Real-Time Anomaly detection for the Secure Water Treatment

No Author Given

No Institute Given

Abstract. Anomaly detection refers to identifying the pat- terns in data that deviate from expected behavior. These non- conforming patterns are often termed as outliers, malwares, anomalies or exceptions in different application domains. The anomaly dedection is done by performing critical comparative analysis using existing approaches, such as SVM, Local Outlier Factor, isolation forest, DBSCAN algorithms. In this report, we had provided information about the Real-Time Anomaly detection for the Secure Water Treatment data-set and the system architecture of the implementation.

Keywords: Anomaly detection · System architecture · Stream Mining.

1 Introduction

Real-time anomaly detection aims to capture abnormalities in system behavior in real time. The abnormalities/anomalies appear in the form of malicious network intrusion, malware infection, over-utilization of system resources due to design defects, etc. Machine learning algorithms have been incorporated into various anomaly detection application areas, such as net- work monitoring, power distribution, IoT sensors devices, healthcare, fraud detection, and transportation traffic anomaly. As such, this paper addresses Real-Time Anomaly detection for the Secure Water Treatment data-set and the system architecture of the implementation. . Accuracy will be improved by hyper-parameter tuning unsupervised machine learning algorithms, which is capable of analyzing data in real-time. Furthermore, data visualization will be presented via Grafa.

The rest of this paper is organized as follows. Section 2 critically overall system architecture and anomaly detection techniques in real time using open source technologies. Section 3 presents experimental setup, data set, features selection, parameter, and problem analysis by evaluating the accuracy rate, performance evaluation, results, and analysis for current anomaly detection approaches. Section 5 evaluates and compares the result with the existing anomaly detection approach. Finally, Section 6 provides concluding remarks.

2 Overall system architecture

This section critically analyzes various clustering algorithms used for real-time anomaly detection. Unsupervised algorithms focus on the data that do not con-

tain any labeling information, and it does not need a separate training and testing phase. Many unsupervised machine learning algorithms have been used for anomaly detection. Unquestionably, clustering helps to classify the patterns into groups and further supports to obtain insight, classification, and compressing of data.

Additionally, this section will show overall system architecture and importance of open source technologies.

2.1 Open source technology and System architecture

The frameworks we are about to use are Kafka, Spark, Influx DB, and docker to communicate. Kafka provides guaranteed message delivery with proper ordering. This means messages published by a producer to a particular topic partition will be delivered in the order they are sent, and a consumer can subscribe to topics and receive messages. Moreover, Kafka can form a cluster to handle a high volume of data with low processing latency and server failures without losing messages.

As the stream data comes continuously and is voluminous, we require a scalable distributed framework. To address the scalability issue, we can use a distributed solution like Hadoop, MapReduce, etc. Hadoop runs in batch mode and cannot handle real-time data. As we are looking for real-time data analysis in a distributed framework, Thus we can use Apache Spark which is fault-tolerant and supports distributed real-time computation systems for processing fast, large streams of data.

Influx-DB is used as a data store for any use case involving large amounts of time-stamped data, it works smoothly with Spark-structured streaming to process, store and visualize data in real-time. Data is transported through a Kafka queue continuously as a stream. It is then split into micro batches (DStreams).

The real-time framework applies machine learning algorithms to analyze stream data. It uses unsupervised clustering algorithms (e.g., k-means, DBSCAN) to build clusters on training data. And after clustering, it predicts data using its cluster information

2.2 Tools

As the stream data comes continuously and is voluminous, we require a scalable distributed framework. To address the scalability issue, we can use a distributed solution like Hadoop, MapReduce, etc. Hadoop runs in batch-mode and cannot handle real-time data. As we are looking for real-time data analysis in a distributed framework, we have decided to use Apache Spark which is fault-tolerant, and supports distributed real-time computation systems for processing fast, large streams of data. To stream the input data, we have used Kafka which provides guaranteed message delivery with proper ordering. This means messages sent by a producer to a particular topic partition will be delivered in the order they are sent, and a consumer will see messages in the order they are stored. Moreover, Kafka can form a cluster to handle a high volume of data with

low processing latency and server failures without losing messages. To store the time series and the detected anomalies, we use Influx-DB as a data store for any use case involving large amounts of time-stamped data, it works smoothly with Spark-structured streaming to process, store and visualize data in real time. Moreover, we implemented a monitoring unit which uses Telegraf, a plug-in server agent for collecting and reporting metrics to listen to and collect metrics from processes and store them in Influx DB, these are further visualized in a dashboard in Chronograf, a visualization solution for Influx DB 1.x.

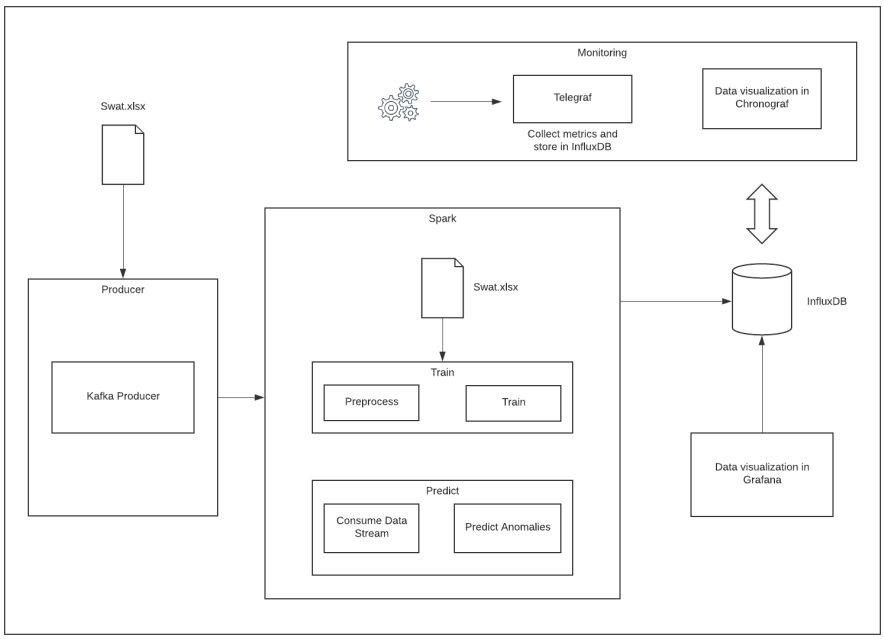

Fig. 1. System architecture

3 Experimental study : problem analysis

In this experiment, the anomaly detection of real-time analytics for the secure water treatment have been evaluated by using four different types of algorithms such as DBSCAN, isolation forest, One class SVM, and KMeans. The following subsections will further provide details of the experiment setup.

3.1 Algorithms

To conduct our experiment, we selected four different frequently used anomaly detection algorithms, especially DBSCAN, isolation forest, One class SVM, and Kmeans (refer to Section 2 for explanation and related work on these algorithms).

3.2 Dataset description

The dataset consists of 13661 samples with 51 attributes which were collected for 6 continuous days. All the data was logged continuously once every second for 136 hours into a Historian server. Data recorded in the Historian was obtained from the sensors and actuators of the testbed. The dataset describes the physical properties of the testbed in operational mode. We infer that the dataset consists of 21 int type, 21 float type, and 3 object type attributes, where 3 object type attributes are converted to float.

3.3 Data preprocessing

The columns with one specific value are checked in the dataset and we find there are 12 attributes with one value. This type of attribute doesn't affect our final outcome. Moreover, it is good to reduce the dimensionality for unsupervised learning techniques. The attributes namely AIT401, P102, P202, P204, P206, P302, P402, P403, P404, P502, P601, P603 are dropped as they are the attributes with one specific value.

We checked with the categorical variables in the dataset and found that there are 15 categorical attributes and the rest are numerical data. Thus standardization takes place over numerical data then Principal Component Analysis(PCA) introduced to reduce the dimensionality of the dataset and on other hand one-hot encoding is applied to categorical data. Finally the preprocessed numerical and categorical data are combined for further analysis.

4 Anomaly detection with algorithms

We apply 4 different algorithms for anomaly detection and also hyper-parameter tuning is used for developing of accuracy.

4.1 Isolotion Tree

The main idea behind Isolotion tree algorithm is, There is the tendency that anomaly points in dataset are easier to be seperated (isoloted) from the non-anomaly points. In order to that, algorithm generates random sample by selecting attribute randomly and then again randomly select a split value for column which must be in range of minimum and maximum range of values.

In order to build an isolotion tree, the algorithm recursively divides by randomly selecting an attribute q and a split value p, until either the nodes has only one instance or all data in node share same value.

After Isolotion tree is applied data, new streaming data is predicted by Logistic regression.Logistic regression is probability based classification algorithm. It assigns each data object value between 0 and 1. Based in these values classes are chosen.

4.2 Isolotion Tree

The local outlier factor is based on a concept of a local density, where locality is given by k nearest neighbors, whose distance is used to estimate the density. By comparing the local density of an object to the local densities of its neighbors, one can identify regions of similar density, and points that have a substantially lower density than their neighbors. These are considered to be outliers.

The local density is estimated by the typical distance at which a point can be "reached" from its neighbors. The definition of "reachability distance" used in LOF is an additional measure to produce more stable results within clusters. The "reachability distance" used by LOF has some subtle details that are often found incorrect in secondary sources, e.g., in the textbook of Ethem Alpaydin

4.3 DBSCAN

DBSCAN stands for Density-Based Spatial Clustering of Applications with Noise. groups 'densely grouped' data points into a single cluster. It can recognize clusters in large datasets by looking at the local density of the data points. The most important feature of DBSCAN clustering is that it is robust to outliers.So it can be used to detect anomalies..

DBSCAN requires only two parameters: epsilon and minimum samples. Epsilon is the radius to be created around each data point to check the density and minimum samples is the minimum number of data points required inside that circle for that data point to be classified as a Core point.

Hyper parameters are tuned in DBSCAN algorithm. Best results obtained when radius is 5 and minimum samples are 5.

4.4 Kmeans

K-means is one of the most popular unsupervised machine learning algorithm thanks to its simplicity and effetviness. K-means is an iterative algorithm that tries to split the dataset into K pre-defined distinct non-overlapping clusters where each data point belongs to only one group. Altohugt it's not widely used for anomaly detection it can be modified in such way that it can find anomalies. The main idea anomaly detector K means is that we set threshold. So this threshold controls distance between point and closest cluster. If the distance between cluster and point is more than the value of the threshold then it won't append to any cluster so it will be classified as anomaly.

4.5 One Class Support Vector Machine

Support-vector machines (SVMs, also support-vector networks[1]) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. There are several types of Support Vector Machines and one of them, namely, One Class Support vector machines are used for detecting anomalies. It creates some kind of circle around non-anomaly points. If one points doesn't fit into that circle then it's classified as anomaly point. One Class Support Vector Machine is one the most used Anomaly detection algorithm and it gave better results in project too.

5 Results and Dashboards

5.1 Results

So out of all algorithms mentioned above K-Mean gave the worst results where One-Class Svm on the other were best.

We used for grafana for dashboard. In time series graph to represent anomalies we changed value to zero and max value in that row when it's not anomaly. In pipeline 1 (one-class svm) dashboard, we visualized the histograms in different days and total number of anomalies. In the second pipeline (iso-log) we visualized the number of anomalies and how they occur in state change graph. Additionaly we compared the results of algorithms obtained by Kmeans, Iso-Log and One Class SVM. That dashboard is also presented below. Chronograph was used to monitor the available memory and GPU and its dashboard is also in the dashboard section.

5.2 Dashboards

References

1. Ariyaluran Habeeb RA, Nasaruddin F, Gani A, et al. Clustering-based real-time anomaly detection—A breakthrough in big data technologies. Trans Emerging Tel Tech. 2019;e3647.
2. M. Solaimani, M. Iftekhar, L. Khan, B. Thuraisingham and J. B. Ingram, "Spark-based anomaly detection over multi-source VMware performance data in real-time," 2014 IEEE Symposium on Computational Intelligence in Cyber Security (CICS), 2014, pp. 1-8, doi: 10.1109/CICYBS.2014.7013369.
3. S. Zhao, M. Chandrashekar, Y. Lee and D. Medhi, "Real-time network anomaly detection system using machine learning," 2015 11th International Conference on the Design of Reliable Communication Networks (DRCN), 2015, pp. 267-270, doi: 10.1109/DRCN.2015.7149025.
4. Ahmad S, Lavin A, Purdy S, Agha Z. Unsupervised real-time anomaly detection for streaming data. Neurocomputing. 2017;262:134-147.
5. Ding Z, Fei M. An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. IFAC Proc Vol. 2013;46(20):12-17.

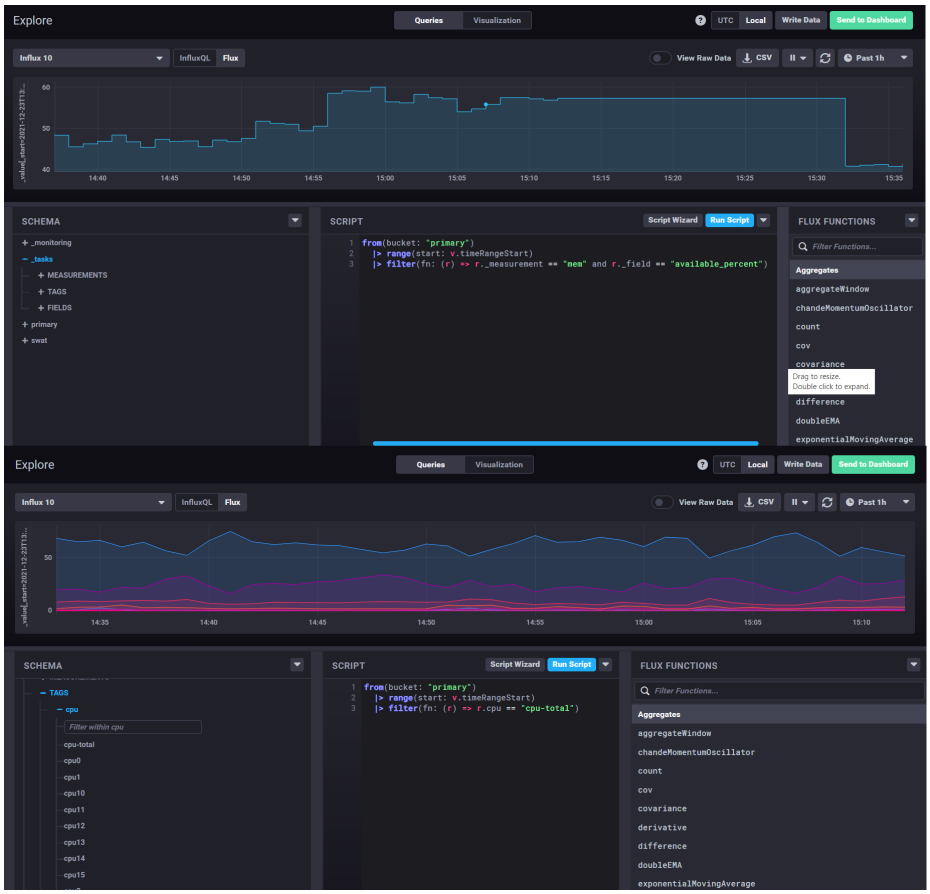

Fig. 2. Dashboard on available memory percentage and total cpu usage

Fig. 3. Iso-Log pipeline Dashboard

Fig. 4. Comparison of methods

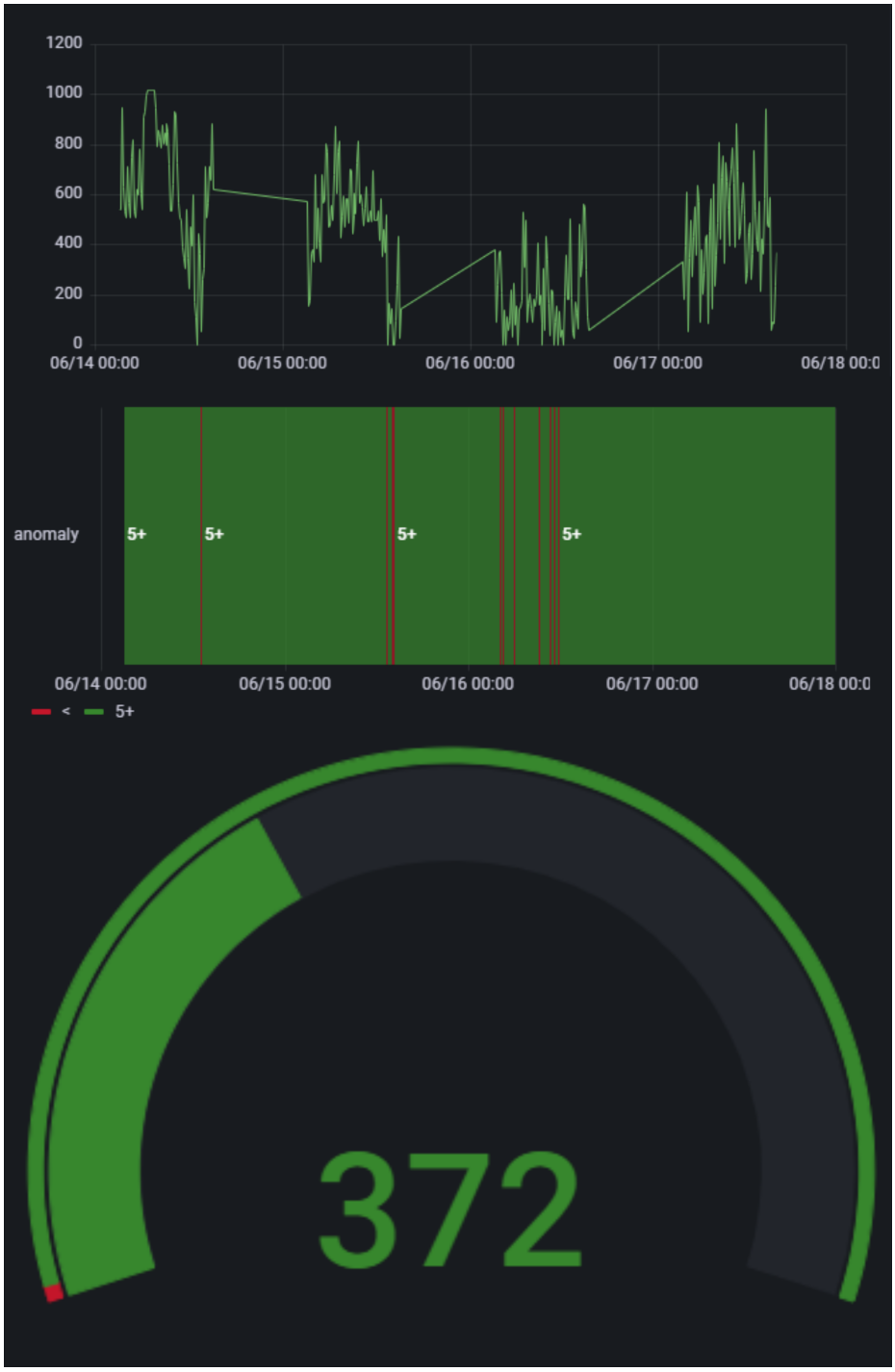

Fig. 5. Iso-Log pipeline Dashboard

6. Rettig L, Khayati M,Cudre-Mauroux P, Pirkowski M. Online anomaly detection over big data streams. Paper presented at: IEEE International Conference on Big Data (Big Data); 2015; Santa Clara, CA.
7. Falcão, Filipe and Zoppi, Tommaso and Barbosa Vieira da Silva, Caio and Santos, Anderson and Fonseca, Baldoino and Ceccarelli, Andrea and Bondavalli, Andrea. (2019). Quantitative comparison of unsupervised anomaly detection algorithms for intrusion detection. 318-327. 10.1145/3297280.3297314.
8. Papastefanopoulos,V.; Linardatos, P.; Kotsiantis, S. Unsupervised Outlier Detection: A Meta-Learning Algorithm Based on Feature Selection. Electronics 2021, 10, 2236. https://doi.org/10.3390/ electronics10182236
9. Alnafessah, Ahmad and Casale, Giuliano. (2020). AI Driven Methodology for Anomaly Detection in Apache Spark Streaming Systems. 1-2. 10.1109/ICCAIS48893.2020.9096667.
10. Solaimani, Mohiuddin and Iftekhar, Mohammed and Khan, Latifur and Thuraisingham, Bhavani. (2015). Statistical technique for online anomaly detection using Spark over heterogeneous data from multi-source VMware performance data. Proceedings - 2014 IEEE International Conference on Big Data, IEEE Big Data 2014. 1086-1094. 10.1109/BigData.2014.7004343.