emerge when a collection of related observations is anomalous concerning the entire dataset, even though each data point individually might not be an outlier.

This categorization provides a foundational framework for understanding and classifying anomalies in diverse datasets, offering valuable insights into their distinct characteristics. For our project's purposes, at least in this iteration we are mainly focused on the 3rd category of anomalies.

3.1.1 Types of Traffic Anomalies

Traffic anomalies manifest in diverse forms, each necessitating specific attention and tailored remedial actions. Recognizing the diverse nature of these anomalies is vital for designing robust anomaly detection systems capable of addressing a wide array of real-world scenarios. Some common types of traffic anomalies which can be either due to planned (i.e. scheduled road closures for maintenance purposes) or unplanned (i.e. sudden congestion surges due to unexpected high demand) events and may include amongst others:

1. **Accidents**

   Unexpected collisions or incidents on the road that disrupt the regular flow of traffic and may lead to congestion.

2. **Road Closures**

Planned or unplanned closures of roads due to construction, maintenance, or unforeseen events, impacting traffic routes and flow.

3. **Inclement Weather Events**

   Adverse weather conditions, such as heavy rain, snow, or fog, influencing traffic behaviour and causing unexpected patterns.

4. **Special Events or Gatherings**

   Large-scale events, festivals, or gatherings leading to altered traffic patterns and increased congestion in specific areas.

A keynote here is that for the scope of this deliverable we adopt a macroscopic approach, which means that the objective is not to detect single events like micro collisions, broken down vehicles or lane closures. Our aim is to detect when the collective performance of the inspected network is unusual in some sense.

3.1.2 Anomaly Detection Techniques

Effective anomaly detection in traffic patterns relies on the utilization of a wide range of advanced techniques. In this section, we explore diverse approaches, ranging from traditional statistical methods to machine learning and intricate deep learning. The choice of technique depends on the specific characteristics of the traffic data and ranges from traditional statistical methods that lay the groundwork with mathematical rigor, all the way to machine learning and deep learning approaches with more dynamic capabilities.

Statistical Methods

Statistical methods have long been employed for anomaly detection in traffic patterns. These approaches leverage mathematical models to analyze historical data and identify deviations from expected patterns. Metrics such as mean, standard deviation, and percentile analysis play a crucial role in detecting anomalies by quantifying deviations from the norm.

Z-Score

The Z-score measures how many standard deviations a data point is from the mean. Unusually high or low Z-scores can indicate anomalies in traffic volume or speed.

Gaussian Mixture Models (GMM)

GMM assumes that the data is generated from a mixture of several Gaussian distributions. Deviations from the assumed distributions may indicate anomalies in traffic patterns.

Machine Learning Approaches

Machine learning techniques have gained prominence in anomaly detection due to their ability to discern complex patterns and adapt to dynamic environments. Algorithms such as SVM, K-NN and Random Forests have demonstrated efficacy in learning and identifying anomalies from large and heterogeneous traffic datasets.

Support Vector Machines (SVM)

SVM classifies data by finding the hyperplane that best separates normal data from anomalies in a higher-dimensional space. The SVMs have been used for their ability to solve the problem of traffic incident detection, because it is adapted to produce a nonlinear classifier with maximum generality, and it has exhibited good performance as neural networks [32].

OneClassSVM

OneClassSVM specializes in detecting anomalies by creating a model that identifies the normal data distribution. It operates by finding a decision boundary that separates the majority of data points (seen as normal) from outliers. This approach is particularly effective in situations with a clear distinction between normal and anomalous data, making it a useful tool in traffic monitoring systems to identify unusual patterns or incidents.

Local Outlier Factor (LOF)

LOF is an algorithm designed for anomaly detection, focusing on the local density deviation of a given data point with respect to its neighbours. It calculates the local density of each point, comparing it to the densities of its neighbours to identify regions of similar density and points that are significantly different. LOF is adept at recognizing anomalies in varied data densities, which is beneficial in traffic systems for detecting irregularities in traffic flow or behaviour.

Isolation Forest

This method is based on characterizing anomalous traffic conditions by exploiting the fact that anomalies tend to be isolated. The most remarkable feature of this anomaly detection method is its high detection performance while having a very simple tuning procedure and an extremely low computational demand. [33]

Deep Learning Approaches

Deep Learning has emerged as a powerful tool for anomaly detection, particularly in handling intricate patterns within vast datasets. Deep Neural Networks, Convolutional Neural Networks, and RNN excel in capturing intricate relationships within traffic data. Deep learning approaches contribute significantly to the identification of anomalies by extracting high-level features and representations.

Autoencoders

Autoencoders are unsupervised machine learning models, specifically neural networks, which extract nonlinear features of traffic flow data that aim to reconstruct the input data. Anomalies are detected based on the reconstruction error between the input and the reconstructed data [34].

RNNs

RNNs are effective for processing sequential data and can be employed to model temporal dependencies in traffic patterns, aiding in anomaly detection.

Hybrid Approaches

Hybrid anomaly detection methods integrate multiple techniques to harness the strengths of different approaches. Combining statistical models with machine learning algorithms or incorporating rule-based systems enhances the overall detection accuracy [35]. Hybrid approaches offer flexibility and robustness, making them well-suited for addressing various challenges in anomaly detection.

By utilizing these diverse anomaly detection techniques, it becomes feasible to design a comprehensive anomaly detection system capable of accurately identifying and responding to anomalies within traffic patterns. A keynote here that we need to take into consideration is identifying which one of the anomaly detection techniques fits best to our unique case, because as in Occam's razor principle "The simplest solution is most often the best".

3.1.3 Evaluation Metrics and Performance Measures

Effective assessment of anomaly detection techniques requires a deep understanding of performance metrics and criteria. These metrics serve as the benchmarks for evaluating the accuracy and reliability of systems designed to identify anomalies in complex datasets. In this section, we explore the common metrics and criteria employed in the evaluation of anomaly detection methodologies. By comprehensively examining precision, recall, F1-score, AUC-ROC curves, sensitivity, specificity, accuracy, False Positive Rate (FPR), and True Positive Rate (TPR), we aim to unravel the intricate tapestry that defines the success of anomaly detection systems. Each metric contributes a unique perspective, shedding light on different aspects of performance, and collectively, they form the basis for rigorous evaluation and comparison.

Performance Evaluation Criteria and Common Metrics:

To quantify the effectiveness of anomaly detection algorithms, comprehensive performance evaluation criteria are essential. Key metrics such as sensitivity, specificity, and accuracy serve as benchmarks, quantifying the system's prowess in correctly identifying both anomalies and normal patterns. The critical parameters of FPR and TPR play a pivotal role in assessing the robustness and reliability of anomaly detection algorithms.[1]

Most Common Metrics for Anomaly Detection:

Within the literature, a diverse array of metrics exists to assess anomaly detection algorithms, encompassing precision, recall (or detection rate), and F1-score. While various definitions of these measures are present, we outline the most frequently referenced definitions, some of which are employed in this report to evaluate the developed models.

Precision denotes the ratio of correctly predicted positive observations to the total predicted positive observations (incidents). Mathematically, precision is expressed as:

$$Precision = \frac{Number\ of\ correctly\ detected\ incidents}{Total\ number\ of\ samples\ predicted\ as\ incidents}$$

Recall, also known as the detection rate or TPR, signifies the ratio of correctly predicted positive observations to all observations in the actual class. In the context of incident detection, a model is considered to correctly detect an incident if an alert was raised at any point during the incident. The recall formula is given by:

$$Recall = \frac{Number\ of\ correctly\ detected\ incidents}{Total\ number\ of\ actual\ \ incidents\ in\ the\ dataset}$$

[1] True Positive Rate (TPR) and False Positive Rate (FPR) Definitions:

True Positive Rate (TPR): also known as sensitivity or recall, measures the system's aptitude in correctly identifying positive instances among the actual positives. Mathematically, TPR is expressed as TPR = TP / (TP + FN).

False Positive Rate (FPR): evaluates the system's tendency to incorrectly label negative instances as positive. It is calculated as FPR = FP / (FP + TN). Striking a balance between TPR and FPR is paramount for achieving optimal performance in anomaly detection.

3.1.4 Challenges in Anomaly Detection

Anomaly detection as every other problem comes with its set of challenges that demand careful consideration. In this section, we focus into three primary challenges: imbalanced data, model sensitivity, and scalability. Each challenge poses unique hurdles, from handling unevenly distributed datasets to fine-tuning model responsiveness and addressing scalability concerns. Understanding and mitigating these challenges are pivotal steps toward enhancing the robustness and reliability of anomaly detection methodologies.

Model Sensitivity

The sensitivity of anomaly detection models to variations in data patterns poses another significant challenge. Sensitivity refers to the model's responsiveness to subtle changes, making it prone to false positives or negatives. Achieving an optimal balance in model sensitivity is crucial for discerning genuine anomalies while avoiding excessive false alarms.

Scalability

Scalability emerges as a challenge when deploying anomaly detection systems to handle large-scale datasets or real-time streams of data. As the volume of data increases, traditional models may struggle to maintain efficiency and timeliness. Developing scalable solutions involves leveraging parallel processing, distributed computing, or selecting algorithms inherently designed for scalability.

Imbalanced data

One of the foremost challenges in anomaly detection stems from imbalanced datasets, where the instances of normal patterns significantly outweigh those of anomalies. In such cases, models can achieve high accuracy simply by predicting the majority class (usually the normal cases), while failing to effectively detect the rare anomalies, which are often the most critical to identify.

Addressing this challenge necessitates techniques such as oversampling, under sampling, or the application of specialized algorithms designed for imbalanced scenarios.

Balancing the dataset can in some cases also help in such scenarios, but it is important to approach it carefully, especially since anomalies are inherently rare in real-world scenarios. Here is a sample of the most commonly used approaches:

- **Oversampling the Minority Class**: Increasing the number of anomaly cases in our training set. This can be done using techniques like SMOTE (Synthetic Minority Over-sampling Technique). However, we ought to be cautious as oversampling can lead to overfitting in the minority class.
- **Under sampling the Majority Class**: Reducing the number of normal cases in our training set. This should be done carefully to ensure that the remaining normal cases are still representative of the overall distribution.
- **Anomaly Detection-Specific Methods**: Instead of traditional classification models, we can also consider using algorithms specifically designed for anomaly detection, which are often better suited for handling imbalanced datasets. Algorithms like Isolation Forest or LOF are examples that could be used to address the aforementioned issue.
- **Adjusting Class Weights**: For some models, we can adjust the class weights to make the model pay more attention to the minority class. This can be done by assigning a higher weight to the anomaly class.

3.2 Evaluation Metrics:

Lastly, we can use evaluation metrics that are more informative for imbalanced datasets, such as precision, recall, F1-score, and AUC-ROC, rather than accuracy. Methodology and Implementation

This section outlines the chosen methodology for developing the Advanced Anomaly Detection framework within the CONDUCTOR project, elaborating on the essential techniques and strategies. It also provides the necessary background for understanding how anomaly detection is uniquely applied and integrated into the broader scope of the CONDUCTOR project's objectives.

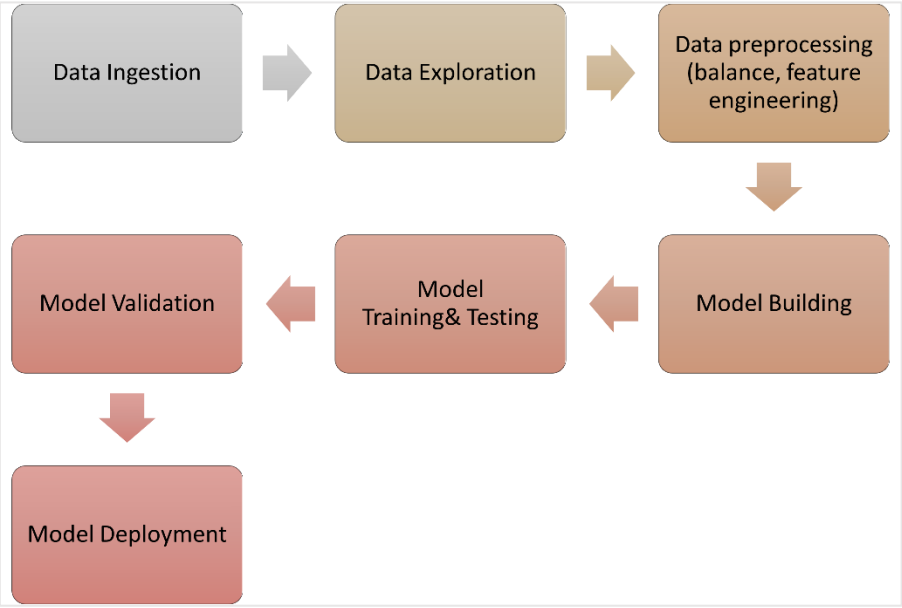

Figure 1 Steps to Anomaly Detection Service Implementation

3.2.1 Data Ingestion and Preprocessing

3.2.1.1 Data Ingestion

Data Ingestion is crucial for the implementation of all the subsequent services that will be developed as part of the CONDUCTOR project. The gathering of data from various sources to create multi-dimensional data points, known as vectors, is key for the services to be optimized. These data can be used immediately or stored for future access and utilization. The foundation of any analytics and machine learning application is rooted in data ingestion. This process can occur in real-time, continuously streaming data from its sources, or through periodic imports of data in either large batches or smaller, more frequent micro-batches, but for further details on this phase of the project, refer to the CONDUCTOR deliverable D3.1(Specification and initial version of data gathering, harmonization, fusion and analysis techniques).

In the broader CONDUCTOR architecture, the data ingestion on the Anomaly Detection service will be implemented through an API endpoint, exposed from the CCAM and traffic data space. Since we are in an earlier stage of the implementation process, using an AGILE approach on the whole spectrum of the project, we are going to ingest the data directly from the source, which is a Greek Governmental site (data.gov.gr). This process will be substituted by a direct ingestion from the CONDUCTOR data space as the project matures.

3.2.1.2 Data Exploration and Preprocessing

This step of the process is essential, as we gather the first insights from the acquired data. This stage of the process is, in a majority of the time, one of the most time-consuming. In cases where a data harmonization stage has occurred beforehand and the data are well defined and understood, this stage can be skipped. It is essential though for the user to have a complete understanding of the data types and characteristics before moving to the next step (preprocessing), which is the following step of the process.

The first step of the data pipeline after the data is ingested in the system is the data preprocessing stage, where a series of techniques are implemented in order to cleanse, integrate and transform the data to the desired form. This has to be performed to improve the overall data quality, which could later on prove to be substantial in the performance and efficiency of our models.

The final step in the pre-processing phase is feature engineering, a critical process that involves creating meaningful variables from the raw data. This step requires both creativity and domain knowledge, as it involves selecting those aspects of the data that are most relevant to the problem at hand. By transforming and combining the raw data into features that can better represent the underlying patterns, we significantly enhance the predictive power of our models. This is one of the key aspects of why a “rich” dataset is important in the process, as it allows better feature engineering, which not only aids in improving model accuracy but also plays a pivotal role in making the models more interpretable and aligned with the specific aspects of the problem at hand.

3.2.1.3 Model Selection

In the model selection phase, there are three main steps that need to be implemented to ensure a successful selection of the most appropriate models. Those are: 1) **Model Building**, 2) **Model Training**, and 3) **Model Validation and Evaluation**.

Model Building

The building phase in model selection is a cornerstone of the Anomaly Detection service in the CONDUCTOR project. This stage involves the careful selection of appropriate models that align with the unique requirements and characteristics of traffic anomaly detection. Given the complexity and variability of traffic data, Frontier focuses on selecting models that are not only robust in handling large datasets but also sensitive enough to accurately detect subtle anomalies. This includes considering approaches that span from rule-based approaches such as Anomaly Detection Toolkit to various machine learning and deep learning architectures, each with their specific strengths. Models like Isolation Forest, SVM, OneClassSVM, LOF and neural networks will be evaluated for their suitability. The aim is to ensure that the chosen models can effectively handle the intricacies of traffic data, such as seasonal variations, peak hour patterns, and unexpected incidents.

Model Training

Once the models are built, the next critical step is model training. This phase involves feeding the pre-processed and feature-engineered data into the models. The training process is iterative, where models learn to distinguish between normal traffic behavior and anomalies. An essential aspect of this phase is parameter tuning, where we adjust various settings within each model to optimize performance. This might include tuning the learning rate in neural networks or the depth of trees in random forest models. Regularization and data balancing techniques will also be employed to prevent overfitting, ensuring that our models generalize well to new, unseen data.

Model Validation and Evaluation

The final and perhaps most crucial phase in the model selection process is validation and evaluation. This step involves testing the trained models on a separate dataset that was not used during the

training phase. It's a critical step to assess how well the model performs in a real-world scenario. We employ a range of metrics to evaluate the model's performance, including precision, recall, F1-score, and the area under the ROC curve (AUC-ROC). These metrics will help us understand not just the accuracy of our models but also their ability to minimize false positives and false negatives – a key requirement in anomaly detection. Additionally, we plan to use confusion matrices to gain a clearer understanding of the model's performance across different classes. The evaluation phase will guide us in selecting the best-performing model or ensemble of models for deployment in the CONDUCTOR ecosystem.

3.3 Implementation and Preliminary Results

As part of this deliverable, and since at the time of writing the implementation phase of the project has not yet started, FI has decided to use the Greek use case (Athens) as a basis for an initial implementation of the Anomaly Detection module, mainly for two reasons. The first reason being to showcase the possible capabilities of such service and its overall added value to the CONDUCTOR project and the second being to also showcase the importance of the data availability, that is of utter most importance for the implementation of the best anomaly detection techniques that are described in previous chapters.

3.3.1 Technical Architecture

At the moment, with the project being at an early technical development phase, FI has created an initial technical architecture of the Anomaly Detection component. This architecture uses as a Data Layer only the datasets used for the implementation of the initial models, results of whom will be presented onwards on the deliverable. The architecture will become more mature as the overall project moves into a more mature development phase.

The Anomaly Detection module currently uses loop detector historical data, obtained from the Greek Government that was mentioned previously, containing timeseries data for the whole Attika peninsula and a set of synthetic data for anomalies. Later on, the data used in this module for the Greek use case will be enriched from different sources with one of the most important ones being the geo-based data from Athen's operator OASA.

As specified in the project's Grant Agreement, OASA will define the specific scenarios to be executed in the framework of this use case and provide data (supply and demand) for the transit system. NTUA will select project solutions and relevant case scenarios to be simulated within the Athens testbed. NTUA will also provide, calibrate, and integrate solutions in the Athens testbed and execute the experiments, part of which requires input from the Anomaly Detection module (Task 3.5).

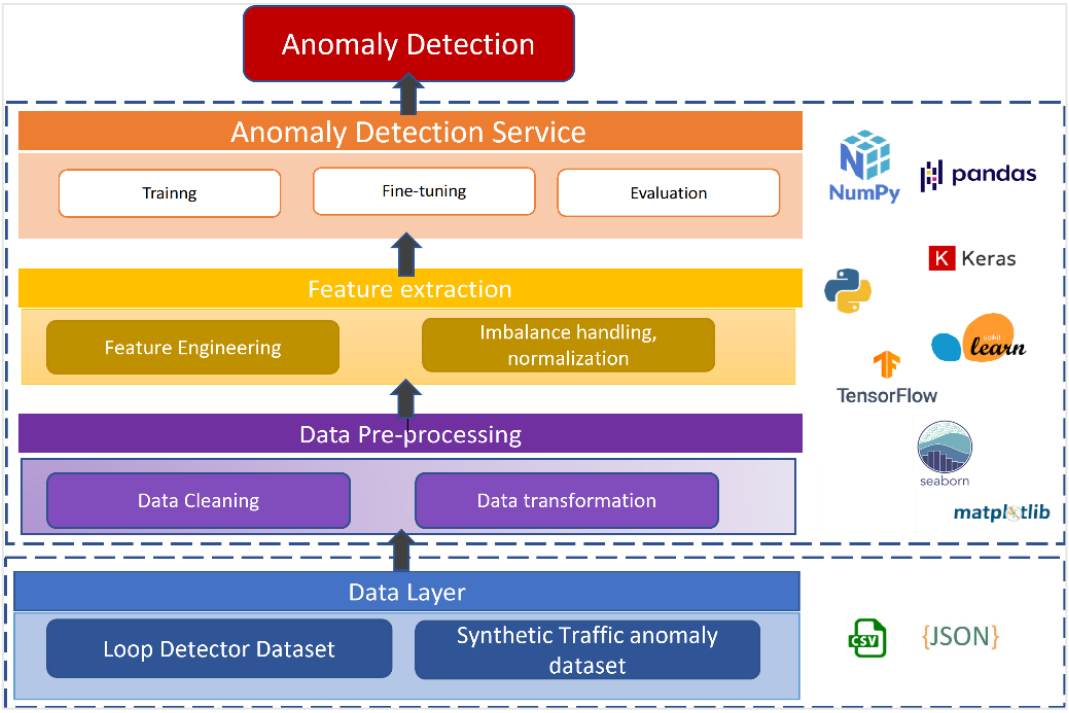

Figure 2 Initial Technical Architecture of Anomaly Detection module

For the initial technical implementation of the module, the programming language Python is used as the main development language. In particular several python libraries have been incorporated into the scripts such as: pandas, scikit-learn, TensorFlow, NumPy, pycarret, keras for data manipulation, modeling and, matplotlib, seaborn, statsmodels.graphs for visualization.

3.3.2 Data

In this section, we will explain the data sources used and the data manipulation process before providing the final datasets to the models.

3.3.2.1 Inductive Loop Detectors (ILDs)

ILDs are a common technology in traffic data gathering, valued for their ability to measure essential traffic parameters such as speed, volume, occupancy, density, queue, and location. However, it's important to note that ILDs, despite their advantages, have known issues with reliability and accuracy. This is partly due to their extensive use over the decades without consistent maintenance or replacement. Thus, while ILDs provide valuable traffic insights, their potential malfunctions and inaccuracies should be carefully considered in traffic monitoring, management, and planning and an analysis of the data is essential before use. Those parameters include among others:

- **Speed**: This parameter according to the data providers claims that it measures the velocity at which vehicles are traveling over a specific section of the road. It is usually expressed in kilometers per hour (km/h) or miles per hour (mph). Speed data is crucial for understanding traffic flow and identifying potential congestion or hazardous conditions, but most ILDs typically estimate vehicle speeds under restrictive conditions and while they can provide valuable insights, the specific measurement may prove to be less accurate than the actual.

- **Volume**: This refers to the number of vehicles passing a point on a roadway over a specified period. It is a fundamental measure of road usage and is often used to analyze traffic load and to plan for road capacity and maintenance.
- **Occupancy**: This parameter indicates the proportion of time that a point on the road is occupied by vehicles. It is expressed as a percentage and is used to estimate how much of the road space is being utilized at any given time. High occupancy rates can signal heavy traffic or congestion.
- **Density**: Traffic density is the number of vehicles occupying a certain length of the road at a given time. It is typically measured as vehicles per kilometer or mile. This measure helps in assessing the level of congestion on a road segment.
- **Queue**: This refers to a line of vehicles waiting, often at traffic signals, toll booths, or other points of delay. Queue length can be an important indicator of congestion levels and the efficiency of traffic control measures at intersections.
- **Location**: While not a traffic flow characteristic like the others, the location parameter in traffic data collection refers to the specific point or segment of the roadway where the data is being collected. Accurate location data is critical for correlating traffic parameters to specific parts of the road network.

3.3.2.2 Location of ILD Sensors

For this specific demonstration, as we have already mentioned the data are collected from ILD sensors from the Attica peninsula and are provided through the use of an API, from the Greek Government as a semi-open dataset (token authentication required).

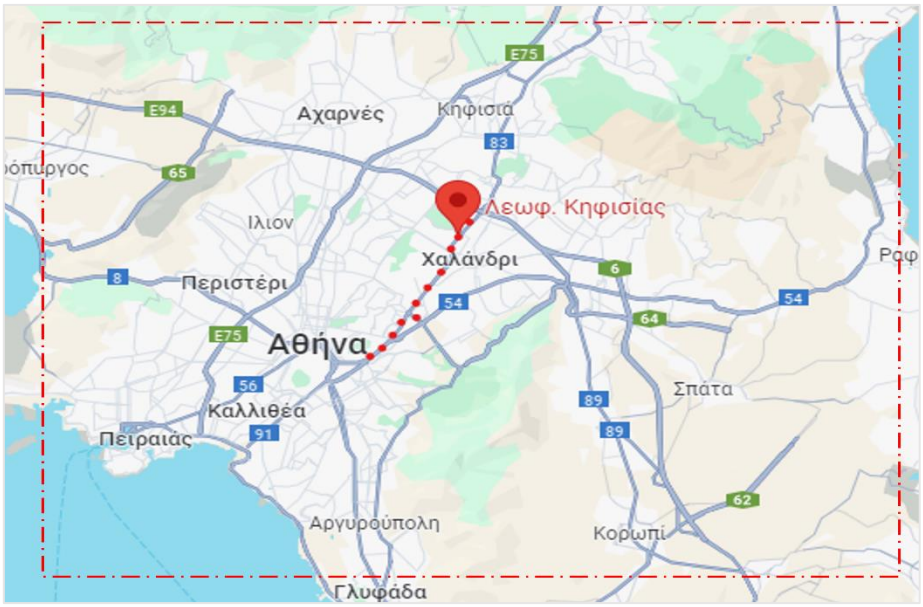

Figure 3 Athens ILD locations

As shown in the figure above (Figure 3), in the boxed area, the data come from all over Attika peninsula in 92 main roads. In the above figure a set of red marks has been added to showcase the areas that will be used for the purposes of the initial implementation.

3.3.2.3 Raw Data Characterstics

As a reference and to train the anomaly detection models, FI will use a set that contains 3 months' worth of data, for specific boulevards of Athens, that are usually high in terms of traffic flow, therefore have a higher chance of experiencing anomalies. Data from the ILDs comes in half hour intervals and from 482 different sensors spread all over Attica. The dataset used does not include a labeled column for actual anomalies, therefore for demonstration purposes, a synthetic dataset will be used, which will be substituted by a dataset with historical data that will include actual anomalies detected. The synthetic data is created using keeping the original dataset's distributions, fluctuated by a set number of standard deviations, which are identified as anomalies.

	appprocesstime	average_speed	countedcars	deviceid	road_info	road_name
0	2022-01-01T18:00:00	102.605381	17840	MS106	ΚΥΡΙΟΣ ΔΡΟΜΟΣ ΜΕ ΚΑΤΕΥΘΥΝΣΗ ΛΑΜΙΑ ΠΡΙΝ ΑΠΟ ΤΗ ...	Λ. ΚΗΦΙΣΟΥ
1	2022-01-01T18:00:00	86.439926	10820	MS857	ΚΥΡΙΟΣ ΔΡΟΜΟΣ ΜΕ ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ, 125 Μ...	Λ. ΒΟΥΛΙΑΓΜΕΝΗΣ
2	2022-01-01T18:00:00	59.667737	12460	MS860	ΚΥΡΙΟΣ ΔΡΟΜΟΣ ΜΕ ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ, 160 Μ...	Λ. ΒΟΥΛΙΑΓΜΕΝΗΣ
3	2022-01-01T18:00:00	42.759036	3320	MS856	ΚΥΡΙΟΣ ΔΡΟΜΟΣ ΜΕ ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ Λ. ΒΟΥΛΙΑΓΜΕΝ...	ΑΛΙΜΟΥ
4	2022-01-01T18:00:00	78.146067	7120	MS865	ΚΥΡΙΟΣ ΔΡΟΜΟΣ ΜΕ ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΒΟΥΛΙΑΓΜΕΝΗ, ...	Λ. ΒΟΥΛΙΑΓΜΕΝΗΣ
5	2022-01-01T18:00:00	60.699275	11040	MS855	ΚΥΡΙΟΣ ΔΡΟΜΟΣ ΜΕ ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ, 200 Μ...	Λ. ΒΟΥΛΙΑΓΜΕΝΗΣ
6	2022-01-01T18:00:00	79.095506	7120	MS866	ΚΥΡΙΟΣ ΔΡΟΜΟΣ ΜΕ ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ, 120 Μ...	Λ. ΒΟΥΛΙΑΓΜΕΝΗΣ
7	2022-01-01T18:00:00	57.990610	8520	MS854	ΚΥΡΙΟΣ ΔΡΟΜΟΣ ΜΕ ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΒΟΥΛΙΑΓΜΕΝΗ, ...	Λ. ΒΟΥΛΙΑΓΜΕΝΗΣ
8	2022-01-01T18:00:00	67.893151	7300	MS867	ΚΥΡΙΟΣ ΔΡΟΜΟΣ ΜΕ ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΒΟΥΛΙΑΓΜΕΝΗ, ...	Λ. ΒΟΥΛΙΑΓΜΕΝΗΣ
9	2022-01-01T18:00:00	58.767790	10680	MS853	ΚΥΡΙΟΣ ΔΡΟΜΟΣ ΜΕ ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ, 200 Μ...	Λ. ΒΟΥΛΙΑΓΜΕΝΗΣ
10	2022-01-01T18:00:00	81.507353	5440	MS868	ΚΥΡΙΟΣ ΔΡΟΜΟΣ ΜΕ ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ, 110 Μ...	Λ. ΒΟΥΛΙΑΓΜΕΝΗΣ
11	2022-01-01T18:00:00	36.842105	760	MS852	ΚΥΡΙΟΣ ΔΡΟΜΟΣ ΜΕ ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΑΛΙΜΟ, 150 Μ....	ΕΘΝ. ΜΑΚΑΡΙΟΥ
12	2022-01-01T18:00:00	51.250712	7020	MS869	ΚΥΡΙΟΣ ΔΡΟΜΟΣ ΜΕ ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΒΟΥΛΙΑΓΜΕΝΗ, ...	Λ. ΒΟΥΛΙΑΓΜΕΝΗΣ
13	2022-01-01T18:00:00	62.931159	11040	MS851	ΚΥΡΙΟΣ ΔΡΟΜΟΣ ΜΕ ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ, 300 Μ...	Λ. ΒΟΥΛΙΑΓΜΕΝΗΣ
14	2022-01-01T18:00:00	69.842466	2920	MS870	ΚΥΡΙΟΣ ΔΡΟΜΟΣ ΜΕ ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ, 150 Μ...	Λ. ΒΟΥΛΙΑΓΜΕΝΗΣ
15	2022-01-01T18:00:00	66.008475	9440	MS850	ΚΥΡΙΟΣ ΔΡΟΜΟΣ ΜΕ ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΒΟΥΛΙΑΓΜΕΝΗ, ...	Λ. ΒΟΥΛΙΑΓΜΕΝΗΣ
16	2022-01-01T18:00:00	53.338583	2540	MS871	ΚΥΡΙΟΣ ΔΡΟΜΟΣ ΜΕ ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ Λ. ΒΟΥΛΙΑΓΜΕΝ...	ΔΙΓΕΝΗ/ΚΑΛΥΜΝΟΥ
17	2022-01-01T18:00:00	61.004016	9960	MS848	ΚΥΡΙΟΣ ΔΡΟΜΟΣ ΜΕ ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ, 150 Μ...	Λ. ΒΟΥΛΙΑΓΜΕΝΗΣ
18	2022-01-01T18:00:00	64.000000	4600	MS872	ΛΩΡΙΔΑ ΑΡΙΣΤΕΡΗΣ ΣΤΡΟΦΗΣ ΑΠΟ Λ. ΒΟΥΛΙΑΓΜΕΝΗΣ Π...	Λ. ΒΟΥΛΙΑΓΜΕΝΗΣ
19	2022-01-01T18:00:00	58.232000	10000	MS847	ΚΥΡΙΟΣ ΔΡΟΜΟΣ ΜΕ ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΒΟΥΛΙΑΓΜΕΝΗ, ...	Λ. ΒΟΥΛΙΑΓΜΕΝΗΣ

Figure 4 Raw Data in Pandas Data Frame Format

The dataset shown in the figure above (Figure 4) is registering measurements of average speed, average flow (countedcars), sensor id (deviceid), as well as road information (road_info) about the geographical position of the respective sensor and the main road it belongs to (road_name). It is apparent that many of the ILD parameters are missing from this specific dataset, hence a good indication that the models will have room for improvement with the acquisition of better datasets from OASA.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 833642 entries, 0 to 833641
Data columns (total 6 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   appprocesstime  833642 non-null  object
 1   average_speed   833642 non-null  float64
 2   countedcars     833642 non-null  int64
 3   deviceid        833642 non-null  object
 4   road_info       831609 non-null  object
 5   road_name       833642 non-null  object
dtypes: float64(1), int64(1), object(4)
```

	average_speed	countedcars
count	833642.000000	833642.000000
mean	46.923661	33109.071754
std	20.270304	33801.806017
min	0.000000	0.000000
25%	34.699680	6240.000000
50%	46.986377	22240.000000
75%	57.508731	50760.000000
max	148.000000	315360.000000

Figure 5 Missing Values from the Raw Data and Descriptive Statistics

Although some basic parameters are missing, the dataset is fairly reliable (see Figure 5), from 2022 onwards, having a mere 0.2% missing values, only in a specific column (road_info), for the time period under examination. The low percentage of missing values saves us, for this particular iteration, having to perform data imputation. Although not needed our module will include different

imputation methods such as: polynomial, free-flow imputation, spatial K-NN, weekday-based imputation to be utterly proofed for the future datasets that will be ingested.

3.3.2.4 Further Data Processing and Transformation

To generate higher quality results, there is a need for data filtering and transformation in order to emphasize the most important aspects of the data. Thus, we decided to proceed for this iteration by creating a more specific dataset that will explore the capabilities of the Anomaly Detection module, on a specific area, using two consequent sensors in one of the most congested roads of our use case.

Examining the anomalies for a specific sensor instead of a broader network allows for detection of localized incidents or anomalies, thus can be more accurate in identifying small-scale variations in traffic patterns. Using this approach though, we might miss broader patterns affecting the entire road segment and it is also more complex to integrate the findings across multiple sensors.

Using combined analysis for a set of sensors on the other hand, can be better for understanding the overall traffic flow and patterns and simplifies the whole process as the models treat the whole road segment as a single entity. This approach could dilute or miss localized incidents and might be less sensitive to small-scale anomalies.

Having decided to proceed with the first approach we isolated two different sensors with device id MS259, MS261, which are located on Leoforos Kifissia's, one of the high-volume traffic avenues of Athens. First, we preprocessed the data and performed some exploratory analysis, exploring the data for each sensor (MS259 and MS261) separately.

3.3.3 Exploratory Data Analysis

The first step to better understanding the data that we are going to use to perform Anomaly Detection is to first provide a basic visual exploration, that can be enriched into a deeper analysis if we consider examining patterns by time of day or day of the week.

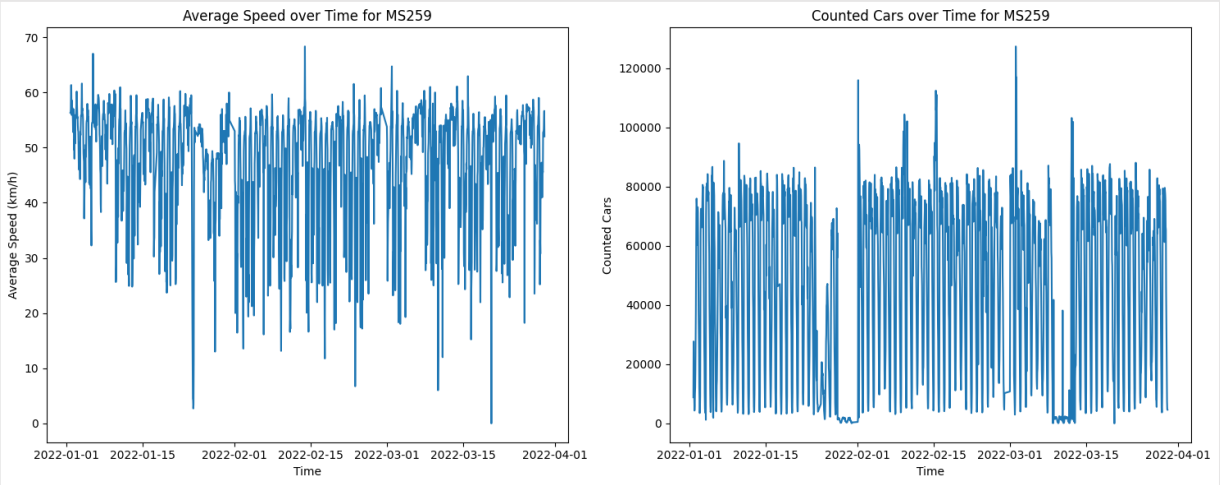

Figure 6 Visual Representation of Single Sensor Data (MS259)

As we can see in the above figure (Figure 6), the first plot (left side) shows the average speed of vehicles per half hour for the specified detector (MS259), and the second plot (right side) displays

the number of counted cars for the set period. These plots are essential for initial exploratory analysis and can help in identifying patterns, trends, and potential anomalies in the data.

The heatmap displayed below (Figure 7) is also one more of the first visualizations in the exploratory analysis phase, that can clearly indicate anomalies or even missing values in the dataset. This heatmap displays information regarding the average counted cars for a set period, each hour of the day. Since in the exploration part of the dataset, we did not identify any missing values in the specific column (countedcars), we can safely assume the all the deep green (<20000) color of the heatmap indicates an unusually low count of cars for a set period of 3 days.

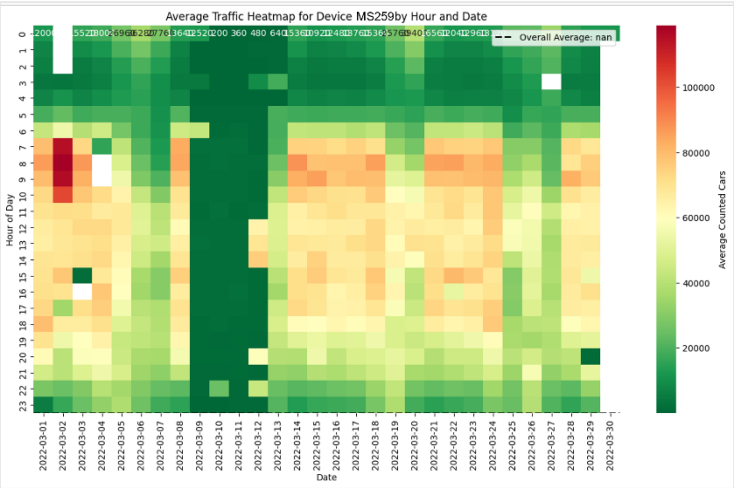

Figure 7 Heatmap indicating usually high-low num of vehicles

3.3.4 Anomaly Detection (Statistical Approach)

Having an initial visual display helped us in a better understanding of the data, hence we moved to the first Anomaly Detection approach.

3.3.4.1 Anomaly Detection using simple thresholds

For this, we used a fairly simple yet widely used statistical approach to identify data points that significantly deviated from the typical traffic pattern. We focused both on the average speed and the number of vehicles over time.

This method involved calculating a threshold. The most widely used threshold in the literature was (mean ± 2 standard deviations), so we proceeded using this for identifying the anomalies. This is a common approach for anomaly detection in time series data.

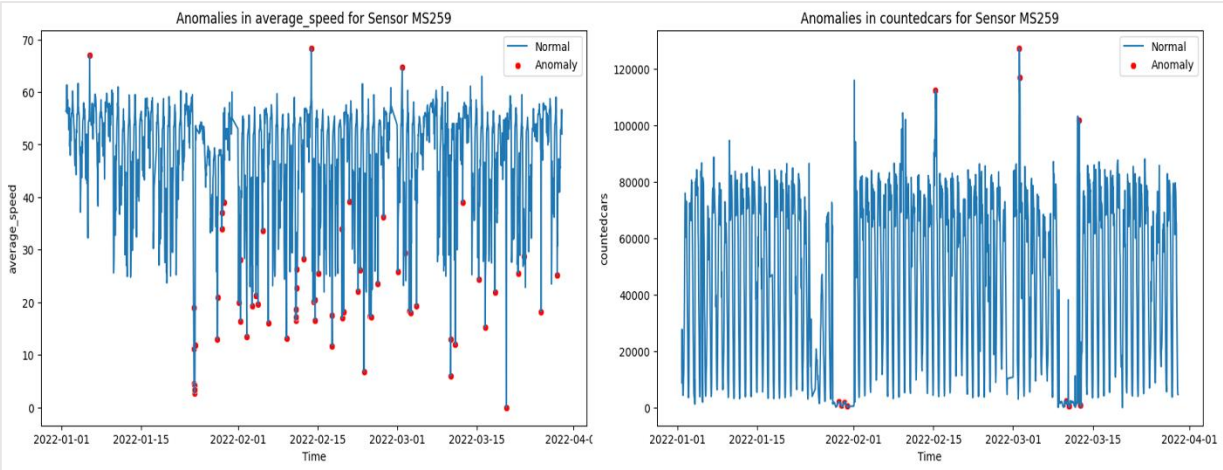

Figure 8 Anomaly Detection using simple statistical thresholds

The above plots (Figure 8) illustrate the anomaly detection results for one of the sensors that we isolated (MS259), for both parameters of average speed and number of vehicles. The red dots represent the anomalies detected in the data while the blue line plot shows the normal traffic pattern over time. As can be inferred from the plot (left side) there are certain periods where the average speed significantly deviates from the norm. Also, there are times when the number of cars is unusually high or low (right side).

The key takeaways from this approach are that these anomalies could indicate various incidents or unusual traffic conditions, such as accidents, road closures, or unexpected events affecting the overall traffic of the network. These anomalies would also be of interest for further investigation, especially if we could correlate them with external events or data (like weather conditions, local events, and roadworks).

3.3.4.2 Anomaly Detection using advanced thresholds

While the first approach yielded some useful results, we decided to proceed in a more advanced statistical approach, to make our anomaly detection even more accurate by making it context-aware, based on time and day of the day.

To do so we followed a more complex set of rules. As a first step we once more sorted the data into chronological order, but now we also created a new data frame that contained the average values for each set period of observation across the whole timespan of the dataset. Afterwards we proceeded to identify anomalies by comparing individual data points against the average value for their respective hours that are set as the new threshold. In this case anomalies are identified as values that deviate more than 20% from the hourly averages.

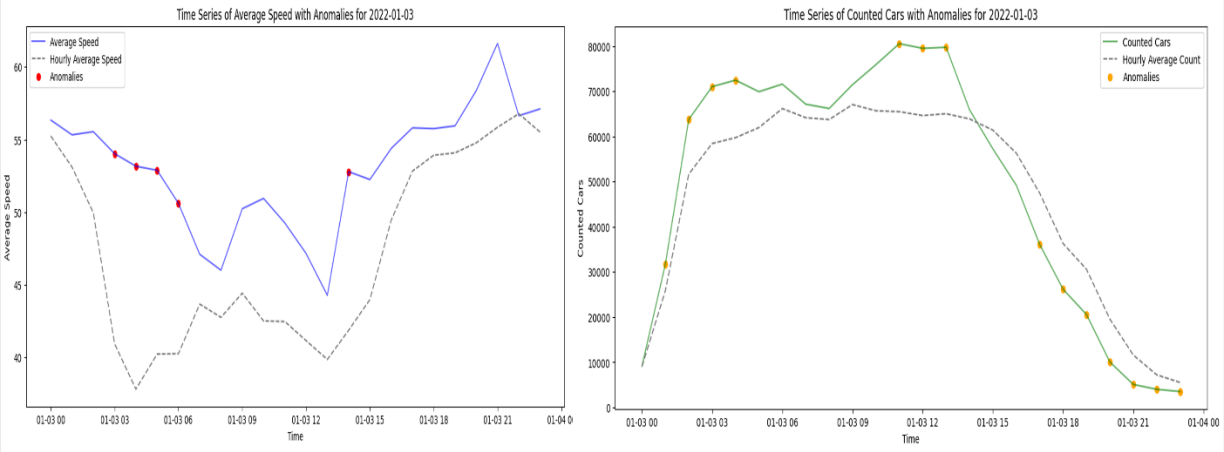

Figure 9 Anomaly Detection using advanced statistical thresholds

In the figure above (Figure 9) we showed the results of the advanced statistic threshold by focusing on a single-day period for clearer visualization of the anomalies.

On the first plot (left side) we can see the anomalies detected in comparison to the specific hourly average speed for the whole set period of the dataset. All the anomalies detected indicate a higher than usual average speed, which in turn might be an indication of unexpected low traffic. On the second one (right side) we can identify the anomalies regarding the average count of vehicles on an hourly basis. The anomalies detected in this plot show an unusual number of cars (both significantly lower and higher than average), in different periods of the day.

The red and yellow dots of the plots represent anomalies, the blue and green line plots represent the normal average speed and vehicle count while the dashed lines on each plot represent the hourly averages for the whole period under study.

Using this approach, we created two new columns on our data frame indicating if there is either a speed or a vehicle counts anomaly (Figure 10).

	apprecesstime	average_speed	countedcars	deviceid	road_info	road_name	hour	average_speed_avg	countedcars_avg	speed_anomaly	carcount_anomaly
0	2022-01-01 18:00:00	56.318182	8800	MS259	ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ ΠΡΙΝ ΑΠΟ ΤΗΝ ΑΡΚΑΔΙΑΣ	Λ. ΚΗΦΙΣΙΑΣ	18	53.640208	39422.456140	False	True
1	2022-01-01 18:00:00	56.804878	9840	MS261	ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ ΠΡΙΝ ΑΠΟ ΤΗΝ ΠΑΝΟΡΜΟΥ	Λ. ΚΗΦΙΣΙΑΣ	18	53.640208	39422.456140	False	True
2	2022-01-02 18:00:00	55.074434	37080	MS261	ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ ΠΡΙΝ ΑΠΟ ΤΗΝ ΠΑΝΟΡΜΟΥ	Λ. ΚΗΦΙΣΙΑΣ	18	53.640208	39422.456140	False	False
3	2022-01-02 18:00:00	55.957071	31680	MS259	ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ ΠΡΙΝ ΑΠΟ ΤΗΝ ΑΡΚΑΔΙΑΣ	Λ. ΚΗΦΙΣΙΑΣ	18	53.640208	39422.456140	False	False
4	2022-01-03 18:00:00	55.752294	26160	MS259	ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ ΠΡΙΝ ΑΠΟ ΤΗΝ ΑΡΚΑΔΙΑΣ	Λ. ΚΗΦΙΣΙΑΣ	18	53.640208	39422.456140	False	True
...	...	...	...	...	...	...	...	...	...	...	...
4048	2022-03-27 17:00:00	55.562863	62040	MS261	ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ ΠΡΙΝ ΑΠΟ ΤΗΝ ΠΑΝΟΡΜΟΥ	Λ. ΚΗΦΙΣΙΑΣ	17	51.911187	51727.100592	False	False
4049	2022-03-28 17:00:00	50.825871	56280	MS259	ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ ΠΡΙΝ ΑΠΟ ΤΗΝ ΑΡΚΑΔΙΑΣ	Λ. ΚΗΦΙΣΙΑΣ	17	51.911187	51727.100592	False	False
4050	2022-03-28 17:00:00	51.260037	64760	MS261	ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ ΠΡΙΝ ΑΠΟ ΤΗΝ ΠΑΝΟΡΜΟΥ	Λ. ΚΗΦΙΣΙΑΣ	17	51.911187	51727.100592	False	True
4051	2022-03-29 17:00:00	35.009659	78680	MS261	ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ ΠΡΙΝ ΑΠΟ ΤΗΝ ΠΑΝΟΡΜΟΥ	Λ. ΚΗΦΙΣΙΑΣ	17	51.911187	51727.100592	True	True
4052	2022-03-29 17:00:00	45.526316	65360	MS259	ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ ΠΡΙΝ ΑΠΟ ΤΗΝ ΑΡΚΑΔΙΑΣ	Λ. ΚΗΦΙΣΙΑΣ	17	51.911187	51727.100592	False	True

Figure 10 Data Frame with added anomaly detection columns

3.3.5 Anomaly Detection using a Machine Learning Approach

After the Anomaly Detection using statistical approaches, we proceeded to approach the problem from a machine learning perspective. To do so it was essential to use a dataset that would consist of labelled data for anomalies, which are extremely important for the anomaly detection evaluation of the models, since the labelled data constitute the ground truth on which the performance metrics of our machine learning models are built.

Since no such dataset was available at the time of the first implementation that could be combined with our ILD dataset, we proceeded to create a synthetic dataset, with the anomaly column being based on a fluctuation of deviations from the average speed or the average number of vehicles, since either significant deviation constitutes an anomaly. We then merged the two datasets forming our final dataset that will be used for the anomaly detection algorithms (Figure 11).

appprocesstime	average_speed	countedcars	deviceid	road_info	road_name	is_anomaly
2022-01-01 18:00:00	56.318182	8800	MS259	ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ ΠΡΙΝ ΑΠΟ ΤΗΝ ΑΡΚΑΔΙΑΣ	Λ. ΚΗΦΙΣΙΑΣ	0
2022-01-01 18:00:00	56.804878	9840	MS261	ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ ΠΡΙΝ ΑΠΟ ΤΗΝ ΠΑΝΟΡΜΟΥ	Λ. ΚΗΦΙΣΙΑΣ	0
2022-01-01 19:00:00	56.368497	27680	MS259	ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ ΠΡΙΝ ΑΠΟ ΤΗΝ ΑΡΚΑΔΙΑΣ	Λ. ΚΗΦΙΣΙΑΣ	0
2022-01-01 19:00:00	55.942138	31800	MS261	ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ ΠΡΙΝ ΑΠΟ ΤΗΝ ΠΑΝΟΡΜΟΥ	Λ. ΚΗΦΙΣΙΑΣ	0
2022-01-01 20:00:00	56.969697	14520	MS259	ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ ΠΡΙΝ ΑΠΟ ΤΗΝ ΑΡΚΑΔΙΑΣ	Λ. ΚΗΦΙΣΙΑΣ	0
...	...	...	...	...	...	...
2022-03-29 22:00:00	53.586667	9000	MS261	ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ ΠΡΙΝ ΑΠΟ ΤΗΝ ΠΑΝΟΡΜΟΥ	Λ. ΚΗΦΙΣΙΑΣ	0
2022-03-29 23:00:00	55.207692	5200	MS261	ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ ΠΡΙΝ ΑΠΟ ΤΗΝ ΠΑΝΟΡΜΟΥ	Λ. ΚΗΦΙΣΙΑΣ	0
2022-03-29 23:00:00	56.649573	4680	MS259	ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ ΠΡΙΝ ΑΠΟ ΤΗΝ ΑΡΚΑΔΙΑΣ	Λ. ΚΗΦΙΣΙΑΣ	0
2022-03-30 00:00:00	52.026087	4600	MS259	ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ ΠΡΙΝ ΑΠΟ ΤΗΝ ΑΡΚΑΔΙΑΣ	Λ. ΚΗΦΙΣΙΑΣ	0
2022-03-30 00:00:00	56.570370	5400	MS261	ΚΑΤΕΥΘΥΝΣΗ ΠΡΟΣ ΚΕΝΤΡΟ ΠΡΙΝ ΑΠΟ ΤΗΝ ΠΑΝΟΡΜΟΥ	Λ. ΚΗΦΙΣΙΑΣ	0

Figure 11 Dataset with synthetic anomaly data column

We created the synthetic anomaly data points in a way that would simulate an actual labelled dataset. Meaning that there is a significant imbalance between normal cases and anomalies. Specifically in our case for the example of the isolated sensor MS259 that was one of the sensors we used in this implementation, for a set period (3 months), only 9 observations were labelled as anomalies. This small value makes sense, since we are examining one specific sensor, that is only influenced by deviations on traffic flow, only from specific areas of the whole network.

The imbalance between normal cases and anomalies is a very important point in anomaly detection. This is because in many cases, models can achieve high accuracy simply by predicting the majority class, while failing to effectively detect the rare anomalies, which are often the most critical to identify (the subject of data imbalance is thoroughly explained in chapter 3.1.4).

After the final dataset's creation, we implemented a set of machine learning and deep learning algorithms (OneClassSVM, Isolation Forrest, LOF, LSTM) to conduct a first set of experiments. It is worth mentioning at this point, that due to the data imbalance, we proceeded to use the SMOTE technique, as it is a more refined technique than just oversampling the minority class by duplicating examples, which does not offer any new information to the model. We even experimented with the simultaneous use of different techniques. It is also worth noting that there was also a fine-tuning phase of the algorithms, but due to the limitations of the datasets, there was not enough room for major improvements. An aspect which will be definitely improved vastly on the second iteration of the models, when a proper set of data will be provided.

3.3.5.1 Evaluation of methods

In the following tables (Table 1, Table 2), the basic evaluation metrics for the implemented models are displayed. The tables contain the values of precision, recall, and F1-score for both the anomaly detection (Table 1) and the norm detection (Table 2). Following the tables, we will draw some preliminary conclusions from the initial evaluation and a comparison between the methods used.

Table 1 Evaluation of models for non-anomalous traffic conditions

Model	Precision	Recall	f1-score
Isolation Forest	0.99	0.73	0.85
OneClassSVM	0.95	0.97	0.96
LOF	0.93	0.99	0.96
LSTM	0.97	0.98	0.97
EnsembleModel	0.96	0.96	0.96

Table 2 Evaluation of models for anomalous traffic conditions

Model	Precision	Recall	f1-score
Isolation Forest	0.22	0.93	0.35
OneClassSVM	0.64	0.43	0.49
LOF	0.5	0.1	0.16
LSTM	0.15	0.2	0.17
EnsembleModel	0.63	0.6	0.55

3.3.5.2 Conclusions

The classification reports for the Advanced Anomaly Detection framework within the CONDUCTOR project present a solid view of the performance of various models. For non-anomalous instances, the models show a strong ability to correctly identify normal traffic conditions, with LSTM and Ensemble Models demonstrating particularly high precision and recall, resulting in F1-scores close to 0.97 and 0.96, respectively. This indicates a robust performance in distinguishing the regular traffic flow, which is crucial for maintaining a baseline standard in traffic management systems.

In contrast, the detection of anomalous instances, which is arguably a more critical aspect of traffic management, exhibits varied performance across models. The Isolation Forest, while having a high recall of 0.93, falls short in precision, which is evident from its lower F1-score of 0.35. This suggests that while the model is adept at identifying a high rate of actual anomalies (high recall), it also misclassifies a considerable number of normal instances as anomalies (low precision).

The OneClassSVM and Ensemble Model show a more balanced performance with F1-scores of 0.49 and 0.55, respectively, indicating a reasonable trade-off between precision and recall. The LOF and LSTM models, however, underperform for anomaly detection, with F1-scores of 0.16 and 0.17, respectively, suggesting that these models may not be as effective in the current configuration for the specific task of detecting anomalies within the project's context.

It is essential to note that the performance of anomaly detection models is often more challenging to optimize due to the rarity and variability of anomalies. The Ensemble Model, which combines the strengths of various models, shows promise with the highest F1-score among anomaly detection methods, demonstrating the potential benefits of a hybrid approach.

To further enhance the detection of anomalies, it may be beneficial to consider additional tuning of model parameters, incorporation of more diverse training data, or exploration of novel modelling techniques tailored to the unique characteristics of traffic data. Moreover, continuous iteration and validation using real-world data are imperative to refine these models, as reflected in the AGILE approach adopted by the CONDUCTOR project.

In conclusion, the models' ability to accurately identify non-anomalies lays a strong foundation for effective traffic management. However, the critical task of anomaly detection necessitates further refinement to improve precision without sacrificing recall. The ongoing development and iteration of these models are vital to achieving the CONDUCTOR project's goal of revolutionizing transportation through state-of-the-art traffic and fleet management solutions.

4 ANOMALY DETECTION IN TRANSPORT DEMAND

A proper characterisation of the demand mobility patterns allows the detection of anomalies, the identification of possible factors leading to them (weather conditions, especial events, etc.), and the analysis of their impact on the transport network (supply). Based on this analysis, demand anomalies can be classified, and response plans can be defined. Moreover, as some of the causes are known in advance (e.g., weather conditions, planned events such as football matches or demonstrations, etc.), for some of these classes the anomalies can be foreseen and mitigated.

With this objective in mind, Nommon is developing a demand anomaly detection algorithm. This algorithm first forecasts the expected demand for a day based on the historical demand, given by the OD matrices that Nommon generates from mobile network data (MND) using the Nommon Mobility Insights solution[2], and then compares it with the observed demand of the day, given by the OD matrix of the day computed also using Nommon Mobility Insights solution. From this comparison it detects whether there is an anomaly or not. This comparison is performed based on confidence intervals OBJ OBJ OBJ.

The objective of this development is twofold. First, we want to develop a time series model able to accurately capture and forecast the mobility demand patterns. Besides, it should be able to detect anomalies in the demand and provide some probable explanation of them (based on calendar events).

The expected demand can be used for the strategic planning of the transport network. While the anomalies can be analysed and categorised based on their possible causes and impacts, allowing anticipation and more efficient decision-making.

4.1 Schema of the solution

The diagram on Figure 12 shows the workflow of the demand anomaly detection algorithm to be developed by Nommon. All modules and the relations between them are described below:

1. The demand prediction algorithm predicts the demand (as OD matrices for a given zoning system) of the next day based on multi-dimensional time series analysis over historical demand information (OD matrices).
2. The demand anomaly detection algorithm identifies whether there is an anomaly in the observed demand on the day by comparing it with the predicted one. We define as potential anomalies all observed demand values that lie outside a certain confidence interval around the expected value.
3. The general private mobility matrix segmented by mode is generated for the day with the anomaly in the demand using Nommon Mobility Insights solution.
4. The detected anomaly and its impact in the demand of each mode is assessed.

[2] Nommon Mobility Insightssolution obtains travel demand information from anonymised mobile network data, generating OD flows for the sampled mobile phone users and expanding these flows to the total population using census data, based on the user's residence location, age, and gender.

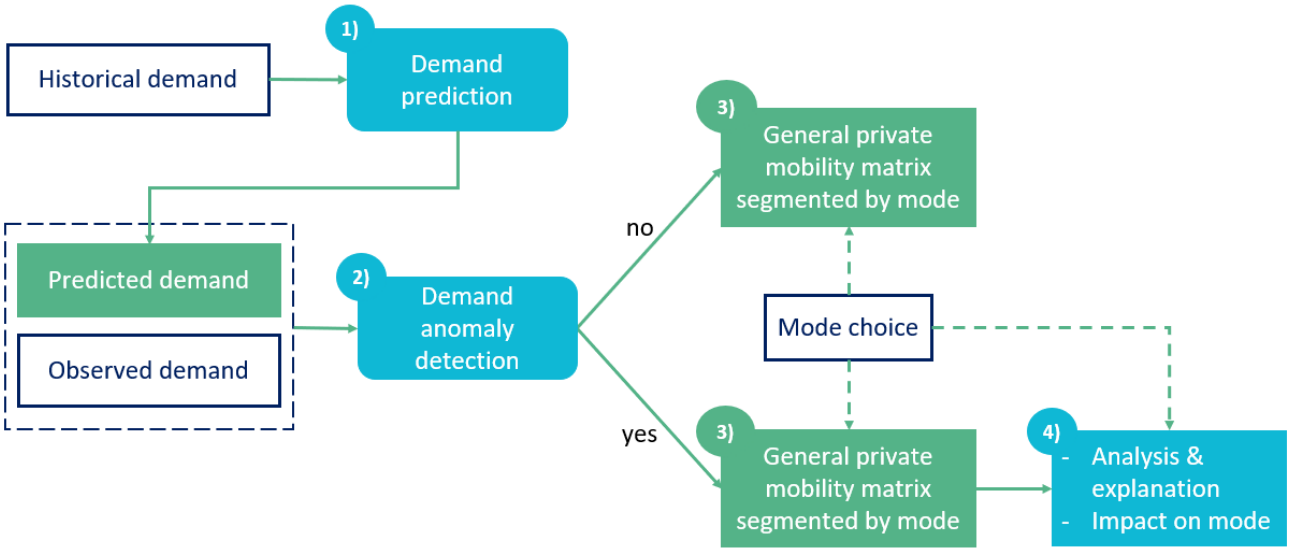

Figure 12 Demand anomaly detection algorithm workflow.

4.2 Data used

The main data source for this development are the OD matrices generated by Nommon. These matrices are generated from MND using the Nommon Mobility Insights solution.

The matrices are segmented by trips and travellers' characteristics. The travellers' characterization includes age, gender, residence place, income, etc. The trips' characterisation includes purpose, time and type of mobility, distance, passenger mobility and professional drivers & delivery. For this last characterisation, Nommon is also developing and algorithm for the identification and characterisation of delivery trips and estimation of delivery demand (see Section 4.2.1 of Deliverable D3.1 for more details on this development).

The data needed to generate the OD matrices are (see Deliverable D1.2 for more details on the data sources):

- Activity and travel diaries generated from MND using Nommon proprietary algorithms.
- Spanish census data, provided by the Spanish National Statistics Institute.
- Land use information, provided by the Spanish National Geographic Information Centre (CNIG).
- Transport supply data, provided by the Statistics Institute of the Community of Madrid and the Madrid Regional Transport Consortium.
- Travel surveys, provided by the Regional Transport Authority of Madrid.

4.3 Methodology

The methodology prosed is based on time series forecasting methods. This way, we consider the historical demand OD matrices as a multi-dimensional time series.

4.3.1 Analysis of historical data: time series

Time series are ordered sequences of values of a random variable documented in constant time intervals.

Time series can be broken down into three components: trend, seasonal pattern or seasonality, and noise. The seasonal pattern appears when the time series is affected by seasonal factors, which occur in a fixed and known period (a day of the week, a month, a season, ...), causing a regular pattern of changes. While the trend reflects the progression of the long-term time series, that is, the increasing or decreasing direction of the data. Finally, the noise component reflects the random and irregular influence of the data. FigureFigure 133 shows an example of a time series decomposition, where "data" corresponds to the original series, "seasonal" corresponds to the seasonal component, "trend" corresponds to the trend component, and "remainder", to the noise component.

Some time series forecasting algorithms perform a decomposition of the time series in their components and work with them independently, others consider the time series as a whole.

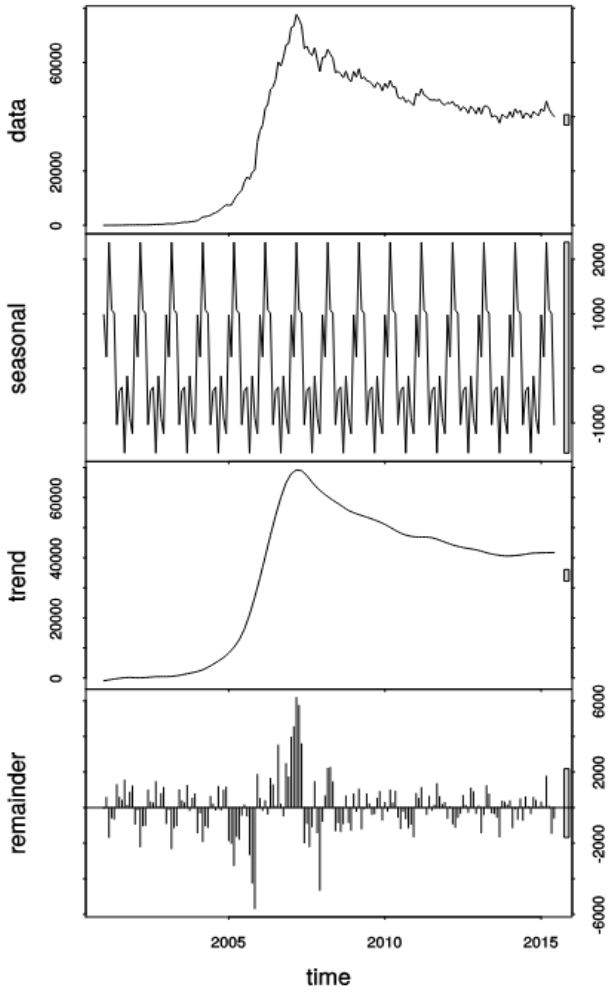

Figure 13 Example of a time series decomposition.

4.3.2 Time series forecasting models

The machine learning algorithm used to generate the forecasting models are the long short-term memory (LSTM) networks. Given the data size and according to the literature review, this algorithm is considered to be very appropriate for this task.

LSTMs are a class (or extension) of recurrent neural networks able to capture nonlinear patterns in time series data, while considering the inherent characteristics of non-stationary time series data.

Neural networks are a type of deep learning algorithm capable of learning complex patterns and non-linear data trends, without the need to make input assumptions. A neural network essentially consists of a sequence of layers, each of them having certain nodes called neurons. The layers are divided into three groups:

- An input layer, the nodes of which represent the input variables.
- A variable number of hidden layers, used to process and transform the input data for the generation of more complex variables that allow the model to understand and analyse the data.
- An output layer that represents the solution to the problem.

Based on the information flow, neural networks can be divided into two main categories: feed-forward neural networks, which only consider information flow in one direction (usually from the previous layer); and recurrent neural networks, which consider the flow of information from both the previous and the next layer.

4.3.2.1 Development and validation of the models

Neural networks have a series of internal parameters, called weights, that are iteratively adjusted during training to minimise a loss function using iterative optimization algorithms such as gradient descent (which finds local minima of differentiable functions by iteratively taking points along the direction in which the first derivative of the function decreases). The objective of the training phase is precisely to adjust the value of these internal parameters so that the predictive error (measured with the loss function) is minimal. When starting the training, these parameters are randomly initialised, and their values are refined as the training progresses.

For the first iteration of the algorithm, the model will be trained using the mean square error as the loss function and the optimization method Adam (adaptive moment estimation), which is a faster and more efficient extension of the stochastic gradient descent.

Following standard machine learning practices, the data will be split into three datasets, each of which assists in a different task of the model implementation process:

- Training set: this set is used to train the model.
- Validation set: this set is used to select the most appropriate combination of hyperparameters for the model (this is explained in Section 4.3.2.2).
- Test set: once the model is trained, this set is used to assess its predictive performance on new data (i.e., its ability to generalise).

Recall that the model predicts the demand of one day based on the observed demand of the previous days. This predictive performance is assessed using the square root mean square error (RMSE) and the mean absolute percentage error (MAPE) metrics, defined as follows:

$$\text{RMSE} = \sqrt{\sum_{t=1}^{n}(y_t - \hat{y}_t)^2 / n}\,,$$
$$\text{MAPE} = 1/n \sum_{t=1}^{n} |(y_t - \hat{y}_t)/y_t|\,,$$

where n is the number of observations in the set, y_t is the actual value of the series at time t, and $\hat{y}_t$ is the model prediction at time t.

The RMSE provides the mean number of predictive errors at each predicted instant. While the MAPE provides the average percentage that the predictive errors suppose with respect to the real value at each predicted instant. It is important to bear in mind that, given the same predictive error with respect to the RMSE metric, the MAPE metric can vary a lot depending on the real value of the

variable. For example, a prediction of 3 when the real value is 4 is not the same as a prediction of 49 when the real value is 50. In both cases, the RMSE is 1, however, in the first case the MAPE is ¼= 0.25 and in the second, 1/50=0.02, this means that the MAPE metric penalises more the predictive error of small values of the variable in question. Therefore, both metrics provide complementary information on the predictive error, giving both absolute and relative information with respect to the real value, and it is important to take both values into account when interpreting the results.

4.3.2.2 Hyperparameter tuning

The objective of hyperparameter tuning is to find the most suitable combination of hyperparameters that allows the model to achieve a good predictive performance. In order to perform the hyperparameter tuning, a grid search is implemented. A grid search consists in defining a set of values for each hyperparameter that needs to be specified, training the model for each of the possible combinations of values (or for a subset of them), and analysing the results for each combination.

Once a first evaluation of the data is performed, a set of values to implement the grid search will be fixed.

In order to ensure good predictive performance, the combination of hyperparameters selected should produce stable results and be non-dependent of the random initialisation of the model weights. To measure this, cross-validation is used to train the model for each hyperparameter combination. This technique, which essentially consists in training the model using different random splits of the training set (called folds) and evaluating the predictive performance on different validation sets, is adapted to time series in order to exploit its benefits. The adaptation respects the time-dependence of the samples by taking all the samples for both the training and the validation folds to be consecutive. This procedure can be performed in two ways:

- Split cross-validation: this method iteratively extends the training fold with the ones used for validation (of fixed size), training the model with the extended fold each time (see Figure 14).

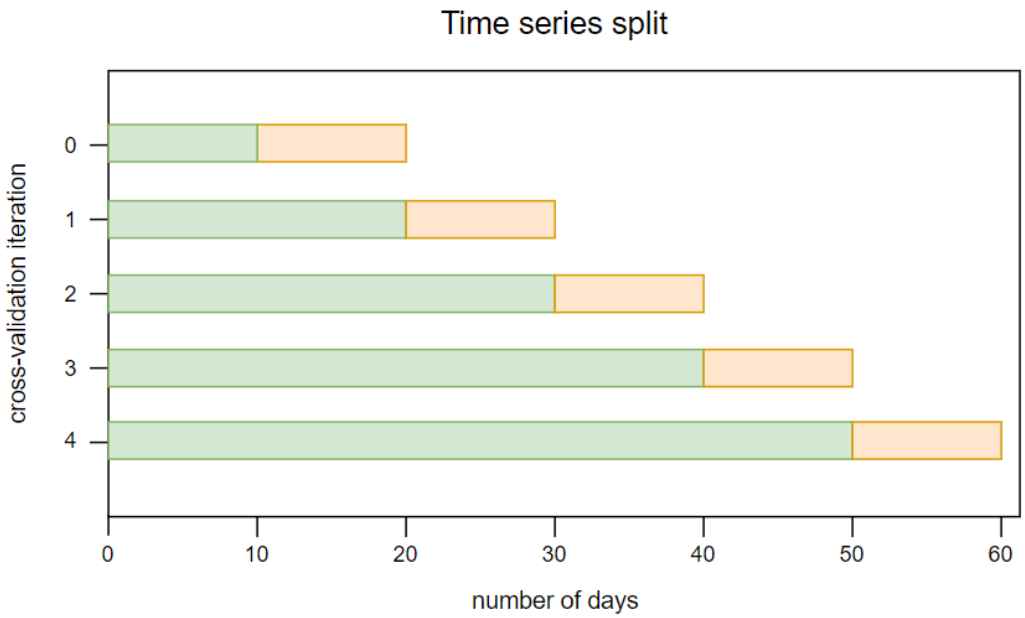

Figure 14 Split cross-validation for time series. In blue, the folds used for training and, in red, the folds used for validation.

- Blocked cross-validation: this method takes disjoint training and validation folds of fixed size to train the model (see Figure 15).

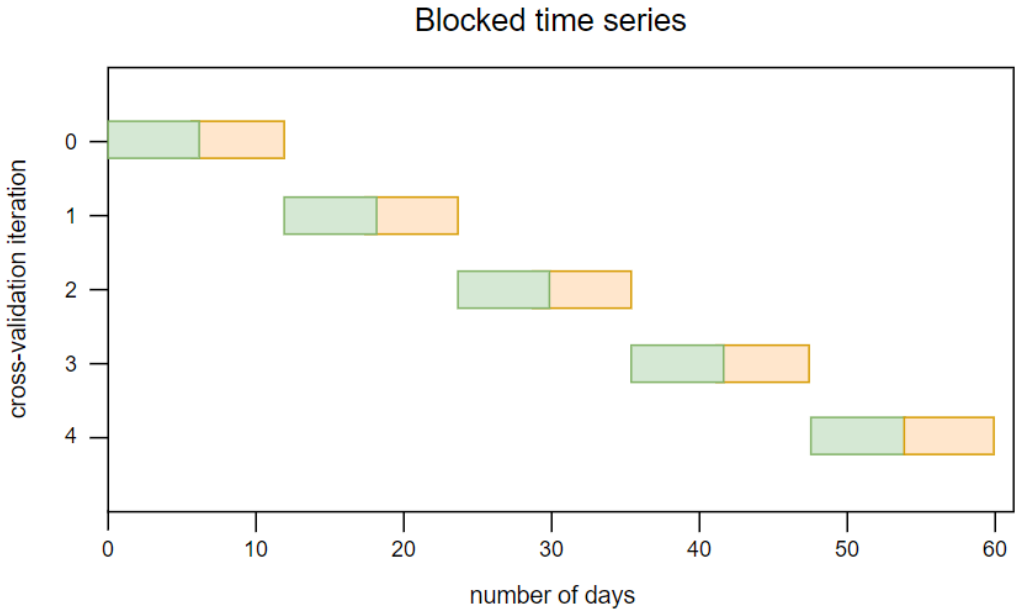

Figure 15 Blocked cross-validation for time series. In blue, the folds used for training and, in red, the folds used for validation.

As the final model should be retrained periodically, the idea is to use as little data as possible to generate the time series and train the models. For this reason, we use the split cross-validation method. With this method, we can find a trade-off between the model that best fits the data and the number of samples needed to train it.

Once the grid search is implemented, the results for each hyperparameters combination are analysed, for both the training folds and validation folds, looking for a trade-off between the predictive error on the training set and the validation set. With that, we ensure that the model fits well to the training set and generalises well.

Finally, the model is retrained with the chosen combination of hyperparameters on the full training set and used to predict the flow in the time interval corresponding to the test set, with the aim of analysing the decay of the prediction's accuracy along the days.

4.3.3 Confidence interval computation

Bollinger bands are used to define the confidence interval for each time series. Bollinger bands consist of an n-period moving average, an upper bound (band) at k times an n-period standard deviation above the moving average, and a lower bound (band) at k times an n-period standard deviation below the moving average.

Two types of confidence intervals will be computed, to distinguish among three kinds of values:

- **normal values**: values that lie within 1.5 standard deviation above and below the moving average,
- **outlier values**: values that lie between 1.5 and 3 standard deviations above and below the moving average, and

- **extreme values**: values that lie outside 3 standard deviations above or below the moving average.

Based on these intervals, anomalies will be detected and characterised.

4.3.4 Analysis of the anomalies

Finally, once the anomalies are detected, the specific OD pairs presenting the anomalies are identified and the calendar events (festivities, holidays, etc.), weather conditions, and planned events of the set of origins and destinations are analysed to look for possible explanations.

Additionally, the impact on the demand of each mode is analysed, to plan for needed actions in the supply to adjust to the demand change. To illustrate this, let us suppose that the demand one day is 2/3 higher than expected, and that this demand is concentrated in two modes, let us say private car and public transport. By analysing the increase in the demand in both modes, we can study the overload that these modes may experience and the impact on the performance of the service.

This information combined can potentially be used to anticipate these anomalies and mitigate their effects, being really useful in the decision-making process.

5 CONCLUSIONS

5.1 Summary

Deliverable 3.3 documents the work that is being conducted primarily by Frontier with the substantial contribution of Nommon as part of Task 3.5 (Anomaly detection), which aims to develop advanced situation detection capabilities to enable the identification of traffic patterns and emerging critical network and traffic anomalies that require specific remedial actions and response plans. Typical cases of such situations include emerging congestion and accidents posing the need for adaptation of the network, redirection and coordination of traffic.

This deliverable is a key document in developing anomaly detection routines and it begins with a detailed literature review. This review focuses on the two main types of anomalies addressed by CONDUCTOR: anomalies in traffic patterns and anomalies in transport demand. The review not only summarizes current knowledge but also identifies future opportunities and challenges, such as improving the adaptability and scalability of models and effectively using real-time data in anomaly detection.

After the literature review, the deliverable outlines the methodologies developed for Anomaly Detection in traffic patterns and transport demand. This section represents the first step of the project, where methods for detecting and managing these anomalies are established and explained. For traffic pattern anomalies, the document describes the early stages of development of the detection component, including the technical procedures and initial findings. These include evaluating various machine learning and deep learning algorithms, with a subset already partially implemented. These initial results help demonstrate the assessment and comparison of different models, which will later be applied in the Athens use case. In contrast, for transport demand anomalies, the document mainly focuses on setting up the methodology, laying the groundwork for further development.

Equally important is the fact that through our research work in the scope of this deliverable, we addressed some research gaps which were identified from the conducted literature review, whilst also tried contributing to their fulfilment by:

- Using various techniques to address the imbalance of data, such as ensemble of different oversampling of the minority and under sampling of the majority class;
- Using state of the art machine and deep learning models with the underlying help of classic yet refined statistical approaches.

5.2 Limitations

In the realm of traffic network anomaly detection, several limitations are evident. One primary challenge is the differentiation between normal traffic variations and genuine anomalies. This can lead to a high rate of false positives, where normal conditions are incorrectly flagged as anomalies, a phenomenon that was evident also in the initial implementation of the anomaly detections models from FI. Additionally, the quality and granularity of traffic data significantly impact the accuracy of anomaly detection. Incomplete or noisy data can skew results, leading to misinterpretations of traffic conditions. Moreover, the dynamic nature of traffic patterns, influenced by factors like weather, events, and unexpected incidents, adds complexity to the task of reliable anomaly detection. This necessitates sophisticated algorithms capable of adapting to varied and evolving traffic scenarios.

5.3 Future Work

As future steps, regarding the traffic anomalies detection algorithm, the focus should shift towards enhancing the accuracy and adaptability of detection algorithms. This involves integrating more diverse data sources, such as real-time weather information, public transport and even floating car data to better understand the factors influencing traffic patterns. Additionally, employing advanced machine learning techniques, including deep learning, could provide a broader understanding of complex traffic scenarios.

Efforts should also be made to improve the real-time processing capabilities of these systems, enabling quicker and more effective responses to identified anomalies. One more important aspect that should be addressed is the forecasting aspect of the anomalies, that could effectively predict an anomaly before it even happens. Furthermore, expanding the scope of research to include a wider variety of urban settings would provide valuable insights into the scalability and versatility of these detection systems.

Finally, the transport demand anomaly detection algorithm will be implemented and iteratively refined.

REFERENCES

[1] B. Hellinga and G. Knapp, “Detection, Automatic Vehicle Identification Technology-Based Freeway Incident,” *Transportation Research Record Journal of the Transportation Research Board,* 2000.

[2] L. Wei and D. Hong-ying, “Real-time Road Congestion Detection Based on Image Texture Analysis,” *Procedia Engineering,* vol. 137, pp. 196-201, 2016.

[3] Amin-Naseri, A. Sharma and P. Chakraborty, “Evaluating the Reliability, Coverage, and Added Value of Crowdsourced Traffic Incident Reports from Waze,” *Transportation Research Record Journal,* 2018.

[4] M. Levin and K. Gerianne, “Incident Detection: A Bayesian,” *Transportation Research Record,* vol. 682, 1978.

[5] H. Payne, He, E. fenbein and H. C. Knobel, “Development and testing of incident detection algorithms. Vol. 2, research methodology and detailed results.,” United States. Federal Highway Administration. Office of Research and Development, 1976.

[6] Gall and Hall, “Distinguishing Between Incident Cpngestion and Recurrent Congestion: A proposed logic,” *Trasnportation Research Record,* 1989.

[7] B. N. Persaud and F. L. Hall, “Catastrophe theory and patterns in 30-second freeway traffic data - implications for incident detection.,” *. Transportation Research,* no. 23, 1989.

[8] D. Sun, C. Zhang, M. Zhao, L. Zheng and W. Liu, “Traffic congestion pattern detection using an improved McMaster algorithm,” in *29th Chinese Control And Decision Conference (CCDC)*, 2017.

[9] Y.-S. Jeong, M. Castro-Neto, M. K. Jeong and L. D. Han, “A wavelet-based freeway incident detection algorithm with adapting threshold,” *Transportation Research Part C: Emerging Technologies,* 2011.

[10] B. Anbaroglu, B. Heydecker and C. , “Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks.,” *Transportation Research : Emerging Technologies,* no. C, 2014.

[11] C. Piciarelli and G. L. Foresti, “On-line trajectory clustering for anomalous events detection. Pattern Recognition Letters,” *Vision for Crime Detection and Prevention,* 2006.

[12] F. Porikli and X. Li, “Traffic congestion estimation using HMM models without vehicle tracking.,” *IEEE Intelligent Vehicles Symposium,* 2004.

[13] E. Kidando, R. Moses, T. Sando and E. E. Ozguven, “Evaluating recurring traffic congestion using change point regression and random,” *Transportation Research Record,* 2018.

[14] C. L. Dudek, C. J. Messer and N, “Incident detection on urban freeways.,” *. Transportation Research Record 495,* 1974.

[15] K. Balke, C. L. Dudek and Mountain, “Using probe-measured travel times to detect major freeway incidents in Houston, Texas.,” *Transportation Research Record 1554,* 1996.

[16] M. Snelder, T. Bakri and B. van Arem, “Delays caused by incidents: Data-driven approach.,” *Transportation Research Record 2333,* 2013.

[17] Y. Chung, “Quantification of nonrecurrent congestion delay caused by freeway accidents and analysis of causal factors.,” *2229,* 2011.

[18] P. Chakraborty, C. Hegde and A. Sharma, “Data-driven parallelizable traffic incident detection using spatio-temporally denoised robust thresholds.,” *Transportation Research Part C: Emerging Technologies 105,* 2019.

[19] Z. Yuan, X. Zhou and T. Yang, “Hetero-ConvLSTM: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal,” in *D International Conference on Knowledge Discovery & Data Mining.*, New York, 2018.

[20] K. Pasini, “Forecast and anomaly detection on time series with dynamic context: Application to the mining of transit ridership data,” 2021.

[21] A. Soule, K. Salamatian, A. Nucci and N. Taft, “Traffic matrix tracking using Kalman filters,” *ACM SIGMETRICS Performance Evaluation Review,* vol. 33, 2005.

[22] N. Davis, G. Raina and K. Jagannathan, “A framework for end-to-end deep learning-based anomaly detection in transportation networks.,” *Transportation research interdisciplinary perspectives,* vol. 5, 2020.

[23] I. Markou, F. Rodrigues and F. C. & Pereira, “Demand pattern analysis of taxi trip data for anomalies detection and explanation,” *96th Annual Meeting of the Transportation Research Board. Washington, DC: Transportation Research Board of the National Academies,* 2017.

[24] I. Markou, F. Rodrigues and F. C. & Pereira, “Use of taxi-trip data in analysis of demand patterns for detection and explanation of anomalies,” *Transportation Research Record,* pp. 129-138, 2017.

[25] W. Yu, H. Bai, J. Chen and X. Yan, “Anomaly detection of passenger OD on Nanjing metro based on smart card big data,” *IEEE Access,* vol. 7, 2019.

[26] J. Liu, F. Zheng, H. van Zuylen, J. Li and J. Luo, “An anomaly detection-based dynamic OD prediction framework for urban networks,” *Forum on Integrated and Sustainable Transportation Systems (FISTS),* pp. 133-141, 2020.

[27] X. Qian, S. V. Ukkusuri, C. Yang and F. Yan, “Short-term demand forecasting for on-demand mobility service,” *IEEE Transactions on Intelligent Transportation Systems,* vol. 23, no. 2, pp. 1019-1029, 2020.

[28] F. Zheng, J. Liu, H. van Zuylen, K. Wang, X. Liu and J. Li, “Dynamic OD prediction for urban networks based on automatic number plate recognition data: Paramertic vs. non-parametric approaches,” *IEEE intelligent transportation systems conference (ITSC),* 2019.

[29] M. He, U. Muaz, H. Jiang, Z. Lei, X. Chen, S. V. Ukkusuri and S. Sobolevsky, “Ridership prediction and anomaly detection in transportation hubs: an application to New York City,” *The European Physical Journal Special Topics,* vol. 231, no. 9, pp. 1655-1671, 2022.

[30] A. Lakhina, M. Crovella and C. Diot, “Diagnosing network-wide traffic anomalies,” *ACM SIGCOMM computer communication review,* vol. 34, no. 4, pp. 219-230, 2004.

[31] V. Kumar, V. Cahnola and A. Banerjee, “Anomaly Detection: A Survey,” *ACM Computing Surveys,* 2009.

[32] S. Chen, W. Wang and H. v. Zuylen, “Construct support vector machine ensemble to detect traffic incident,” *Expert Systems with Applications,* 2009.

[33] P. Mercader and J. Haddad, “Automatic incident detection on freeways based on Bluetooth traffic monitoring,” *Accident Analysis & Prevention,* 2020.

[34] H. Yang, Y. Wang, H. Zhao, J. Zhu and D. Wang, “Real-time Traffic Incident Detection Using an Autoencoder Model,” IEEE, 2020.

[35] L. Li, Y. Lin, B. Du, F. Yang and B. Ran, “Real-time traffic incident detection based on a hybrid deep learning model,” *Transportmetrica A Transport Science,* 2020.

[36] B. N. Persaud and F. L. Hall, “Catastrophe theory and patterns in 30-second freeway traffic data— Implications for incident detection,” *Transportation Research Part A: General,* vol. 23, no. 2, 1989.

A. ABBREVIATIONS AND DEFINITIONS

Term	Definition
EVT	Extreme Value Theory
FPR	False Positive Rate
GCRF	Gaussian Conditional Random Field
GMM	Gaussian Mixture Models
ILD	Inductive Loop Detectors
K-NN	K-Nearest Neighbor
LOF	Local Outlier Factor
LSTM	Long Short-Term Memory
MAPE	Mean Absolute Percentage Error
MND	Mobile Network Data
OD	Origin-Destination
PCA	Principal Component Analysis
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SMOTE	Synthetic Minority Oversampling Technique
SVM	Support Vector Machine
TPR	True Positive Rate