Towards a data-driven public bus operation: Monitoring bus door anomalies

Ilsu Kazkayasi
Department of Electrical and Computer Engineering
Technical University of Munich
Munich, Germany
ilsu.kazkayasi@tum.de

***Abstract*—Door system failure of a public transportation bus causes delays, trip cancellations and operational inefficiencies for operating companies. In order to prevent this, a predictive maintenance method is needed to be developed specific to Munich public transportation busses. However nature of the data logged from bus data systems prevents defining anomaly patterns in an easy manner. Moreover the type of anomalies that are wanted to be detected, makes the problem even more complex. This work aims to develop a method that minimizes data transfer size from bus to the backend cloud, to investigate and to manipulate the raw data, to propose visualization methods for detecting anomaly patterns and lastly to create use cases for anomaly detection. All of the aims are reached successfully except the last one. Defining use cases are left for future work.**

I. Introduction

The trend of urbanization and increasing housing prices in Munich created a great demand on the means of public transport. As the number of passengers commuting everyday via public transport increased, number of trips conducted per day is also increased. Increase in usage of public transportation assets results in an increased rate of system and machine failures [1]. These failures cause severe problems for public transportation operations like unexpected downtime, delays, trip cancellations, increasing costs, and unreliability.

In operation a bus door opens up to more than 1000 times a day. In parallel with that a large fraction of downtimes is caused by door systems. In order to minimize inefficiencies caused by door system failures a door anomaly detection system is needed to be developed.

A single bus on operation in Munich creates 2 GB of data per day. Size of the raw data puts a heavy load on transmission and processing. In order to accomplish an effective monitoring and detection, a method of collecting and transmitting only the relevant data during operation must be developed.

In this research project the possibilities to infer bus door failures from measurements of the opening and closing patterns of the doors near stops are explored, thereby reducing the necessary data transmission volume to the minimum.

The analysis of bus door status patterns is first conducted on a historical data set recorded from a fleet of eight busses in Munich. Using the insights from the analysis of historical bus door data, a filter function is designed, which will enable the efficient remote monitoring of bus door anomalies and passenger flow patterns, by only transmitting relevant data attributes collected only at a certain vicinity to bus stops.

II. Related Work

Failure prediction of doors in railway systems is already investigated by [1]. In the literature, no similar work is found on public transportation busses. In [1] machine learning methods are applied to detect and predict anomalies. Machine learning methods are out of scope of this work.

On the other hand detecting anomalies in data has been studied since many years and a large variety of well-established detection techniques are present [3] [4] [5]. In [5] those techniques are summarized based on their underlying approach. According to [5] to identify the best approach for the anomaly detection problem, three aspects of the problem must be identified carefully; nature of input data, type of anomaly and data labels. Investigated data for this work falls under the spatio-temporal data category, since it is a time series data and data points are related to each other in space (if they are in proximity to a bus stop or not).

The type of anomaly tried to be detected is categorized both as contextual and behavioral. That means definition of anomalies are related to the context, for example, a door opened at a point far away from any bus stop may signify an anomaly, however this behavior is normal in proximity of a bus stop. It is also defined as behavioral since each different bus line and bus stop may have different patterns. For example first and last stops of the bus lines has different behavioral pattern than a intermediate stop, which can change definition of the anomalies. In that case [5] suggests that, anomaly must be determined using the values for the behavioral attributes within a specific context.

Raw data used for this work is not pre-labeled for malfunctioning door systems, which means, data collection systems don't recognize a malfunctioning door. This kind of labeling is not possible for this work, since anomaly that is searched for is not anomaly of a single data instance, but an anomaly pattern formed by a collection of many data instances. Therefore [5] suggests unsupervised anomaly detection.

As suggested in the related work data is separated and labeled according to the context. This includes whether or not a data point is in proximity to a bus stop, the name of the proximity bus stop and time slot of the day data is

recorded. Also data examined separately for different bus lines, on different days and for different directions.

With the help of created context and behavioral groups visualization is done. With this, defining anomaly cases therefore creating a labeling for anomaly detection is aimed, however no anomalies are observed in the investigated data sets.

III. Methodology

A. Materials

Data from the bus is collected from three different sources. Location information of the bus comes from a GPS source. Another data source is specific to the bus manufacturer and follows the FMS (fleet management system) standards [2]. This interface gives data from bus internal network as fuel level, engine speed and information about door statuses. FMS also has an attribute called tell-tale status, which detects anomalies of several important systems and provide the output as labels. However this attribute does not include door systems [2]. [2] creates only a set of binary data on bus door states which indicates if a door is opened, locked and enabled. A third data source is a proprietary protocol used on displays of the bus, ticket machine timing and door states. All data is collected via an IoT gateway onboard and sent to a cloud-based backend. A custom software runs on the custom onboard unit and decides which information is useful to be sent.

Output filter function of this project is designed to be implemented in afore mentioned software.

The filter function is designed in python to be converted into a configuration file using java script for implementing on the onboard unit. Output data from that function is planned to be collected and analyzed remotely.

B. Approach

The function minimizes the data set by filtering out only relevant attributes of the data in 50 meters proximity to any bus stop. The idea of designed solution is summarized in Fig. 1.

The analysis is first conducted on a historical data set which is pulled from the cloud in csv format. Each analyzed data set is recorded from a single bus through the day. The bus can work on different lines in a single day. Time resolution is in milliseconds, defined by a timestamp attribute.

C. Attributes of Data Sets

Raw data pulled from the cloud is neither tidy nor clean. It needs to be manipulated before being able to be processed. This is due to facts that, data is coming from different sources and each attribute has its own sampling rate. An unmanipulated data set taken from the cloud has more than 7 million data points.

As first step of the manipulation, data is grouped into bins of 1 second. Then attributes of binned data set are examined carefully in the pursuit of obtaining a minimized data set.

Some important observations about attributes are summarized below.

The raw data has been enriched with a trip indexing added. This attribute counts number of trips of a bus line in between

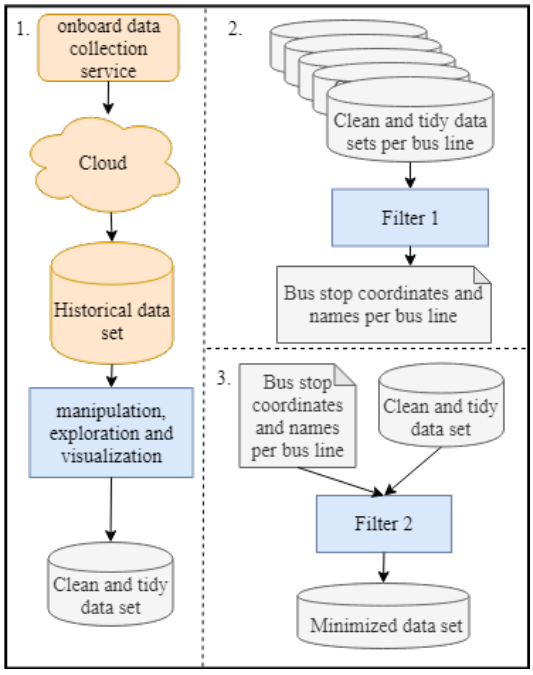

Fig. 1. 1. Data is collected on the bus and sent to the cloud–based backend. Historical data sets are pulled from the backend, processed and a tidy data set is obtained. 2. A function is created to detect location of bus stops for each bus line. 3. Tidy data set is filtered out by using created bus stop coordinates and a minimum data set for anomaly detection is obtained.

two destinations (1 for going, 2 for coming back). However, a bus line usually has more than 1 destination. Depending on the constructions works and operation hours, buses can operate till a middle stop. In that case trip number starts from 1 again.

Some off-service routes are detected in the data with destination names “einrückfahrt”, “ausrückfahrt”, “dienstfahrt”. Since these routes are out of interest of the study they are excluded.

In wheel-based speed attribute some outlier data points are detected which have a value more than 250km/h (Fig. 2). These outliers are also excluded. Moreover, in order to validate the integrity of doors data it is controlled if wheel-based speed is always 0 when any of the doors is open. These two attributes are found to be coherent.

In FMS standard description, it is stated that “Most of the values are reliable after ca. 10 seconds after ´ignition on”. Therefore, each first 10 seconds after ignition state changes from 0 to 1 are dropped.

Destination, current stop, line number, timestamp, latitude longitude and doors are found to be most relevant attributes to visualize door usage pattern. Most useful attribute for analyzing door status pattern is “doors” attribute. This attribute can contain status information up to 10 different doors of a bus [2]. All buses analyzed in this study have 3 doors and 2 additional doors if they are equipped with a trailer. Therefore, information about first 5 doors out of 10 is used to obtain door status information only. This attribute has binary value, 1 for open door and a 0 for closed door.

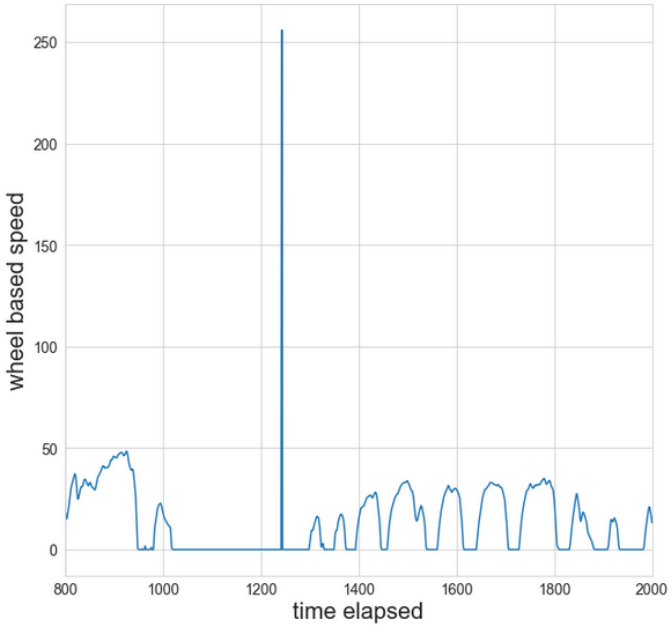

Fig. 2. Wheel based speed of the bus against time elapsed in seconds. The outlier of 250km/h is filtered out.

Current stop is another important attribute for this study. This attribute is updated to the next stop, when the bus doors are closed for the last time at a stop.

In the pursuit of findings, a smaller data set is created only with relevant attributes. Missing values of these attributes then are filled by fill forward method.

D. Filter Design

After manipulation is done, data exploration and visualization are performed to find out the best approach for filter function design.

As first step, a function is created that obtains bus stop coordinates from the change in current stop attribute. In Fig. 3 bus stops are shown with yellow circles of 50 m diameter around coordinates of the data point where current stop name changes. Data points, where any of the doors are detected open are visualized as blue dots. Fig. 3 shows clearly that the created function works accurately.

Then the main function is designed to filter historical data with the output of first function. For this, geo-fencing method is adopted. Distances of data points to bus stops are calculated and then, data points which are located more than 50 m away from a bus stop are filtered out. 50 m is chosen as filter value since the buses with trailer are 23 m in length.

After filter is designed, filtered data is visualized in a fourfold manner, for detecting door anomalies and passenger flow patterns.

IV. Results

A bus door is expected to be opened at a stop just once and closed before bus leaves the stop. However, it is observed that at some of the stops some doors open and close more than

Fig. 3. Yellow circles has 50 m diameter and centered to bus stops. Blue dots are the points where at least one door is open.

once, before bus lefts the stop. Since this behavior can be an indicator of anomalies, two additional attributes are created namely count and duration, which represent respectively how many times and for how long a door stays open consecutively, at the same stop for the same trip. In total 12 historical data sets are visualized and analyzed. However, with respect to the report length only one example of each output plot is shown in this report.

Door opening durations at each stop can give clues about passenger flow pattern, hence it is computed and plotted as a grouped bar plot. Each door is represented by different plots and each plot per door represents one day, one line number and a single destination (Fig. 4). Count attribute is also represented by different colored bar plots. From Fig. 4 it is observed that, bus doors are opened at the first stop multiple times while waiting for start of the trip. Also, duration of opening at the first stop is much longer than of an intermediate stop. From Fig. 4 it can be easily seen that Moosach Bahnhof is the first stop of the line 51 where door is opened 5 times while waiting. At Laimer Platz, Holzapfelkreuth and Romanplatz more passengers have used the doors than other stops.

To improve the visualization for detecting door anomalies, opening instances are counted at each stop during a day for a bus line for each destination (Fig. 5). Different colored bars represent the count attribute. Fig. 5 shows the count plot of the

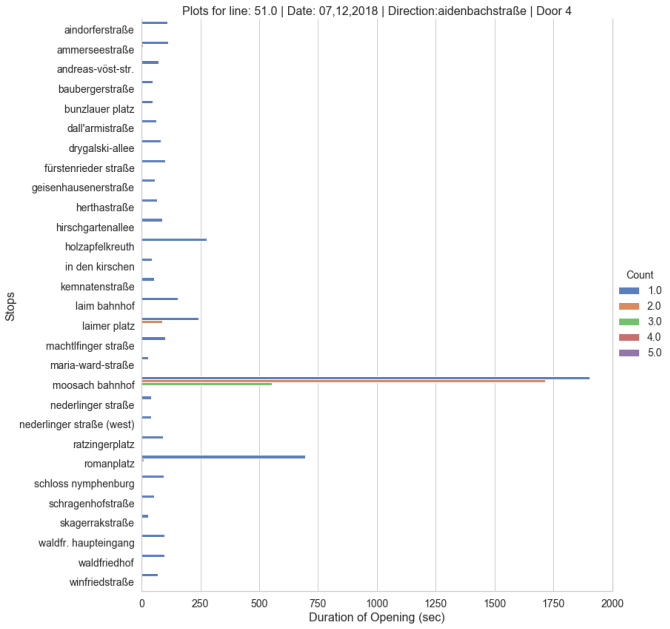

Fig. 4. Opening duration of the fourth door, in seconds, at each stop of line 51 for destination Aidenbachstraße.

same data as Fig. 4. It is observed that at most of the stops doors are opened only once per trip. At moosach bahnhof doors are opened five times, and this only happened once. Since Moosach Bahnhof is the first station this behavior does not signify an anomaly.

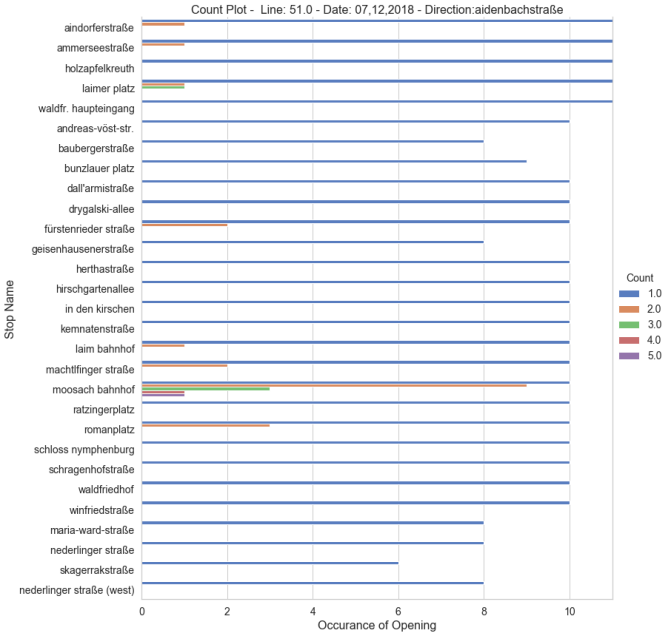

Fig. 5. Count of all door opening instances at each stop of line 51 for destination Aidenbachstraße.

A third kind of plot is created in order to visualize passenger flow patterns during the day. This is again a count plot where vertical axis shows time at the start and end of each trip instead of stops, colors represents different doors on bus. Horizontal axis shows how many times in total doors are opened during that trip.

From Fig. 6 it is observed that all doors are used more often at noon time which indicates more passengers using public transportation at that time of the day. Door 2 is the most often used door during the course of the day. This plot indicates if a door is broken: If a door is not opened any more after a time point that indicates a broken door. However, in the investigated data sets no broken doors are observed.

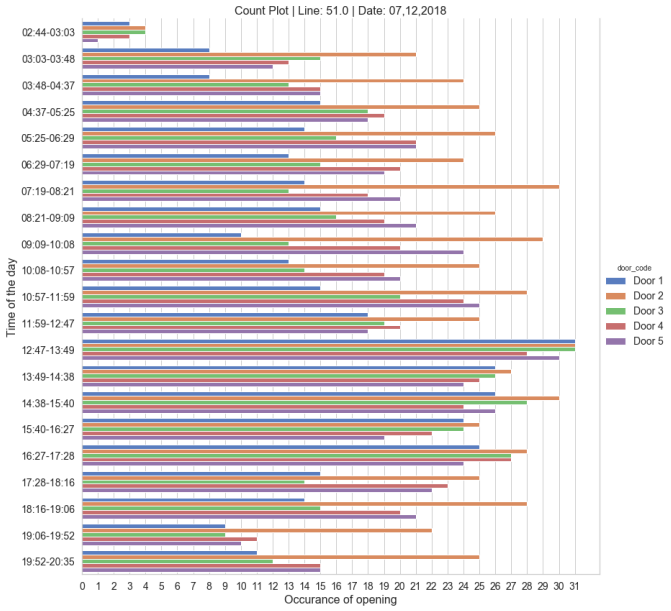

Fig. 6. Count of door opening instances for each door during different trips of line 51 for destination Aidenbachstraße.

Lastly a kernel density estimation plot is drawn for all doors. Only up to 40 seconds of openings are considered, since an average door opening duration is between 5 to 20 seconds and more than 40 seconds is observed only if the bus waits long period of times at a stop, for example at the first stop . It is seen from Fig. 7 that doors are mostly stayed open around 10 seconds and trailer doors (door 4 and 5) are stayed open for shorter times than main doors of the bus.

Another result of this work is the detection of precise bus stop locations from the bus data. This is done by applying the first filter function (Fig. 1) on 41 different data sets, which are sub-sets of raw data sets collected from buses but contains smaller number of attributes. As the output, a static json file is created which holds coordinates and names of each bus stop for each line. In order to apply geo-fencing method on onboard unit, location of bus stops must be known to the function, creating them onboard every time will result in exclusion of the points before the bus stop. The static json file created with the first filter function can be used directly on custom onboard unit to detect vicinity of stops accurately during operation.

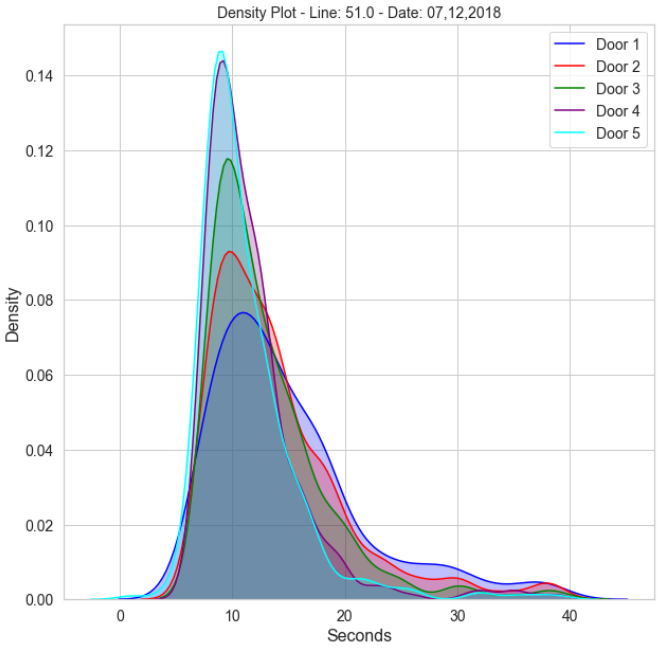

Fig. 7. Kernel Density Estimation plot for all doors. Opening durations only up to 40 seconds are considered.

A. *Evaluation of Results*

Unfortunately no door anomalies are detected with the created method for analyzed data sets. This is most probably due to the limited amount of data analyzed for this work. Although the related data was not enough for detecting anomalies and therefore creating use cases, amount of data processed was relatively big (approximately 5 GB).

A bus generates up to 2 GB of data per day, and a bus fleet of a city with the size of Munich consists of more than 400 vehicles. Therefore, transferring raw data from each bus every day and processing it for door anomaly detection is not an efficient method. It is necessary to minimize amount of data transferred from the bus in order to create an effective system for anomaly detection.

In the created method it is aimed to filter data onboard before it is saved, and transfer only the related data. After implementing this filter the data received by the backend should be the same data set as the output data set of the function applied on raw data in this work which is named as minimized data set in Fig. 1.

Raw data set received from the backend cloud for this work has size up to 500 MB. Minimized data set output of the filter has up to 60 KB size. That means created method minimizes data size to the order of 0.00012.

V. Conclusion

During this internship a custom function is created that allows effective remote monitoring of door anomalies. The created function achieves 0.00012 times smaller data transfer size from onboard unit to the cloud, therefore allows processing of larger data sets effectively for defining anomaly patterns. Although anomaly patterns could not be detected from the input data sets analyzed in this work, different contexts and behaviors are defined and four different ways of visualization are proposed to apply on bigger data sets. Moreover these plots created can be used to interpret passenger flow patterns of Munich public transportation busses. A second function is designed (first filter function in Fig. 1) to create a json file which holds precise bus stop coordinates per bus line from the bus data itself. This file can be directly utilized on onboard unit to detect bus stop locations effectively during operation. Converting python code into java script is left as future work to experts developing the data collection service. Also, to define meaningful use cases for door anomalies in the future, analysis on much bigger data is advised.

References

[1] P. Pereira, R.P. Ribeiro, J. Gama, “Failure Prediction - An Application in the Railway Industry” presented at International Conference on Discovery Science, 2014, pp 264-275.

[2] HDEI / BCEI Working Group, “FMS-Standard description”, version 03, September 2012.

[3] A. Patcha, J.M. Park, “ An overview of anomaly detection techniques: Existing solutions and latest technological trends”, Computer Networks, Volume 51, Issue 12, 2007, pp 3448-3470.

[4] M.A. Hayes, M. AM Capretz, “ Contextual anomaly detection framework for big sensor data”, Journal of Big Data, Volume 2, Article 2 , Springer, 2015.

[5] V. Chandola, A. Banerjee, V. Kumar, “ Anomaly detection: A survey”, ACM Comput. Surv. 41, 3, Article 15 ,Minnesota, Jul. 2009.