Detecting Anomalies over Message Streams in Railway Communication Systems

Lucas Foulon¹ ², Serge Fenet³, Christophe Rigotti⁴, and Denis Jouvin²

¹ Univ Lyon, CNRS, LIRIS, UMR5205, F-69621, Villeurbanne, France
lucas.foulon@sncf.fr
² SNCF Mobilité, DSI Production Ferroviaire, F-69393, Lyon, France
denis.jouvin@sncf.fr
³ Univ Lyon, Université Claude Bernard Lyon 1, CNRS, LIRIS, UMR5205, F-69621, Villeurbanne, France
serge.fenet@liris.cnrs.fr
⁴ Univ Lyon, INSA-Lyon, CNRS, INRIA, LIRIS, UMR5205, F-69621, Villeurbanne, France
christophe.rigotti@insa-lyon.fr

Abstract. This work focuses on the detection of anomalies in a railway communication system. We present the detection method as well as the architecture of the system supporting it. The method is based on two preliminary stages to collect and aggregate the data. It then combines the use of a multidimensional indexation tree (i.e., iSax tree) to store time series, and the computation of an anomaly detection score (i.e., CFOF score). The information stored in the indexation tree enables a fast estimation of the anomaly score, that can therefore be computed on the fly over the stream of messages in the communication system. We show by mean of experiments that this estimation is close to the exact score. We describe the platform that has been implemented, and we show that it is effective to support abnormal behaviour detection in the real stream of messages within the communication system of the French Railway Company (SNCF).

Keywords: Anomaly Detection · Time Series · Message Stream

1 Introduction

Detecting abnormal behaviours in numerical data streams is an important task, necessary in a wide range of domains. Such behaviours, called anomalies, novelties, or outliers, tend to deviate from the main dynamics as if they were generated by a different underlying mechanism [8]. In the current context of exponentially growing volume of numerical data coming from a wide panel of applications, anomaly detection represents a challenge of ever-increasing importance.

The French Railway Company (SNCF) produces and processes in real time, in its information system, a large amount of heterogeneous data. It operates the national rail traffic, including the high-speed rail network (TGV), and its functions include railway services for passengers and freight, as well as railway
maintenance and signalling. Some data, like for example real time passenger information including upcoming departures, disturbances, etc., originate from ground local information systems. Others, like geolocation data, remote administration, train state, etc., are generated on board, produced by the so-called communicating trains. All these data are transferred throughout the main information system by messages, generating traces handled mainly using the ELK Stack \[1\], an open source software platform including the Elasticsearch, Logstash and Kibana tools\[5\].

In the industrial context of the SNCF company, the aim of this work is to detect abnormal behaviours in the stream of message traces. The main contribution of this paper is twofold. Firstly, it introduces the detection method based on aggregation, indexation and scoring to assess the current state of the system with respect the past states. Secondly, it presents the whole platform based on ELK Stack and supporting the method. The software has been implemented, and tested on real data. These experiments show that the approach is effective and efficient to detect abnormal behaviours in the communication system of SNCF. In addition, when compared to the anomalies detected by the Machine Learning toolbox of ELK Stack, it detects more relevant anomalies and provides much more information about anomaly duration.

For the technical part, we use iSAX trees \[14, 15\] that are very efficient multidimensional indexing structures for time series. They enable fast similarity-based searches and have been shown to remain operational even if the number of indexed time series is greater than one billion. In order to compare the last observed time series with the indexed history, we use the CFOF anomaly detection score recently proposed by Angiulli \[4\]. This score is the only one for which the robustness to the curse of dimensionality has been observed experimentally and formally proven. Unfortunately its existing computation methods are not suitable for data stream, and we propose to take advantage of the iSax trees to compute efficiently a close approximation of the CFOF score over this kind of data.

The rest of the paper is organised as follows. The next section presents the related work. The method is presented in Section 3 and the system architecture is described in Section 4. The results are presented and discussed in Section 5, and we conclude with a summary in Section 6.

2 Related Work

New challenges are emerging in the intelligent transport literature, related in particular to the use of data mining and simulation based solutions. With the fast development of connected transport, many axes of research focus on improving the quality of service and on reducing the maintenance cost in this context. Successful application of data mining approaches have been recently reported, as for instance, the detection of defective rail anchors investigated by Khan

---

5 Elastic, Elasticsearch, Logstash and Kibana are trademarks of Elasticsearch BV, registered in the U.S. and in other countries.
et al. [9], the wavelet-based identification of rail surface defects using images presented by Molodova et al. [12], or the detection of illegal pickups using GPS traces reported by Yin et al. [18]. In this paper, we consider the problem of detecting anomalies in a railway communication system. Anomaly detection is a very active research area with many different applications, as for instance the recent work of Abamoff et al. [2] to help diabetic retinopathy diagnosis using convolutional neural networks. There are a multiple methods applied to various domains [7, 6]. The main approaches are quickly recalled hereafter.

2.1 Different methods

There are two main families of anomaly detection methods: supervised and unsupervised. One of the representative approaches among the unsupervised methods is, for instance, the isolation forests used by Ting et al. [16], where data that can be isolated easily (by several decision tree-like structures) are considered as abnormal. Another more recent approach, proposed in [17], is distributed in the widely use ELK software solution. It is based on Bayesian methods to build a statistical model of the system state, and is available in the Machine Learning toolbox of ELK.

For the supervised methods, typical approaches are the one of Mukkamala et al. [13] that uses a support vector machine classifier in order to detect intrusions, and also patterns/rules-based methods as the work by Li et al. [11] to detect objects having abnormal trajectories. The main difference between supervised and unsupervised approaches is that the supervised ones require a training dataset containing objects that are already labelled as normal or abnormal, and in most applications this dataset needs to be large and representative of many of the possible normal and/or abnormal objects that could be encountered. In our case, the data are not labelled and thus the method must be unsupervised. Let us then we focus on the most widely used family of approaches that are the proximity based anomaly detection methods.

2.2 Proximity based methods

These methods have a lot of variants, and can be split in three subfamilies [3]: clustering based, density based, and distance based methods. Clustering based methods assess if an object belongs or not to a cluster. The object is considered abnormal if it is not sufficiently close to any of the clusters [13]. Density based methods include techniques that take into account the density distribution of objects in their representation space like the LOF method proposed by Breunig et al. [5]. Distance based methods are the most prevalent, and simply use the distances between an object and objects in its neighborhood to calculate anomaly score [10].

Recently, Angiulli proposed CFOF [4], a new distance based score to detect anomalies. The main advantage of this score is that it can be applied in high dimensional space, i.e., when object are described by many features. Thus, it could be well adapted to handle sequence of measures (where each measure is
a dimension). However, the existing methods to compute this score [4] are not designed to process streaming data as it is the case in our railway communication system. In this paper we present a method to detect anomalies in such data, combining the CFOF score and multidimensional indexation in iSax trees [14, 15].

3 Method

This method aims to detect abnormal patterns in sequences containing the counts of messages at a crucial node in the information system. Anomaly detection in such sequences of measures is a classical task, that is usually performed by comparing news sequences to a base of sequences of reference and by computing an anomaly score [3]. However, in general, this approach is adapted only for small sequences, since, as shown in [4], when the number of dimensions (i.e., the size of the sequences) increases, the anomaly scores suffer from the concentration phenomenon. This phenomenon is part of the so called curse of dimensionality problem, and limits the ability to distinguish between normal and abnormal sequences. In order to overcome this limitation, in this work, we use the CFOF score, proposed by [4], and that has been proven not to concentrate when dimensionality increases. [4] proposed an efficient technique for calculating the CFOF score by sampling, where the scores of all the objects of a sample are computed with respect to all the other objects of the same sample. This technique takes advantage of a the factorisation of the necessary operations within each sample. The quality of the approximation depends on the size of the sample, and this technique is well suited when one wishes to calculate the scores of all the objects of a database. However, it is not adapted when one wants to compute only the score of a new object against a reference history. So, we propose to take advantage of the properties of the iSAX tree structure [14, 15] to efficiently compute a close approximation of the CFOF score of new sequence arriving in a data stream.

In this section, we first describe how we build the sequences from the raw data gathered from the SNCF information system. Then, we briefly recall the definition of the CFOF score, and present the approximation scheme of this score using an iSAX tree.

3.1 From raw streams to iSAX data

The main component of the information system, called CanalTrain, manages the exchange of all digital information between ground-stations and mobile equipment. When the behaviour of any of the routing component involved in this essential service becomes abnormal, serious malfunctions can not only disrupt this central component, but also indirectly impact other services. The raw stream of data is too detailed and voluminous to be directly analysed. As we are interested in the global health of the communication system, we extract a simplified
description of the dynamics of the stream from the available raw data. This process is described in figure 1.

The raw data flow is steadily gathered from the various collection points implemented in the system, and made available through a Logstash service. The content and semantics of the raw messages are ignored and the message rate is monitored by counting the number of messages per minute. Every $dm$ minutes, an average rate value $M$ is computed. A series of $dk$ consecutive average rate values $M_1, M_2, \ldots, M_{dk}$ is called a sequence, and corresponds to a time interval $dh = dk \times dm$. Such sequences are generated every $dd$ minutes and can overlap over time. These sequences are the objects that will be inserted in the iSAX tree, and for which the CFOF score will be computed. In the results presented Section 5, the sequences are built with $dm = 15$ min, $dk = 24$ (and thus $dh = 6$ hours), and $dd = 30$ min.

### 3.2 CFOF score

The CFOF score of an object $q$ in a dataset $R$ is related to the size of the minimum neighbourhood $k_m$ such that $q$ is among the $k_m$ nearest neighbours of at least a fraction $\varrho$ of all the objects of $R$. The score $\text{CFOF}(q)$ is then simply defined as the value of $k_m$ normalised (i.e., divided) by the size of $R$. The only parameter is the threshold $\varrho \in (0, 1)$, that determines the proportion of objects from $R$ that must include $q$ in their neighbourhood. This parameter controls the sensitivity of the score to the number of objects to which $q$ should look alike, and thus the sensitivity of the whole detection process.
3.3 Score approximation

The iSAX tree being an indexing tree, each of its leaves contains a set of similar sequences. This particular tree however ensures that every subset of the multidimensional space is only referenced in a single leaf. This property let us compute and store statistical information related the spatial distribution of the objects within every leaf. Moreover, the index provided by the tree let us restrict the distance that we have to consider when we perform a neighbour search. This second property is particularly interesting when we are looking for the neighbourhood of a sequence: we can efficiently prune entire regions of the space in which we are certain that no pertinent sequences exists.

The CFOF score of any new object $q$ in the stream, is computed relatively to the existing sequences that are already stored in the tree. This requires to determine, for every object $p$ stored in the tree, the value of the rank of $q$ in the neighbourhood of $p$. This rank will be equal to 1 if $q$ is the nearest neighbour of $p$, 2 if $q$ is the second nearest neighbour of $p$, and so on. We call this value $v\text{-}rank_p(q)$. It is quite simple to compute the score $\text{CFOF}(q)$ when the $v\text{-}rank_p(q)$ are known for every objects. Let $\varrho$ be the parameter of the score defined previously, then we can sort the values of the $v\text{-}rank_p(q)$ by increasing order and pick the $\left\lceil \varrho \times |R| \right\rceil$th element ($|R|$ being the number of objects in the tree). The value of this element is the size of the minimal neighbourhood $k_m$ such that $q$ is among the $k_m$ nearest neighbours of at least a fraction $\varrho$ of the objects in $R$.

As a consequence, the efficient computing of $v\text{-}rank_p(q)$ is a key aspect of the implementation of the method. To avoid the costly count, over $R$, of the number of objects $r$ such that $\text{distance}(p,r) \leq \text{distance}(p,q)$, we replace the exact number of objects by an approximation computed using statistical information about the distribution of distances stored in the iSAX tree. To compute the value of $v\text{-}rank_p(q)$, we therefore search the iSAX tree and safely prune the branches that contain only leaves that are too far from $p$ to contain any objects $r$ such that $\text{distance}(p,r) \leq \text{distance}(p,q)$. Then, using the distribution of distances for each visited leaf, we can estimate the number of objects $r$ of $N_{\text{leaf}}$ for which $\text{distance}(p,r) \leq \text{distance}(p,q)$ as follows. For vectors whose components follow a normal distribution, the magnitude of the vector follow a $\chi$ law. This law quickly tends to follow a normal law when the dimension of the vector increases. Let $F_{\mu,\sigma}(x)$ be the cumulative distribution function of the normal distribution with mean $\mu$ and standard deviation $\sigma$. For a leaf $N_{\text{leaf}}$ and an object $p$, let $\hat{\mu}$ and $\hat{\sigma}$ be approximations (computed in a preprocessing step) of the mean and standard deviation of the distances between $p$ and the objects of $N_{\text{leaf}}$. Then, the fraction of the objects of $N_{\text{leaf}}$ that are at a distance from $p$ that is lower than or equal to $\text{distance}(p,q)$ can be approximated by $F_{\hat{\mu},\hat{\sigma}}(\text{distance}(p,q))$.

4 System Architecture

Nearly 5 billion messages are exchanged every day within the SNCF information system. All are handled by the central plateform CanalTrain, and are redistributed between trains and other software components providing additional
services, like for instance the Geomobiles platform to geolocalize trains. The ELK software is used to monitor these messages and to ensure that they transit properly within the information system. This global architecture is illustrated in figure 2. ELK is composed of three tools: Elasticsearch, Logstash, and Kibana. Elasticsearch is used to index and store data in the JavaScript Object Notation (JSON) format. Logstash is a tool used to collect and forward data and events (also named traces) to Elasticsearch – in our case, Logstash creates a trace for each new message arriving in CanalTrain. Kibana, the third tool of ELK suite is a visualization tool used to create dashboards. Elasticsearch is used to build and store a monthly index of all the traces created by Logstash. SNCF supervisors create dashboards to visualize statistics and features (histograms, scatter plots, ...) related to these traces. The central indicator is the number of messages received per minute, the measure on which we perform the anomaly detection.

The data preprocessing is structured in three steps: collection, aggregation and indexation. The data collection and aggregation steps are performed using requests to Elasticsearch, that return the result of the queries in JSON format, including the number of messages received per minute. Afterward, data sequences are constructed by aggregating the measurements of the number of messages. The last preprocessing step consists in indexing these sequences in an iSAX tree, each of them representing the history of objects of reference for the anomaly detection.

For every new object, an approximation of the CFOF anomaly score is computed with the method described in Section 3, and stored in a specific index of Elasticsearch, in order to be available in Kibana for visualization.
5 Results

We present in this section the results of the evaluation of our anomaly detection method in a real industrial context of the SNCF communication infrastructure. The indicator that is monitored is the overall number of messages per minute.

To compute the sequences of reference (i.e., the objects stored in the \( iSAX \) tree) as defined in section 3, we use \( dk = 24 \) measures where each measure is the average over \( dm = 15 \) minutes of the number of messages per minute. Thus, each sequence represents 6 hours (24 * 15 minutes), which is consistent with the time scale of the railway activity regularities. One such sequence was built every 30 minutes (i.e., sequence shift \( dd = 30 \) minutes).

The same setting is used to build the new sequences for which the CFOF score is computed.

We defined a period of reference starting the 1st of November 2017 and ending the 31st of August 2018, and constructed a base of sequences of reference with the method described above. It let us obtain 14592 overlapping sequences. These sequences were extracted by requests to the Elasticsearch engine, and indexed into the \( iSAX \) tree.

We defined the period of test from the 1st of September 2018 to the 25th of September 2018. The measures over this period are shown in Figure 3 (top).

5.1 Anomaly score

The first test sequence for the period starts at midnight on the 1st of September and finishes at 6:00 AM on the same date. The first CFOF anomaly score is computed for this sequence ending at 6:00 AM, and then for all the next sequences (one every 30 minutes). The rho parameter was set to \( \rho = 0.001 \) and the resulting scores are given Figure 3 (bottom). On this figure, we can observe four periods of disturbance, that are pointed out by a higher CFOF score. The first occurs from the 6th to 7th of September, the second from the night between the 13th and 14th to the 15th of September, the third during the 19th of September, and the last spans from the evening of the 20th to the 22nd of September. The analysis of these periods of disturbance is performed Section 5.2.

Figure 3 (bottom) shows the fast estimation of the CFOF score made by our method using the \( iSAX \) tree (in orange on the figure), as well as the exact CFOF score (in blue on the figure). One can observe that the approximation closely follows the exact score. The computation of the approximation of the CFOF score took less than 2 minutes per sequence, on a standard desktop computer (using a single core on a 3.40GHz PC with 4GB of main memory), and was 25 times faster than the computation of the exact score.

The Spearman rank-order correlation confirm the quality of the approximation. Its value, calculated between the real and estimated CFOF scores during the period reported in Figure 3, is 0.788. Computing the CFOF score with other \( \rho \) values shows that when rho increases, the Spearman rank-order correlation
Detecting anomalies over message streams in communication systems

Fig. 3. Number of messages received per minute during the period from the 1st of September 2018 to the 25th of September 2018 (top). CFOF anomaly score over this period with $\varrho = 0.001$ (bottom).

is even better and increases monotonically. We observe for instance a value of 0.886 for $\varrho = 0.01$ and up to 0.998 for $\varrho = 0.1$.

To study the impact of the parameter $\varrho$ on the detection, we consider two extreme values of $\varrho$. For $\varrho = 0.1$, the CFOF value of a sequence is computed by taking into account its rank in a neighbourhood of 10% of the sequences of reference. This leads to higher scores, that are based on a very stringent similarity requirement. The second value of $\varrho = 0.001$ takes into account the rank in a neighbourhood of 0.1% of the sequences of reference. This is a much weaker requirement, leading to lower scores. Some anomalies are not detected for all values of $\varrho$. For instance, in Figure 4, with $\varrho = 0.1$, we can observe two small anomalies the 17th and the 18th of September, that are not captured with $\varrho = 0.001$. However, the most salient anomalies are detect for all $\varrho$ in $[0.001; 0.1]$, as for example the one occurring on the 19th of September (see Figure 5 (bottom) for other intermediate values of $\varrho$). In the case of an automatically-controlled detection process, the last step of the anomaly detection can be performed by setting a threshold above which the score generates an alarm message. In this context the tuning of $\varrho$ can be made by increasing/decreasing $\varrho$ to be more/less tolerant to variations of the shape of the signal with respect to the sequences of reference. In our case, we use dashboards inspected by experts. Thus, it is not necessary to select a single value of $\varrho$ since the CFOF anomaly scores can be plotted for several values of $\varrho$ on the same graph, and presented to the experts, as for instance in Figure 5 (bottom).

5.2 Description of the anomalies

In this subsection, we describe the four periods of disturbance observed in figure 3. Within the SNCF information system, two main classes of known anomalies are related to messages buffering and signal collapse. The first kind of anomaly is caused by the near saturation at a given point in the processing and communication chain, entailing the buffering and accumulation of messages in a node of the information system. This creates oscillations in the signal as the buffer is emptied/filled. Such anomalies occur when a node reaches capacity limits or when a computing resource becomes too scarce. When this load bursts
have been processed and corrective actions have been implemented, the traffic returns back to a normal state. The second class of anomalies occurs when the processing and communication pipeline becomes completely clogged at some point. This situation requires most of the time a partial or complete restart of the impacted equipment.

The first anomaly, from the 6th to the 7th of September, happened because of messages accumulated on an upstream platform. The period of disturbance, from the night between the 13th and the 14th to the 15th of September has not been detected by neither the SNCF supervisors nor the different SNCF services. Our method however detected this anomaly, which could be interpreted as the premises of the next one arising the 19th of September. The last anomaly, where the signal collapses during the night between the 20th and the 21st of September, results from an internal dysfunction of a message broker (Apache ActiveMQ). These two last anomalies (19th and 20th-21st of September) are typical representatives of the two classes of known anomalies aforementioned (i.e., messages buffering and signal collapse).

5.3 Comparison with ELK Machine Learning method

In a second step of validation of the method, we compared the results with the anomaly detection method proposed by the ELK Machine Learning toolbox (ELK-ML) on the period ranging from the 1st of September to the 25th of September. We show on the figure 5 this comparison from the 13th to the 24th of September. The raw signal (the number of messages received per minute) is given Figure 5 (top) with the lower and upper bounds computed by the ELK-ML method. If the signal value stays between these two thresholds, then ELK-ML considers it as normal. Otherwise ELK-ML provides an anomaly score greater than zero as shown Figure 5 (middle).

When comparing these values with our anomaly scores given Figure 5 (bottom), it turns out that the anomaly on the 22nd of September is captured by our method but not by ELK-ML. For the other anomalies, both methods are able to underline the beginning of the anomaly. For ELK-ML the anomaly however
quickly vanishes (with an anomaly score dropping back to zero), whereas our method provides additional information about the anomaly duration.

6 Conclusion

In this paper, we focused on the problem of anomaly detection in the message streams of a railway communication system. We presented a method dedicated to this task, as well as a software infrastructure supporting it. This method is based on the collection and aggregation of message counts, the indexation of reference data, and the scoring of the new data arriving in the stream with respect to the reference data. The score used is the CFOF anomaly score, and the method proposes its fast and efficient approximation using data indexing in an iSAX tree.

We presented experiments in a real industrial context, showing that the method is effective in detecting relevant anomalies. When compared to an approach of reference for this kind of data (the ELK Machine Learning toolbox), one of the main advantage of the method relies in the information it provides about the anomaly duration. The approximation of the score performed by the method is very close to the real CFOF score and is computed 25 times faster, enabling the scoring of the anomalies to be performed on the fly. Future work includes the application of the method to other indicators in the SNCF communication system, such as the variance of the message rate, or the average latency between two points of collect. Another promising direction is to use the context of the messages - as for example the type of the data or the type of the source - to define and compute context-dependant anomaly scores.
Acknowledgement

This work was performed within the framework of the LABEX IMU (ANR-10-LABX-0088) of Université de Lyon, within the program Investissements d’Avenir (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR). It was funded by the French National Association of Research and Technology (ANRT). We gratefully acknowledge strong support from the CNRS/IN2P3 Computing Center (Lyon/Villeurbanne - France), for providing a significant amount of the computing resources needed for this work.

References

1. ELK Stack Homepage. 2019-06-12, https://www.elastic.co/fr/elk-stack