Using Machine Learning to Improve Interactive Visualizations for Large Collected Traffic Detector Data

Rifat Mehreen Amin
rifat.amin@ifi.lmu.de
LMU Munich
Munich, Bavaria, Germany

Pia Hammer
pia.hammer@campus.lmu.de
LMU Munich
Munich, Bavaria, Germany

Andreas Butz
andreas.butz@ifi.lmu.de
LMU Munich
Munich, Bavaria, Germany

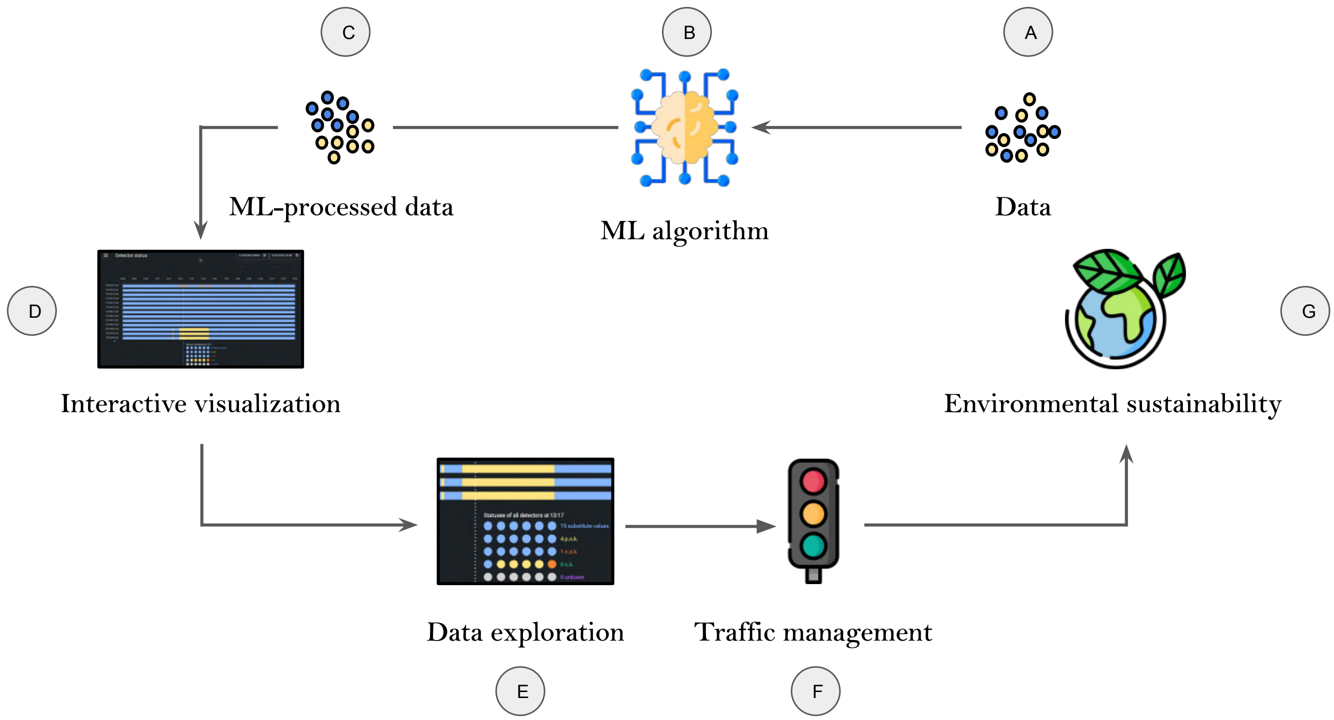


Figure 1: Marina is an expert in the traffic domain. She uses our interactive visualization tool, which processes raw traffic data (at A) using machine learning (ML) algorithms (at B) to find hidden patterns and anomalies (at C). The ML-processed data is then seamlessly visualized through the tool (at D). With her expertise, Marina explores the data and finds some partially okay (p.o.k) and not okay (n.o.k) traffic detectors (at E). She contacts the service technicians from the city, presents her findings, and asks them to check the corresponding detectors on the roads (at F). Marina also gets in touch with the engineering team and tries to pinpoint why certain detectors start behaving unexpectedly. This enables them to make informed decisions that contribute to efficient traffic management, leading to a more sustainable traffic system, aligning with broader environmental sustainability goals (at G).

ABSTRACT

In traffic engineering, cities rely on large detector datasets to manage traffic. Visualizing these big, multi-dimensional datasets poses challenges such as overplotting and dimension reduction, often rendering traditional visualization techniques inadequate. To address this, we added two machine learning (ML) algorithms (Local Outlier Factor algorithm and K-Prototypes clustering) to an interactive time series visualization to improve exploration by both domain experts and non-experts. We used an original detector dataset of a mid-sized German city. Our findings reveal that the ML algorithms greatly enhanced data exploration in these interactive visualizations, particularly for users with limited domain knowledge. This research directly contributes to the design of traffic data analysis tools, offering a foundation for traffic detection hardware and

IUI '24, March 18–21, 2024, Greenville, SC, USA

ACM ISBN 979-8-4007-0508-3/24/03
https://doi.org/10.1145/3640543.3645177

software improvements but also advancing complex dataset visualization in general. It will ultimately lead to more informed decisions, improved traffic management, and has the potential to reduce air pollutants, thus counteracting climate change.

CCS CONCEPTS

• **Human-centered computing → Information visualization**; • **Computing methodologies → Anomaly detection**; **Cluster analysis**.

KEYWORDS

Information Visualization, Traffic Detector Data, Clustering, Anomaly Detection

ACM Reference Format:
Rifat Mehreen Amin, Pia Hammer, and Andreas Butz. 2024. Using Machine Learning to Improve Interactive Visualizations for Large Collected Traffic Detector Data. In *29th International Conference on Intelligent User Interfaces (IUI '24), March 18–21, 2024, Greenville, SC, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3640543.3645177

1 INTRODUCTION

The traffic domain generates a vast amount of data crucial for traffic management and environmental decisions [11]. This research addresses the analysis of a high-dimensional traffic dataset from a mid-sized German city, provided by a German company, comprising input from over 500 detectors from different detector groups. We use the data from one detector group with 37 detectors. Ensuring the quality of the detector data and detecting anomalies is vital because inaccuracies can disrupt data analysis and traffic management. The specific problem addressed in this research is the need to effectively explore and identify anomalies in a complex, multidimensional traffic detector dataset. Anomalies, or outliers, represent deviations from expected data patterns [5, 16] and are challenging to define precisely across diverse fields [18]. Detecting anomalies is crucial for data accuracy [5], and error removal [39], as these anomalies can take the form of both global and local outliers [18], potentially indicating systematic biases in sensors [39].

There are ML algorithms that help detect the mentioned anomalies. Previous research in this field has primarily focused on supervised ML approaches [19], which require extensive labeled data. However, according to Chegini et al. [11], the unavailability of labeled data in real-world datasets often poses a significant challenge. To avoid this issue, analysts resort to deriving initial labels from the data itself by leveraging the data's inherent characteristics or drawing upon specialized domain knowledge [11]. In contrast, unsupervised ML techniques offer an alternative approach that does not require labeled training data. Clustering algorithms exemplify unsupervised ML techniques as they effectively group similar records within a dataset, thereby revealing latent structures that may remain concealed during manual exploration [11].

Identifying anomalies is the basis of traffic management requirements such as traffic congestion monitoring, hot spot analysis, and incident detection. Data visualization plays a vital role in unveiling these patterns and anomalies within big datasets [12], fostering questions and insights [17]. In real-world datasets, latent structures such as data anomalies or clusters often go unnoticed in visualizations like scatterplots and parallel coordinates [15]. Visual analytics (VA), encompassing data analysis and human visual pattern recognition [12], aids in these big data exploration, understanding, and decision-making [13, 23]. However, large-scale multidimensional data presents challenges in visualization due to human cognitive limitations [40], necessitating innovative techniques. Moreover, visualizing large-scale data with numerous features proves challenging on limited screen space [26].

In response to these problems, we have developed an interactive application that enables users (both domain experts and non-experts) to explore traffic detector data using time series visualizations, the Local Outlier Factor (LOF) algorithm [7], and K-Prototypes clustering [20].

What sets our work apart is the customization of visual encoding and interaction methods, specifically tailored to address the unique challenges posed by traffic detector data. We found that ML techniques significantly improve data exploration, especially for users with limited domain knowledge. The clustering of measured data provides distinct user groups with better insights into the data. This implies that our approach can enhance data quality exploration and contribute to more effective traffic management strategies.

Our work seeks to bridge the gap between ML and data visualization, ultimately enhancing anomaly detection and data quality assessment in traffic management by asking the following research questions:

RQ1 *How can ML be used to build insightful visualizations that allow users to explore traffic detector data and spot anomalies?*
RQ2 *How can data visualizations aided by ML assist users with varying levels of domain knowledge in discovering anomalies effectively?*

2 BACKGROUND AND RELATED WORK

Proper traffic management can minimize the number of road accidents, air pollutants, and energy consumption, besides other factors [14]. This section summarizes some recent research approaches in road traffic analytics and anomaly detection techniques.

2.1 Road Traffic Analytics

Different data sources influence the resulting data quality and the data's impact on traffic management. In addition to special detectors that are permanently installed on the road, data from the mobile network are also used. Some cities even offer public traffic data, leading to tools like "Traffic-Cascade" by Kwee et al. [24], detecting congestion cascades, which are clusters of spatiotemporal congested segments defined by slower speeds compared to the normal pattern using public bus data and GenClus [32]. The dashboard displays a list of these cascades, along with charts summarizing their timing and a map showing their spatial distribution. Expanding beyond public traffic data, Molina et al. [29] used IoT technology and AI for real-time traffic monitoring. Their system combined DeepSort and YOLOv5, displaying real-time traffic information, using color-coded heatmaps to show traffic volume on a dashboard. To ensure traffic data privacy, Costa et al. [14] used the k-anonymity algorithm on telecommunication data in their traffic analytics dashboard with heat maps. Still, traffic data quality issues remain that can be caused by sensor failures and network faults. Focusing on missing values and data anomalies of the Greek traffic Open Government Data, Karamanou et al. [22] employed flow-speed correlation and STL (Seasonal-Trend decomposition using LOESS) for anomaly detection, improving data with spatial information and concise visualizations with line charts and maps. Therefore, diverse data sources and advanced analytical techniques have significantly improved the understanding of traffic patterns and management, although challenges related to data quality and anomalies persist.

2.2 Anomaly Detection

Identifying anomalies is crucial for making informed decisions in data analysis [27, 38]. Despite its essential role in scientific domains, research on anomaly detection frequently leans towards ML methods, which may lack the interpretability necessary for effective exploration. In collaboration with NASA, Wright et al. [41] developed interpretable anomaly detection models tailored for extraterrestrial exploration. Furthermore, current anomaly detection systems focus on a subset of anomalies and require human experts for verification, making the process biased. The RAMAN framework by Ratadiya et al. [30] addressed multimodal anomaly detection within the Mars Science Laboratory power subsystem, demonstrating robustness across diverse anomaly types, input data, and domain constraints. In a similar vein, Meng et al. [27] introduced VADAF, a visual interface designed for abnormal client detection in a federated learning setting. Furthermore, Abello et al. [1] present ATLAS, a graph exploration framework dedicated to anomalous subgraph detection. In the context of social media, Schaffer et al. [33] delve into commuter traffic anomalies extracted from Twitter messages, employing automated anomaly detection methods. According to Ahmed et al. [2], these different types of anomaly detection techniques can fall into three major categories: statistical approach, classification, and clustering.

2.2.1 Statistical Approach. Statistical anomaly detection methods, as described by Ahmed et al. [2], offer simplicity and do not require labeled training data, making them capable of detecting unknown anomalies. However, they struggle with complex anomalies and may produce a high number of false positives [2]. In urban air quality data analysis, van Zoest et al. [39] introduced an outlier detection method based on confidence intervals. They visualized their findings using line charts, scatterplots, and boxplots. Similarly, Belhaouari et al. [5] utilized statistical approaches for outlier detection. They transformed multidimensional data into one-dimensional distances and employed unsupervised detection, visualizing their results with boxplots. Their method overcame the limitations of traditional boxplot outlier detection, particularly for noisy data and small anomaly clusters, through the application of probability density estimation using k-nearest neighbors (KNN) distance vectors. Turkay et al. [37] introduced an abstraction layer for high-dimensional datasets. They assigned representative factors to sub-groups of dimensions with similar statistical values, visualizing these relationships in scatterplots. Their iterative process involved comparing data structures, computing statistical values for dimensions, and using Principal Component Analysis (PCA) to create new features for sub-groups. This method proved effective in discriminating different medical conditions based on group factors.

2.2.2 Classification. According to Ahmed et al. [2], classification algorithms excel at identifying anomalies and generating alerts, but they cannot detect unknown anomalies. However, many ML algorithms require an extensive amount of labeled training data [2]. Chegini et al. [11] conducted research on interactive labeling of a multivariate football player dataset for supervised ML using linked visualizations, clustering, and active learning. Their focus was on seamlessly integrating interactive visualizations with traditional ML techniques to simplify labeling tasks while managing user workload. They proposed a visual analytics approach for exploratory data analysis and partitioning of multivariate datasets into meaningful labeled sections. Users could semi-supervise the labeling process with active learning, marking interesting patterns or outliers. Once the classifier learns from the training data, it can automatically partition similar datasets.

2.2.3 Clustering. ML-based approaches, like clustering, excel in detecting complex anomalies, yet their performance on unknown outliers may vary with hyperparameters [2]. Using K-Means and Gaussian Mixture Models, Riveiro et al. [31] visualized anomalies in road traffic data with circular layouts and heatmaps. The circular layout also has its drawbacks when many features exist. For football data analysis and labeling tasks, Chegini et al. [11] used K-Means and hierarchical clustering. Although methods like K-Means and hierarchical clustering are used for anomaly detection, choosing them depends on the situation and requires domain expertise [11]. Moving towards a fuzzy clustering, Fan et al. [16] proposed an interactive visual analytics approach for network anomaly detection using a fuzzy c-means-based algorithm. Basurto et al. [3] employed unsupervised visualization methods like Curvilinear Component Analysis and t-distributed stochastic neighbor embedding (t-SNE) for robot failure detection, visualized in 3D scatterplots. However, no single anomaly detection algorithm outperforms others across various time series datasets, with the challenge of parameter sensitivity [34]. Systematic testing with different parameter values, especially for k-nearest neighbor based methods in unsupervised outlier detection, is crucial [10]. Hence, while ML-based techniques show significant potential for detecting anomalies in diverse contexts, appropriate methods and parameter selection are essential for effective application.

Due to the lack of labeled data, our application uses ML algorithms that can work with raw data, such as the LOF algorithm. Clustering, especially for mixed-type data, is valuable for finding anomalies in traffic detector data [21]. Existing tools for visualizing traffic data often skip assessing data quality, and the complexity of visualizations can be a barrier for non-expert users. Our work aims to fill these gaps by evaluating real-life traffic detector data quality and providing user-friendly visualizations by testing the hypotheses below:

H1: *ML techniques make it possible to develop visualizations that facilitate the exploration of traffic detector data and enhance users' ability to detect anomalies effectively.*

H2: *ML integration has the potential to support users with varying levels of domain knowledge in improving their effectiveness in identifying anomalies in interactive visualizations.*

3 SYSTEM DESIGN

To test our hypotheses, we developed a graphical user interface for exploring the traffic detector data with and without ML integration. The traffic detector data visualization process is shown in Figure 2 and explained below.

3.1 Data Collection and Preprocessing

3.1.1 Data Collection. The company collected traffic detector data from 37 different Traffic Eye Universal 5 (TEU) detectors in December 2022 in a mid-sized city in Germany. These detectors employ overhead Passive Infrared Technology (PIR) and can be installed without additional cables on poles or bridges, seamlessly integrating into existing traffic data systems. They are powered by solar batteries capable of lasting four weeks without sunlight. TEUs can count vehicles, measure their speed and street occupancy, and classify them into five vehicle classes based on length. However, they cannot recognize cyclists and pedestrians. Data is transmitted to the traffic center through mobile networks (3G or 4G/LTE) via an integrated cellular modem. A terminal collects, preprocesses, and transmits data from connected detectors at set intervals, aiding traffic operations and environmental management strategies.

3.1.2 Data Preprocessing. The data was provided by the company's platform, downloaded, and stored locally for independence from cloud platform availability. Each day had a separate folder containing multiple JSON files. For each day, there were around 150 files, resulting in 4650 files for December 2022. These files consisted of concatenated key-value pairs with attributes describing the keys in the dataset. Each observation entry had five attributes, some of which contained nested attributes, arrays, or objects. Notably, the attribute containing the city's name was anonymized. In addition, some columns had mixed data types that required conversion to strings. As a result of performance considerations, HDF files[1] were found to load faster in the backend than JSON or CSV files, leading to the conversion of preprocessed data into HDF format.

The goal was to analyze and detect anomalies through clustering and the LOF algorithm and create visualizations. Consequently, all attributes in the traffic detector dataset were considered, as no labeled dataset was available to identify outliers or group similar observations.

3.2 Anomaly Detection and Clustering

The data was first examined using statistical parameters such as mean and standard deviation. For this purpose, the one month dataset was considered as a whole to get a first impression and orientation. Since the dataset did not contain any labels, classification algorithms were not feasible. Previous studies suggested dimensionality reduction with PCA [11, 28]. But visualizing the results of the evaluation for a subset of the traffic detector data with PCA showed that overplotting hides data points, and the dimensions were confusing to the user. In our case, K-Prototypes showed more promising results during the testing phase. K-Prototypes combines K-Modes and K-Means and can cluster mixed-type data, which involves selecting prototypes and assigning units to clusters based on the nearest prototype [21]. For anomaly detection, due to the lack of a labeled dataset for training, we used the LOF algorithm that performs well for multi-modal datasets [9]. LOF can effectively detect anomalies when having two datasets with different densities. The algorithm compares one sample's anomaly score with its neighbors [9]. Our application uses the LOF algorithm to highlight anomalistic metadata, specifically timestamp differences. To identify anomalies, conspicuous daily values within the month were extracted by experts. These values are the ones that should be extracted from K-Prototypes and the LOF.

3.3 Interactive Visualization

For building the interactive visualization tool, we conducted unstructured interviews with a traffic engineer, a product owner, and a developer to gather their initial impressions and address specific questions. These interviews aimed to provide insights into the necessity for anomaly detection and the creation of concise visualizations. Based on these insights, we created mockups, which were then shared with potential end-users for their feedback and integrated into the tool's development. Inspired by Grimmeisen and Theissler [19], our tool intentionally avoided general navigation features to minimize interference with users' decision-making. Given the vast dataset, we offered filtering and slicing options for users to specify data subsets based on attributes, ensuring fine-grained analysis. Users could easily navigate between different data views using an on-click side menu, promoting efficiency. Fan et al. [16] stated that users often switch between overview and detail views to judge abnormal conditions. Hence, the visualization's goal was to provide the user with situational awareness and the ability to detect anomalistic data at the same time. Therefore, Shneiderman's mantra [35], "Overview first, zoom and filter, then details-on-demand" was applied. To create effective and efficient visualizations, we prioritized showing complex data in a simple manner, taking into account the challenges of high-dimensional spaces and complex relationships, as noted by Behrisch et al. [4]. This approach helped users focus on primary visual patterns, enhancing efficiency in clustering and anomaly detection tasks. Since the traffic detector data lacked a pre-existing hierarchical structure and was time-based, our application included a time selector for all visualizations, enabling users to adjust the date range for their analysis.

Below, we explore different views used in the application. These views are essential for monitoring detector statuses, identifying data duplicates, examining data submission intervals, assessing timestamp differences, and exploring measured data patterns. We discuss each view's unique features and design considerations.

3.3.1 Detector Status View. In the first view (see Figure 3), users were able to monitor the statuses of 37 detectors over time. To prevent clutter, detectors were grouped, and users could toggle between groups. The system used colored bars to represent detector statuses, enabling users to track changes over time. Clicking on a bar revealed a legend at the bottom, displaying all detectors and their statuses at that specific time. When selecting a broader time range in the top time selector (as shown in Figure 4), the x-axis switched from

[1] https://www.hdfgroup.org/solutions/hdf5/, last accessed February 9, 2024

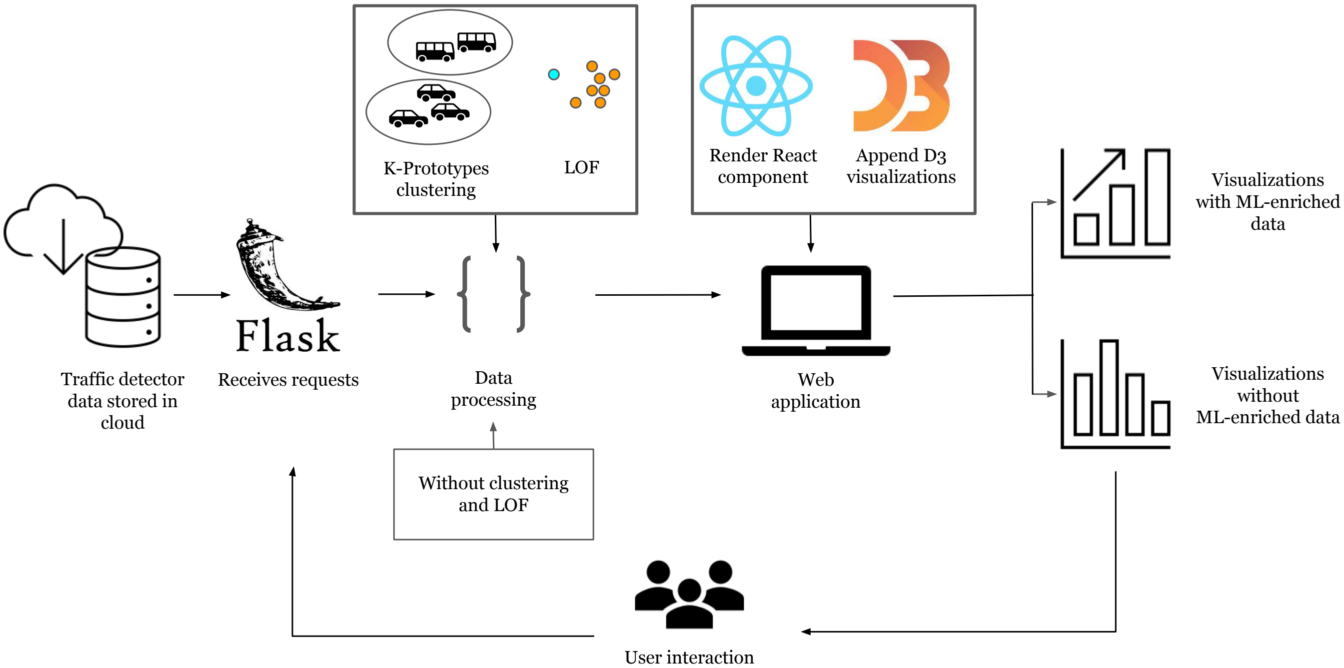
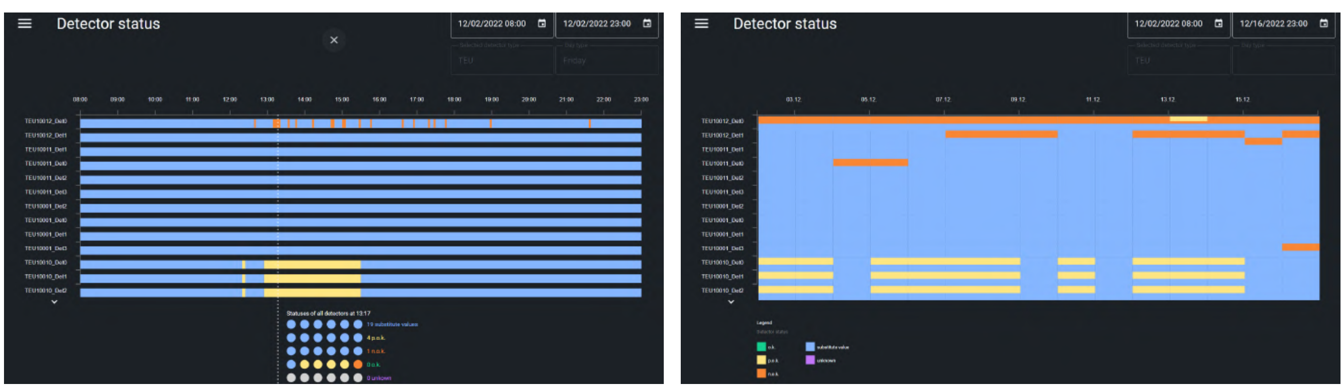


Figure 2: Traffic detector data visualization process. The traffic detector data was stored in the cloud. After that, Flask received requests. The data was then processed via machine learning (ML) algorithms. Here, K-Prototypes and Local Outlier Factor (LOF) algorithms. Afterward, the ML-processed data was sent to the front end (the web application). The application rendered the react component and appended D3 visualizations. Users could interact with the tool, exploring visualizations both with and without ML-enriched data.



(a) Single Day View **(b) Multiple Days View**

Figure 3: Detector Status View. Here, p.o.k. = partially okay, n.o.k. = not okay.

hourly to daily intervals. Multiple colors for a detector on a given day indicated varied statuses. We maintained a fixed color scale to align with our design system and user clarity. Our visualization accommodated multiple detectors on one page without overlapping or losing historical data, enhancing data presentation and user experience.

3.3.2 Duplicate Entries View. The duplicate entries view (see Figure 5) displayed data duplicates in the dataset. The x-axis, as in the detector status view, represented hours, and the y-axis listed detectors. Turquoise bars indicated duplicate data on certain days, while gray bars signified no duplicates. Empty rows meant no data submissions by the detector. Each row displayed the total observations within the chosen time range. In the single day view (See 5a), users could click turquoise bars to explore duplicate details, while in the multi day view (See 5b), bars showed only the days with duplicates, requiring users to adjust the time range for closer examination.

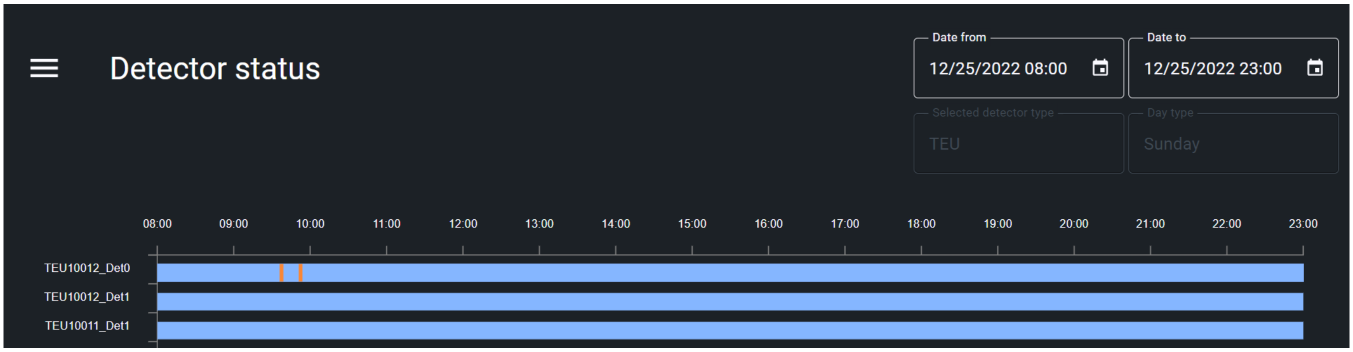


Figure 4: Application's header showing the time selector (top right).

(a) Single Day View

(b) Multiple Days View

(c) Submission Intervals View

Figure 5: Duplicates View (a) Single Day View, (b) Multiple Days View, and (c) Submission Intervals View.

3.3.3 Data Submission Intervals View. Detectors submitted data at varying intervals, indicated using a bubble chart (See 5c). Larger circles represented more observations in a specific interval. Clicking a circle updated sensor colors in the list, indicating data submission in that interval. Users could select up to six detectors in a line chart for a detailed comparison. The x-axis represented chosen detectors, and the y-axis displayed submission intervals (formatted as hh:mm). The dot size on the line chart indicated the number of entries for each interval. Users could add or remove detectors in the detail view, available only for one-day selections.

3.3.4 Timestamp Differences View. In this view, two timestamps were analyzed: the control unit timestamp and the publishing timestamp. In the single day view (see Figure 7), the x-axis represented time, and the y-axis listed detectors. Each detector row contained a small scatterplot illustrating time delays in seconds. Users could select up to three sensors on the y-axis to display them in a detailed line chart at the bottom. This line chart's x-axis showed time, while the y-axis showed timestamp differences in seconds. Red dots indicated anomalies detected by the LOF algorithm, sensitive to rapid increases in seconds. Users could deselect detectors in the detailed view (see 6b). Zooming in by narrowing the time range in the selector allowed for more detailed data exploration. For a multi-day perspective (see 7c), users could switch to a view displaying dates on the y-axis and a 24-hour time format on the x-axis. Bars in each detector row represented hourly data availability, with color intensity indicating mean timestamp differences. Darker hues signified larger mean differences, and users could hover over bars for detailed means. To delve deeper into specific dates, users could adjust the time selector and return to the single day view once they had gained an overview.

3.3.5 Measured Data View. This view (see Figure 8) complemented the previously described metadata visualizations. It displayed data for a single day and employed the K-Prototypes clustering algorithm to reveal data patterns. Users could toggle between four clusters. They could click on a cluster to zoom in and access detailed information, including small bar charts depicting attributes like *alarm*, *quality*, *measuring duration*, and *detector status*. The y-axis showed observation counts, and the x-axis displayed unique attribute values. Above these charts, users found cluster averages for *vehicle counts*, *occupancy*, and *speed*, which were updated when switching clusters. On the left side, a list of detectors in the cluster was provided, acknowledging that multiple clusters might include the same sensor. Users could click on a specific sensor to view its measured data in a line chart. The chart encoded data using glyph characteristics such as position, color, size, and stroke style. Dot height represented vehicle counts (y-axis), while the x-axis indicated measurement time. Stroke design indicated detector status, color represented measurement time, and circle fill reflected occupancy. Hovering over dots revealed detailed observation values. The measured data without clustering view (See 8b) lacked cluster differentiation and displayed all detectors with overall averages simultaneously.

3.4 Back End

The application's back end module was responsible for handling data processing and applying anomaly detection algorithms. It used Flask[2] to manage routing, entry, and endpoint (see Figure 2). The data was analyzed depending on the request triggered by the user's input. Flask handled incoming requests, with each request being associated with a dedicated function. Each routing function checked whether the time range selected in the front end exceeds 24 hours. Depending on this evaluation, distinct functions were executed when either a single day or multiple days were chosen on the front end. These functions were responsible for data analysis, organization, and the application of ML techniques.

3.5 Front End

After Flask processed incoming requests, it sent formatted data to the front end without relying on a database or caching, which posed challenges for handling large traffic detector data. To improve loading times, clusters were computed in the background with every data change in the visualization. The application's front end utilized JavaScript[3], React[4], and the Material UI[5] library for web interfaces. Data visualizations were created using D3.js[6], allowing users to manipulate and interact with data. Application styling was achieved through CSS (Cascading Style Sheets).

4 STUDY DESIGN AND APPROACH

We conducted a within-subject study, approved by the university with 24 participants comparing the timestamp difference view with (see Figure 6) and without the LOF algorithm (see Figure 7) and the observations of the measured data with and without K-Prototypes clustering (see Figure 8).

4.1 Experimental Setup

4.1.1 Independent variables. The independent variables in this study are the visualizations (with and without ML techniques) shown to the participants.

4.1.2 Dependent variables. As dependent variables, we measured the users' ability to identify anomalies and effectiveness in identifying insights. Additionally, we assessed the user experience using the User Experience Questionnaire (UEQ)[7] to identify any potential interface design impacts.

4.1.3 Interview and Questionnaire. We conducted unstructured interviews with each participant. Additionally, we added single-choice and open-ended questions to gather demographic information and previous experience using ML-integrated visualizations and traffic detector data. The overall experience using the application and ML integration section asked participants to rate their experience on a 5-point Likert scale. An open question encouraged users to elaborate on how ML integration affected their experience. Screenshots were provided for participants to select visualizations in their responses.

4.1.4 Study Approach. Each participant completed the study within a 60 to 120 minutes time frame, including the unstructured interviews. The difference in expertise caused the varying duration. The web application is a tool requiring some domain knowledge and background information. So, the participants needed a thorough explanation and the possibility to ask questions when using the tool. Except for one interview conducted in English, the conversations and thinking aloud exercises were conducted in German. In order to ensure accessibility, we present translations of the participants' quotes and key points into English. The translation process was done using the DeepL Translator [8] and cross-checked with native speakers. At the beginning of each study session, the participants watched a 7-minute video introducing the study background, ML techniques, and the user interface (see supplementary material for the tutorial text). After watching the video, the participants explored the visualizations by themselves. They evaluated the visualizations showing detector status, duplicate entries, data submission intervals, timestamp differences, and measured data. The visualizations

[2]https://flask.palletsprojects.com/en/2.3.x/, last accessed February 9, 2024

[3]https://developer.mozilla.org/en-US/docs/Web/JavaScript?retiredLocale=de, last accessed February 9, 2024

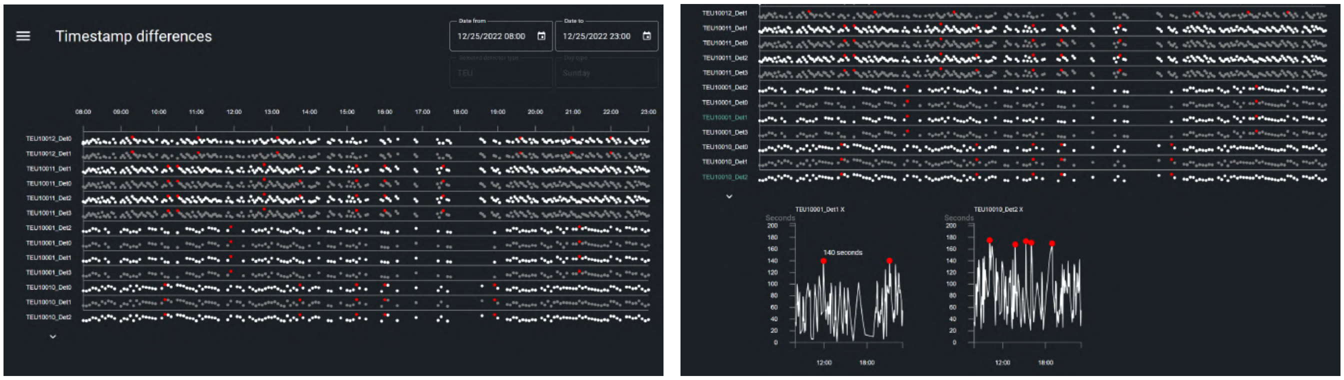
[4]https://react.dev/, last accessed February 9, 2024

[5]https://mui.com/, last accessed February 9, 2024

[6]https://d3js.org/, last accessed February 9, 2024

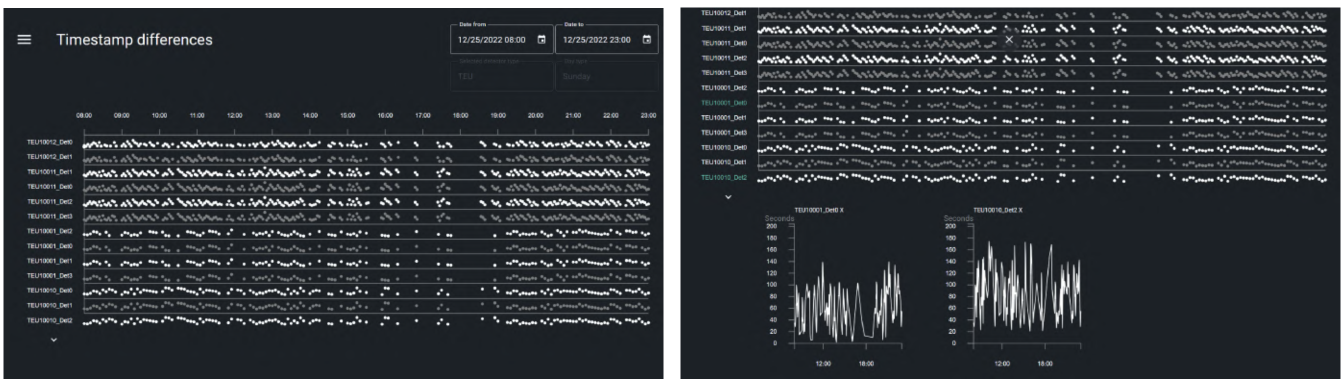
[7]https://www.ueq-online.org/, last accessed February 9, 2024

[8]https://www.deepl.com/en/translator, last accessed February 9, 2024



(a) Overview **(b) Detailed View**

Figure 6: Timestamp Differences View with LOF algorithm marks.



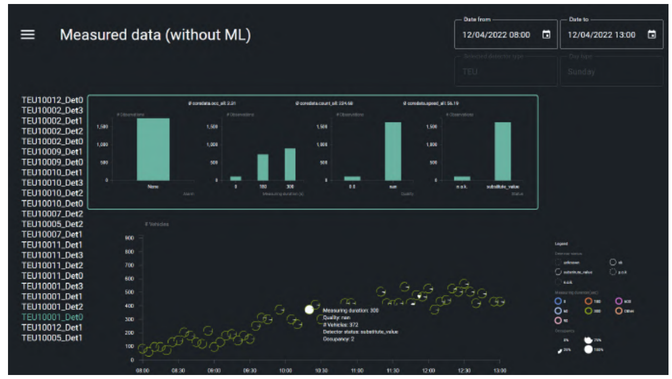
(a) Overview Single Day **(b) Detailed View Single Day**

(c) Multiple Days View

Figure 7: Timestamp Differences View without LOF algorithm marks.

displaying detector status (see Figure 3) and duplicate entries (see Figure 5) show the data without enhancing them with ML techniques for anomaly detection. Due to the relatively small number of participants, experts and non-experts explored identical data for comparable group sizes. In our study design, all participants used the timestamp difference view, with and without the LOF algorithm, and observed measured data with and without clustering. The data for the data submission intervals (see Figure 5c) were manually clustered depending on the interval. Users were asked how they perceived them and the visualizations with horizontal bar charts showing time series data (Figure 3, Figure 5). The participants' screens and voices were recorded during the study to be available

(a) With Clustering



(b) Without Clustering

Figure 8: Measured Data View.

for analysis after the experiment. After users identified potential data anomalies and had no further comments, they completed a digital questionnaire via Microsoft Teams, ensuring compliance with company data policies.

4.2 Participants

24 people aged 25 to 64 ($M = 33.25, SD = 8.03$) participated in the study. Among them, 14 participants identified as male, nine as female, and one preferred not to say. 17 participants had experience using time series data visualization tools, but only eight participants had used tools combining visualization with ML. As depicted in Figure 9, one participant had experience with ML visualization but not with time series data decision-making and had no prior ML knowledge. Eight participants had experience with only visualizations for decision-making with time series. Five used visualizations with ML for time series data. Two users had used visualizations with ML for decision-making; nevertheless, they did not have any ML knowledge. Two other users had ML knowledge and used visualizations for decision-making but never with ML. Six participants stated that they had no prior experience with ML algorithms and visualizations or decision-making visualizations. The participants were invited personally to the study. The study was not compensated and involved recruiting participants from different sources such as the company, university, and acquaintances. From the company, one solution architect, two product managers/owners, two traffic engineers, and one developer participated. Nine participants were domain experts from the company, while the remaining 15 were unfamiliar with the domain (non-experts). The supplementary material offers more information regarding participant profiles.

5 RESULTS

5.1 User Expectations

Participants' qualitative feedback on their expectations from the application were grouped into four categories. In the first category, participants conveyed their expectations to gain insights into specific detectors' functionality and quality, the ability to observe the detector's operation and traffic, and to identify detectors that might not be performing as expected. Eight participants expressed the expectation to delve into raw data, seeking insights about its credibility. Two participants mentioned their desire to export the resulting data for testing applications and bug identification, while one participant expressed the aspiration to optimize and debug traffic data collectors using insights from the application. The second category focused on an intuitive user interface, with two participants highlighting the importance of clear data visualization of time series data and comprehensible representation of anomalies. They expected interactions and various views to facilitate a deeper understanding of the data. The third category emphasized the importance of identifying anomalies for debugging purposes, with five participants explicitly mentioning this need. One participant aptly summarized their expectation: *"More transparency in data. Too much aggregation leads to blind spots. Would rather keep data raw so that diagnostics is easier and more transparent. Use clustering only to identify potential anomalies as a guidance, but access to raw data should still be available."* — P3 (expert). Lastly, participants in the fourth category expressed expectations related to improving traffic management products based on the insights gained from the application. These responses set the background for subsequent analysis to assess whether these expectations were met.

5.2 Experience with Visualization and ML

The study results indicated that the introduction of ML techniques facilitated the exploration of traffic detector data and significantly aided in detecting anomalies. In the timestamp differences visualization (see Figure 6, Figure 7), 14 participants could identify patterns based on location in the timestamp differences view with LOF (Figure 6). They considered the entire patterns to be more anomalous than individual data points. *"I can see a pattern here for the different detector locations."* — P14 (non-expert). 10 participants specifically identified problematic days through the mean delay visualization. Throughout their use of the application, participants discussed anomalies and recommended the inclusion of additional explanations in the visualizations to highlight them. *"Helpful would be a view that gathers all detected anomalies. This would help to get a preselection and an overview."* — P3 (expert). The majority of participants (96%) preferred the timestamp difference

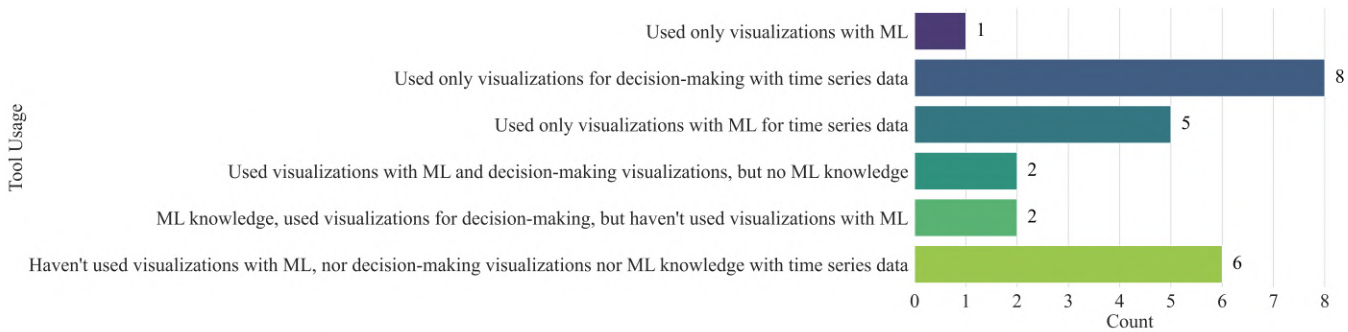


Figure 9: Experience in the usage of visualization tools and ML knowledge.

view, which displays anomalies detected by the LOF algorithm as red dots (Figure 6). *"I can spot anomalies way faster than in a table."* — P13 (non-expert). Another participant (non-expert) explicitly mentioned that the small row-wise scatterplots helped in spotting large delays. Despite their varying levels of ML experience, 11 participants found the measured data view with clustering to be the most helpful for identifying insights, suggesting that ML-based visualizations can effectively support users, even those with limited prior knowledge of ML or the traffic domain. Those who do not frequently work with traffic detector data were able to identify important insights with the help of visualized clustered data $\chi^2(3, N = 24) = (8.96, p = 0.03)$. The resulting Cramer's V of 0.61 indicated that the integration of ML, specifically clustering, in the UI, had a notable impact on users' ability to identify important insights and make informed decisions in the context of analyzing traffic detector data. These results, therefore, support **H2**.

Identifying patterns and trends and interpreting the data with the clustering approach was more manageable for two participants. Another participant stated that they see the data as problematic because the quality seems insufficient. Participants acknowledged the usefulness of clustering, *"The clustering helped significantly in showing trends that would have been harder to find without the clustering"* — P4 (non-expert), but also emphasized that it depends on the quality of input data. They remarked that it could be helpful if there were only a few anomalies and the data quality was not a general problem. *"The values without an alarm value, for example, don't help us, but imagine we had more meaningful data. Then the clustering could provide more insights."* — P1 (expert). However, two traffic engineers did not find value in seeing all attributes in the measured data visualization. Nonetheless, they appreciated the detailed views for status, duplicates, and delay. Some experts recommended using a combination of cluster and non-clustered views and displaying multiple detectors simultaneously. As participants detected anomalies in the data, they confirmed that the ML integration enhanced their ability to spot irregularities. This observation aligned with the expectations set in **H1**, suggesting that ML-based visualizations indeed support the exploration of traffic detector data and improve anomaly detection. We tested **H1** on four aspects (Figure 10): identifying anomalies, clearest data representation, identifying important insights and making informed decisions, and the easiest to understand. The answers to these (with ML vs. without ML) shed light on whether visualizations of ML-integrated data help users explore data and detect anomalies. The result $\chi^2(3, N = 24) = (19.6, p = 0.0002)$ indicated a significant association between the UI (with ML or without ML) and participants' perceptions of the UIs concerning ease of understanding, clarity of data representation, effectiveness in identifying insights, and anomaly detection, supporting **H1**.

5.3 User Experience

We ran the Shapiro-Wilk test for normality for our small sample size and performed a two-tailed t-test to find out whether differences exist between experts and non-experts. The UEQ results indicated that the application generally met user expectations. Participants considered the system to be attractive ("Attractiveness") with a high agreement ($p = 0.93, M = 5.46, SD = 0.82$) while finding the system easy to get familiar with and learn how to use it ("Perspicuity") with a medium agreement ($p = 0.14, M = 4.90, SD = 0.93$). The observation was consistent, with the non-experts saying they needed more time to familiarize themselves with the tool and the domain. Participants also showed medium agreement in the "Efficiency" ($p = 0.45, M = 5.48, SD = 1.01$) and "Stimulation" ($p = 0.62, M = 5.67, SD = 0.8$) categories, respectively. The users felt differently about the amount of control they had while using the application ("Dependability") ($p = 0.76, M = 5.07, SD = 1.02$). "Novelty" ($p = 0.63, M = 5.28, SD = 1.02$) had lower agreement levels too. Overall, the application's performance compared favorably to the UEQ benchmark, with "Attractiveness," "Efficiency," "Stimulation," and "Novelty" rated as "Excellent," "Perspicuity" as "Good," and "Dependability" not comparable due to a removed item. Participants suggested improving filtering, searching options, and providing an overview page for guiding users to anomalies. Some users found ML integration helpful in enhancing their understanding, emphasizing visual highlighting and the ML's simplicity. However, participants desired more detailed information in visualizations, especially for extended data analysis, and suggested UX enhancements for date selectors and information accessibility. No category had a p-value smaller than 0.05, indicating no significant difference between experts and non-experts. Figure 11 shows the overall mean per UEQ category.

6 DISCUSSION

Our study involved the exploration of traffic detector data visualizations to detect anomalies. We investigated two distinct scenarios: one with the integration of ML techniques and one without, as

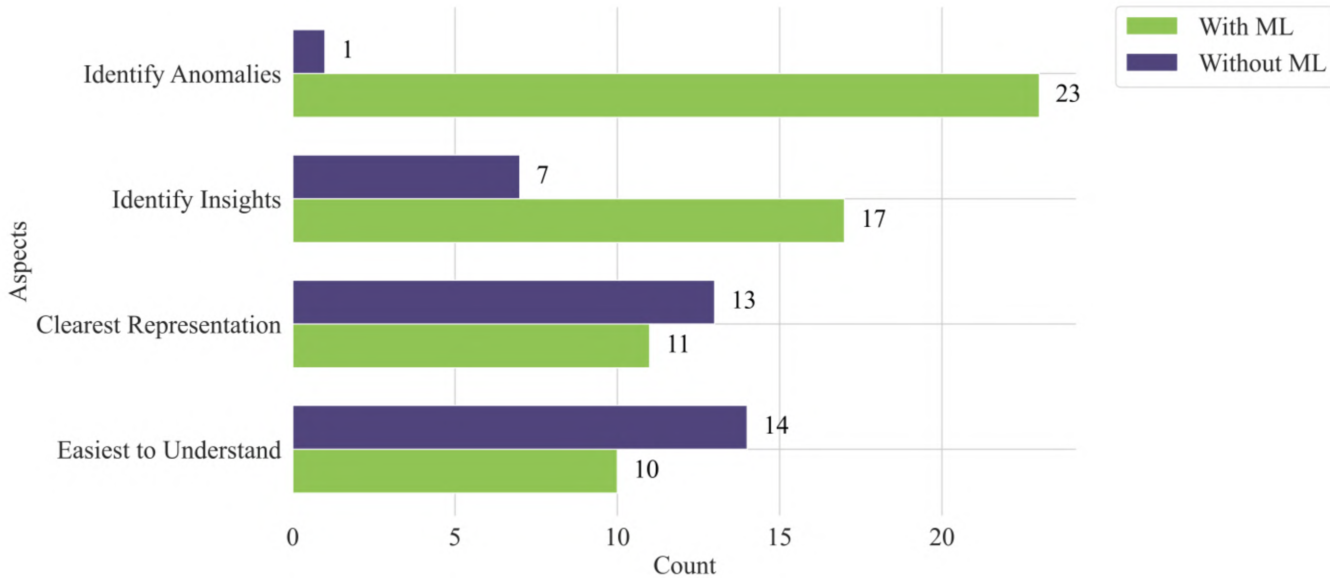


Figure 10: User preferences for UI (With ML vs. Without ML).

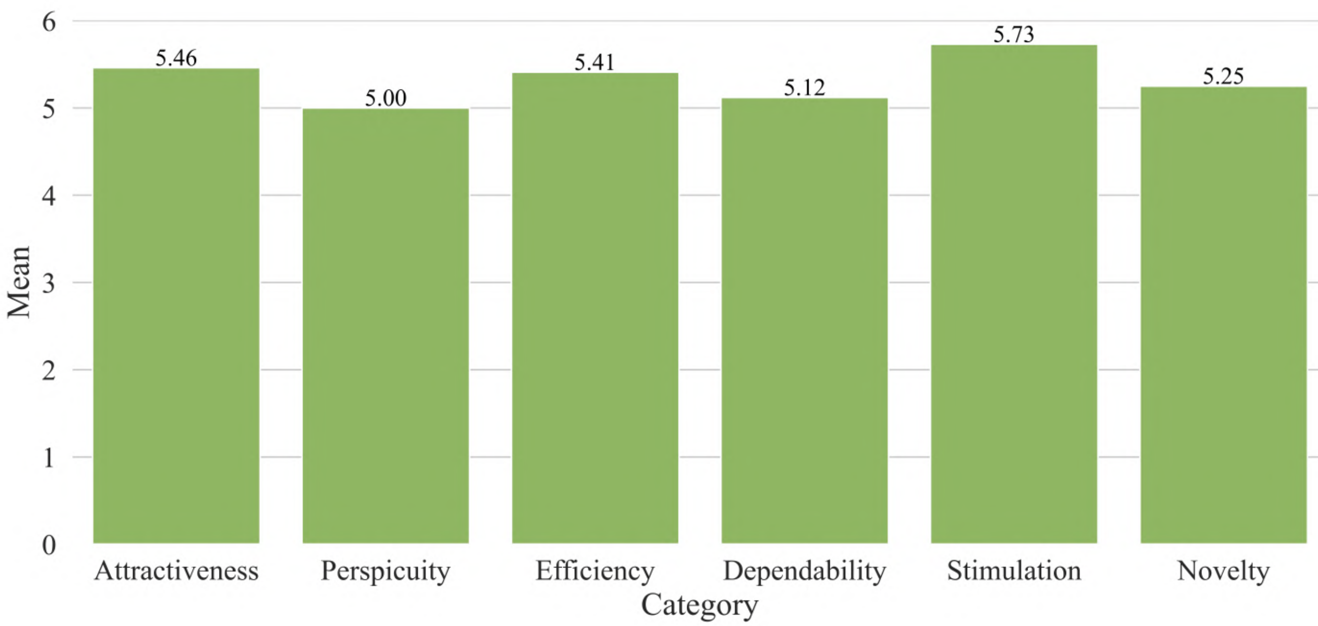


Figure 11: Mean per UEQ category.

outlined in **RQ1**. Additionally, we aimed to determine if the ML integration enhances anomaly detection for users with different levels of domain knowledge, aligning with **RQ2**. Our results validate both **H1** and **H2**, providing support for our initial hypotheses. In the following discussion, we will unravel the significant insights from our analysis.

Allowing users to customize the features to include in the clustering could benefit future application versions. The participants expressed the necessity for manual inclusion or exclusion of specific features, reflecting the need for flexibility. Additionally, as we explore visualizations that refrain from aggregating information, evaluating their effectiveness is crucial, especially in accommodating more data features. Taking cues from examples like *DendroMap* for image data [6], manual cluster setting or adjustment can offer flexibility in the clustering process.

Optimizing clustering techniques for outlier detection is essential. As many conventional clustering methods inadvertently identify anomalies [5]. Evaluating specialized outlier detection methods such as FastABOD, explicitly designed for high-dimensional data [10], holds the potential to provide valuable insights. Furthermore, considering the choice of distance measure in scenarios involving mixed-type data, such as employing Euclidean distance for continuous variables and Hamming distance for categorical variables, can significantly impact clustering outcomes [21].

Data quality is a cornerstone of algorithm performance. Prioritizing data quality is essential for algorithm effectiveness. Addressing data insufficiency issues should precede the pursuit of better algorithms. Filters within the application can also influence algorithm performance. Consider implementing filters for detector type and weekdays, resulting in varied data due to weekday-dependent traffic fluctuations. Creating a benchmark dataset marked with anomalies beyond the norm is valuable for improving classification algorithms. This involves dataset labeling for precise efficiency assessments. Inspired by Chegini et al. [11], an anomaly detection and labeling tool enriched by expert insights can be developed. Additionally, integrating active learning strategies can reduce user labeling efforts and enhance the system's anomaly detection capabilities [19].

To enhance user trust, clear explanations of anomaly detection and clustering are needed. When considering the inclusion of labeling features with LOF and clustering, a prime objective should be to empower users with guidance toward relevant observations. From a user experience perspective, the requirement of extensive explanations concerning anomaly detection, particularly the complexities of the clustering process, is pivotal. Users had expressed a strong desire to understand why algorithms like K-Prototypes cluster data in specific ways and why LOF flags specific timestamp differences as anomalies. The desire to understand the algorithms better comes from a sincere intention to evaluate how well they work and how reliable they are. It is not just about labeling data but also about making users feel confident in using the application. So, finding ways to give personalized guidance based on each user's needs using ML can make the application much more helpful.

7 LIMITATIONS AND FUTURE WORK

Most participants, experts, and non-experts expressed a need for more detailed information in the visualizations. Addressing their requests for more detailed visualizations and improved usability, such as a different time selector, is essential for enhancing the overall user experience. Moreover, there is a potential to create a categorized dataset for accurate anomaly identification, fulfilling the user demand for benchmark metrics and labeled datasets. However, algorithm accuracy and efficiency concerns must be addressed through proper training and explanation. Moreover, combining user feedback, usability evaluations, and cognitive effort assessments, such as the System Usability Scale (SUS) [8], can guide the application's refinement.

Furthermore, some potential **threats to the validity** should be considered. To enhance the generalisability of the results with 24 participants, our study involved diverse profiles (ages 25 to 64) to ensure comparable group sizes for both experts and non-experts. To mitigate potential bias, a specific order was imposed on participants, alternating the presentation of visualizations with and without the LOF algorithm. However, despite these precautions, order effects may still influence participants' responses, particularly regarding their ability to identify anomalies and insights. Given the 60-120 minutes experiment duration and extensive data observation, minimal learning or fatigue effects were expected for treatment ordering in controlling learning effects. We encouraged the users to select different dates in the different visualizations but did not want to restrict them to specific dates. The goal was to let them explore freely. There were no visible learning effects during the study. To tackle this, we propose a future study with more participants, separating them into groups (with ML vs. without ML) and measuring learning curves [25]. Regarding ML algorithms, while we used K-Prototypes for clustering, concerns still remain about not employing density-based clustering, e.g., Density-Based Spatial Clustering of Applications with Noise (DBSCAN). We avoided this as Python's scikit-learn DBSCAN is unable to handle mixed-type data. However, literature [36] suggests that Gower distance with DBSCAN solves mixed data problems. Objective metrics, such as the number of anomalies detected or the time required for specific tasks, were not employed due to the study's design constraints as the study duration varied depending on the participants' questions. Hence, the overall time was not representative, and no helpful mean could be calculated. Future work could benefit from integrating learning sessions to familiarize participants with the tool. Consecutive second sessions in which they find anomalies independently. Afterward, the latter would be timed, and the anomalies found would be counted.

8 CONCLUSION

Our work deals with a large traffic detector dataset containing numerical and categorical data. Since location information is absent, the developed application covers a research gap in handling traffic detector data without location attributes and creating improved time series visualizations. The application visualizes time series traffic detector data using innovative techniques tailored to time series data characteristics. Furthermore, we explored the potential of interactive visualizations in detecting anomalies effectively while considering the varying levels of user expertise and familiarity with the raw data. The study results demonstrate that applying ML techniques facilitates the exploration of traffic detector data and also reveals that ML-based visualizations can support users with scarce domain knowledge in ML or the traffic domain. This research contributes to the field of traffic data analysis. The findings serve as a basis for extended investigations into hardware and program quality improvements in the company. It paves the way for future advancements in visualizing and understanding complex datasets for traffic management and decision-making processes.

ACKNOWLEDGMENTS

This research was funded by Elitenetzwerk Bayern.

REFERENCES

[1] James Abello, Fred Hohman, Varun Bezzam, and Duen Horng Chau. 2019. Atlas: local graph exploration in a global context. In *Proceedings of the 24th International*

Conference on intelligent user interfaces. 165–176. https://doi.org/10.1145/3301275.3302275

[2] Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. 2016. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications* 60 (2016), 19–31. https://doi.org/10.1016/j.jnca.2015.11.016

[3] Nuno Basurto, Carlos Cambra, and Alvaro Herrero. 2022. A visual tool for monitoring and detecting anomalies in robot performance. *Pattern Analysis and Applications* 25, 2 (May 2022), 271–283. https://doi.org/10.1007/s10044-021-01053-0

[4] Michael Behrisch, Michael Blumenschein, Nam Wook Kim, Lin Shao, Mennatallah El-Assady, Johannes Fuchs, Daniel Seebacher, Alexandra Diehl, Ulrik Brandes, Hanspeter Pfister, et al. 2018. Quality metrics for information visualization. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 625–662. https://doi.org/10.1111/cgf.13446

[5] Samir Brahim Belhaouari et al. 2021. Unsupervised outlier detection in multidimensional data. *Journal of Big Data* 8, 1 (2021), 1–27. https://doi.org/10.1186/s40537-021-00469-z

[6] Donald Bertucci, Md Montaser Hamid, Yashwanthi Anand, Anita Ruangrotsakun, Delyar Tabatabai, Melissa Perez, and Minsuk Kahng. 2022. DendroMap: Visual Exploration of Large-Scale Image Datasets for Machine Learning with Treemaps. http://arxiv.org/abs/2205.06935 arXiv:2205.06935 [cs].

[7] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 93–104.

[8] John Brooke. 1995. SUS: A quick and dirty usability scale. *Usability Eval. Ind.* 189 (11 1995).

[9] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 108–122.

[10] Guilherme O. Campos, Arthur Zimek, Jörg Sander, Ricardo J. G. B. Campello, Barbora Micenková, Erich Schubert, Ira Assent, and Michael E. Houle. 2016. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery* 30, 4 (July 2016), 891–927. https://doi.org/10.1007/s10618-015-0444-8

[11] Mohammad Chegini, Jürgen Bernard, Philip Berger, Alexei Sourin, Keith Andrews, and Tobias Schreck. 2019. Interactive labelling of a multivariate dataset for supervised machine learning using linked visualisations, clustering, and active learning. *Visual Informatics* 3, 1 (March 2019), 9–17. https://doi.org/10.1016/j.visinf.2019.03.002

[12] Wei Chen, Fangzhou Guo, and Fei-Yue Wang. 2015. A Survey of Traffic Data Visualization. *IEEE Transactions on Intelligent Transportation Systems* 16, 6 (Dec. 2015), 2970–2984. https://doi.org/10.1109/TITS.2015.2436897

[13] Kristin A Cook and James J Thomas. 2005. *Illuminating the path: The research and development agenda for visual analytics*. Technical Report. Pacific Northwest National Lab.(PNNL), Richland, WA (United States).

[14] Constantinos Costa, Georgios Chatzimilioudis, Demetrios Zeinalipour-Yazti, and Mohamed F. Mokbel. 2017. Towards Real-Time Road Traffic Analytics Using Telco Big Data. In *Proceedings of the International Workshop on Real-Time Business Intelligence and Analytics* (Munich, Germany) *(BIRTE '17)*. Association for Computing Machinery, New York, NY, USA, Article 5, 5 pages. https://doi.org/10.1145/3129292.3129296

[15] Rene Cutura, Michaël Aupetit, Jean-Daniel Fekete, and Michael Sedlmair. 2020. Comparing and Exploring High-Dimensional Data with Dimensionality Reduction Algorithms and Matrix Visualizations. In *Proceedings of the International Conference on Advanced Visual Interfaces*. ACM, Salerno Italy, 1–9. https://doi.org/10.1145/3399715.3399875

[16] Xin Fan, Chenlu Li, Xiaoru Yuan, Xiaoju Dong, and Jie Liang. 2019. An interactive visual analytics approach for network anomaly detection through smart labeling. *Journal of Visualization* 22, 5 (Oct. 2019), 955–971. https://doi.org/10.1007/s12650-019-00580-7

[17] Jean-Daniel Fekete, Jarke J. van Wijk, John T. Stasko, and Chris North. 2008. The Value of Information Visualization. In *Information Visualization*, David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Andreas Kerren, John T. Stasko, Jean-Daniel Fekete, and Chris North (Eds.). Vol. 4950. Springer Berlin Heidelberg, Berlin, Heidelberg, 1–18. https://doi.org/10.1007/978-3-540-70956-5_1 Series Title: Lecture Notes in Computer Science.

[18] Peter Filzmoser and Mariella Gregorich. 2020. Multivariate Outlier Detection in Applied Data Analysis: Global, Local, Compositional and Cellwise Outliers. *Mathematical Geosciences* 52, 8 (Nov. 2020), 1049–1066. https://doi.org/10.1007/s11004-020-09861-6

[19] Benedikt Grimmeisen and Andreas Theissler. 2020. The machine learning model as a guide: pointing users to interesting instances for labeling through visual cues. In *Proceedings of the 13th International Symposium on Visual Information Communication and Interaction*. ACM, Eindhoven Netherlands, 1–8. https://doi.org/10.1145/3430036.3430058

[20] Zhexue Huang. 1998. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery* 2, 3 (1998), 283–304.

[21] Jarrett Jimeno, Madhumita Roy, and Cristina Tortora. 2021. Clustering Mixed-Type Data: A Benchmark Study on KAMILA and K-Prototypes. In *Data Analysis and Rationality in a Complex World*, Theodore Chadjipadelis, Berthold Lausen, Angelos Markos, Tae Rim Lee, Angela Montanari, and Rebecca Nugent (Eds.). Springer International Publishing, Cham, 83–91.

[22] Areti Karamanou, Petros Brimos, Evangelos Kalampokis, and Konstantinos Tarabanis. 2023. Exploring the Quality of Dynamic Open Government Data for Developing Data Intelligence Applications: The Case of Attica Traffic Data. In *Proceedings of the 26th Pan-Hellenic Conference on Informatics* (Athens, Greece) *(PCI '22)*. Association for Computing Machinery, New York, NY, USA, 102–109. https://doi.org/10.1145/3575879.3575974

[23] Boris Kovalerchuk, Răzvan Andonie, Nuno Datia, Kawa Nazemi, and Ebad Banissi. 2022. Visual Knowledge Discovery with Artificial Intelligence: Challenges and Future Directions. http://arxiv.org/abs/2205.01296 arXiv:2205.01296 [cs].

[24] Agus Trisnajaya Kwee, Meng-Fen Chiang, Philips Kokoh Prasetyo, and Ee-Peng Lim. 2018. Traffic-Cascade: Mining and Visualizing Lifecycles of Traffic Congestion Events Using Public Bus Trajectories. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino, Italy) *(CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 1955–1958. https://doi.org/10.1145/3269206.3269216

[25] Sébastien Lallé, Dereck Toker, Cristina Conati, and Giuseppe Carenini. 2015. Prediction of Users' Learning Curves for Adaptation while Using an Information Visualization. In *Proceedings of the 20th International Conference on Intelligent User Interfaces* (Atlanta, Georgia, USA) *(IUI '15)*. Association for Computing Machinery, New York, NY, USA, 357–368. https://doi.org/10.1145/2678025.2701376

[26] Shusen Liu, Dan Maljovec, Bei Wang, Peer-Timo Bremer, and Valerio Pascucci. 2017. Visualizing High-Dimensional Data: Advances in the Past Decade. *IEEE Transactions on Visualization and Computer Graphics* 23, 3 (March 2017), 1249–1268. https://doi.org/10.1109/TVCG.2016.2640960

[27] Linhao Meng, Yating Wei, Rusheng Pan, Shuyue Zhou, Jianwei Zhang, and Wei Chen. 2021. VADAF: visualization for abnormal client detection and analysis in federated learning. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 11, 3-4 (2021), 1–23.

[28] Tauno Metsalu and Jaak Vilo. 2015. ClustVis: a web tool for visualizing clustering of multivariate data using Principal Component Analysis and heatmap. *Nucleic acids research* 43, W1 (2015), W566–W570.

[29] Jundee Mark G. Molina, Angelica S. Sotis, Rogamestica C. Pascual, James Earl D. Cubillas, and Junrie B. Matias. 2023. Development of Real-Time Traffic Monitoring and Visualization System Using Stationary Roadside Sensor. In *Proceedings of the 2022 11th International Conference on Networks, Communication and Computing* (Beijing, China) *(ICNCC '22)*. Association for Computing Machinery, New York, NY, USA, 154–160. https://doi.org/10.1145/3579895.3579919

[30] Pratik Rajendra Ratadiya, Ryan Alimo, Brian Kahovec, Farrokh Vatan, and William Gilbert. 2024. RAMAN: Robust Approaches for Multimodal ANomaly detection in Mars Rover Power Systems. In *AIAA SCITECH 2024 Forum*. 1378.

[31] Maria Riveiro, Mikael Lebram, and Marcus Elmer. 2017. Anomaly Detection for Road Traffic: A Visual Analytics Framework. *IEEE Transactions on Intelligent Transportation Systems* 18, 8 (Aug. 2017), 2260–2270. https://doi.org/10.1109/TITS.2017.2675710

[32] Sauravjyoti Sarmah and Dhruba K Bhattacharyya. 2010. An effective technique for clustering incremental gene expression data. *IJCSI International Journal of Computer Science Issues* 7, 3 (2010), 31–41.

[33] James Schaffer, Prasanna Giridhar, Debra Jones, Tobias Höllerer, Tarek Abdelzaher, and John O'donovan. 2015. Getting the message? A study of explanation interfaces for microblog data analysis. In *Proceedings of the 20th international conference on intelligent user interfaces*. 345–356.

[34] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. 2022. Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment* 15, 9 (2022), 1779–1797.

[35] Ben Shneiderman. 1996. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE symposium on visual languages*. IEEE, 336–343.

[36] Denys Teslenko, Anna Sorokina, Kyrylo Smelyakov, and Oleksii Filipov. 2023. Comparative Analysis of the Applicability of Five Clustering Algorithms for Market Segmentation. In *2023 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*. 1–6. https://doi.org/10.1109/eStream59056.2023.10134796

[37] Cagatay Turkay, Arvid Lundervold, Astri Johansen Lundervold, and Helwig Hauser. 2012. Representative Factor Generation for the Interactive Visual Analysis of High-Dimensional Data. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (Dec. 2012), 2621–2630. https://doi.org/10.1109/TVCG.2012.256

[38] Raihan Ul Islam, Mohammad Shahadat Hossain, and Karl Andersson. 2018. A novel anomaly detection algorithm for sensor data under uncertainty. *Soft*

Computing 22, 5 (2018), 1623–1639.

[39] Vera van Zoest, Alfred Stein, and Gerard Hoek. 2018. Outlier Detection in Urban Air Quality Sensor Networks. *Water, Air, & Soil Pollution* 229, 4 (April 2018), 111. https://doi.org/10.1007/s11270-018-3756-7

[40] Pak Chung Wong, Han-Wei Shen, Christopher R. Johnson, Chaomei Chen, and Robert B. Ross. 2012. The Top 10 Challenges in Extreme-Scale Visual Analytics. *IEEE Computer Graphics and Applications* 32, 4 (July 2012), 63–67. https://doi.org/10.1109/MCG.2012.87

[41] Austin P Wright, Peter Nemere, Adrian Galvin, Duen Horng Chau, and Scott Davidoff. 2023. Lessons from the Development of an Anomaly Detection Interface on the Mars Perseverance Rover using the ISHMAP Framework. In *Proceedings of the 28th International Conference on Intelligent User Interfaces* (Sydney, NSW, Australia) *(IUI '23)*. Association for Computing Machinery, New York, NY, USA, 91–105. https://doi.org/10.1145/3581641.3584036