INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

IJARIIT

ISSN: 2454-132X

Impact factor: 4.295

(Volume 4, Issue 2)

Available online at: www.ijariit.com

Anomaly Detection and Data Classification using KSQL

Radhika Gupta

radhikagupta531@gmail.com

Maharaja Agrasen Institute of Technology, Rohini, Delhi

ABSTRACT

We are going to use KSQL for data analysis. KSQL is an open source streaming SQL engine for Apache Kafka. It allows for identifying patterns or anomalies in real-time data. We will attempt to do classification on a publicly available set of data. We will build models using the training data. We can then compare them by running test data through them. The dataset that is going to be used is the KDDCUP99 Network Intrusion dataset.

Keywords: *KSQL, Anomaly Detection, Decision Tree, Logistic Regression, K-nearest Neighbour, K-means, Network Intrusion Detection, KSQL, Accuracy.*

1. INTRODUCTION

DATA ANALYSIS

Data analysis, or data analytics, is a process of inspecting, transforming, and modeling data with the aim of discovering useful information and supporting decision-making. Data analysis has multiple approaches, it incorporates many diverse techniques under a variety of names, in a variety of domains. In order to perform analysis on this dataset, we are going to be using KSQL.

ANOMALY DETECTION

Anomaly detection is a process of identifying items with the intent of finding the ones that do not conform to an expected pattern or other items in a dataset. The anomalous items will translate to some kind of problem such as bank fraud, medical problems or errors in a text. Anomalies are also referred to as outliers, novelties, deviations, and exception.

KSQL

KSQL is a streaming SQL engine that enables stream processing in Apache Kafka. It makes it easy to read, write, and process the streaming data in real-time using SQL-like semantics [2]. It offers an easy way to express stream processing transformations as an alternative to writing an application in a programming language such as Java or Python [1].

PRINCIPAL COMPONENT ANALYSIS

Principal Component Analysis (PCA) is a dimension-reduction tool that can be used to reduce a large set of variables to a small set that contains most of the information in the large set. It is a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called principal components.

2. DESCRIPTION OF THE DATASET

The dataset that has been used is the Network Intrusion dataset. It is called the kddcup99 dataset [6]. The KDD data set is a well-known benchmark in the research of Intrusion Detection techniques. With the widespread use of computer networks, the number of attacks has grown extensively and many new hacking tools and intrusive methods have appeared [7].

3. RESEARCH MOTIVATION AND RELATED WORK

3.1 In paper 'SQL injection attack and guard technical research', by Xue-Ping-Chen, explains the various vulnerabilities and SQL injection attacks that are prevalent. He has introduced the concept of SOL injection attack and principle, and the realization process of SQL injection attack. Further, its has been described how to detect SQL injection attack, summarizes the general SQL injection attack prevention methods. This paper goes to explain how deeply these kinds of attacks have impacted. It also explains how it has impacted the web security situation. From this, the nature of attacks has been perceived [15].

3.2 In the paper 'A Detailed Analysis of the KDD CUP 99 Data Set', by Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani, provides an in-depth analysis of the kddcup 99 dataset. It explains its significance in the intrusion detection system. It has connections specified and labeled as either normal or an attack. It briefly mentions the role of Network Intrusion Detection Systems (IDS) in the detection of anomalies and attacks. This dataset is raw and we have classified by defining and registering the schema.This has been used as the available dataset on which anomaly detection and models have been built [8].

3.3 In the paper 'Analysis of KDD Dataset Attributes - Class wise for Intrusion Detection', by Preeti Aggarwal, discusses the analysis of the kdd cup dataset with respect to four classes which are Basic, Content, Traffic and Host in which all data attributes can be categorized. It asserts that by doing so this data set is proves to be an improvement over the original dataset from which duplicate instances were removed. This has helped in understanding the nature of the dataset and provided the idea of using pca to reduce the size of the dataset while still keeping the relevant information intact [11].

3.4 In the paper 'A Brief Introduction to Intrusion Detection System', by Ashara Banu Mohamed,Norbik Bashah Idris,Bharanidharan Shanmugam provides an insight to Intrusion Detection Systems (IDS). It talks about the growth of these systems and their contributions in curbing the advancement of network attacks. It further goes to explain how IDS acts a security layer and categorizes the types of attacks into various categories. These categories are dependent on the behavior of the attacks and the pattern in which they carry out the attack. It has helped in providing a better understanding of these systems [3].

3.5 In paper 'Anomaly-based Network Intrusion Detection with unsupervised outlier detection', by Jiong Zhang, Mohammad Zulkernine discusses anomaly detection as a critical issue in Network Intrusion Detection Systems. It states that there are two major intrusion detection techniques: misuse detection and anomaly detection. Misuse detection discovers attacks based on the patterns derived from known intrusions. Anomaly detection identifies attacks based on the deviations from the pattern of normal activities. It discusses the challenges faced in anomaly detection, which is to minimize the occurrence of false positives. This has provided a deeper insight into the understanding of anomaly detection [5].

4. RESEARCH METHODOLOGY

4.1 MODEL FRAMEWORK

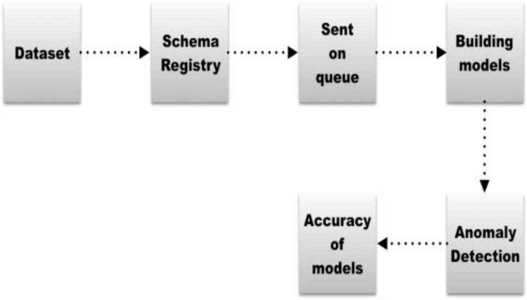

Fig 1

So the dataset was first examined and necessary conversions were carried out to make dataset compatible to be sent to the queue. This is because the dataset is initially available to us as static data and it is not possible to perform any analysis on it [4]. Once that conversion was don't the dataset was classified by defining a schema. This schema was registered for type checking purposes. After schema registry was done, the dataset was sent to the queue. Next models were built on three algorithms - decision tree[13], logistic regression[14] and k-nearest neighbour[12]. For the trained model's anomaly detection rate and accuracy of the models was found out and in the subsequent pages, graphical representations of the figures achieved has been established.

5. EXPERIMENTAL SETUP

5.1 IMPLEMENTATION

The dataset that is available to us is in the csv format. CSV stands for comma separated values. It is not possible to send this static data onto the kafka queue. So in order to make it compatible with kafka, the dataset was converted into stream data [4]. For this, a python script was written and data were converted to the dictionary format. Kafka is not really concerned with type checking, however, this causes problems. Such as, the producer sends a float data type. However when the consumer reads it he interprets it as an integer data type because no type safety has been provided. This results in unnecessary confusion and hassles and ends up taking a lot of time for both parties involved. So in order to resolve and take care of this issue, we define the schema registry. Once the schema was defined, the data was sent by the Kafka Producer onto the queue. This was done by writing a producer python script. Now in order to see that whether the consumer read the data as was intended by the producer, we wrote a python script for the consumer. The consumer script was able to read and display the data in the intended data type format. Hence data was successfully sent by the producer and duly read by the consumer taking into account the type safety. Next, the training models were built on the three algorithms as specified below.

5.2 ALGORITHMS

The following algorithms have been used for anomaly detection:

- Decision Tree Classifier
- Logistic Regression
- K-nearest Neighbour

The following are the codes for the algorithms used:

5.2.1 DECISION TREE CLASSIFIER

- Place the best attribute of the dataset at the root of the tree.
- Next, split the training set into subsets.
- Subsets should be formed in such a way that each subset contains data with the same value for an attribute.
- Repeat step 1 and step 2 on each subset till leaf nodes in all the branches of the tree are found.

5.2.2 LOGISTIC REGRESSION

- Load the data set.
- Next, split the data into training and test dataset.
- Then, use the training dataset to model the logistic regression model.
- Next, calculate the accuracy of the trained model on the training dataset.

5.2.3 K-NEAREST NEIGHBOUR

- First, calculate "$d(x, x_i)$" $i = 1, 2, \ldots.., n$; where d denotes the Euclidean distance between the points.
- Then, arrange the calculated n Euclidean distances in a non-decreasing order.
- Let us assume k to be a +ve integer, subsequently, take the first k distances from this sorted list.
- Find the k-points corresponding to the k-distances.
- Let k_i denote the number of points belonging to the i^{th} class among k points i.e. $k \geq 0$
- Then, if $k_i > k_j \ \forall \ i \neq j$, put x in class i.

6. RESULTS AND IMPLICATIONS

6.1 GRAPHS

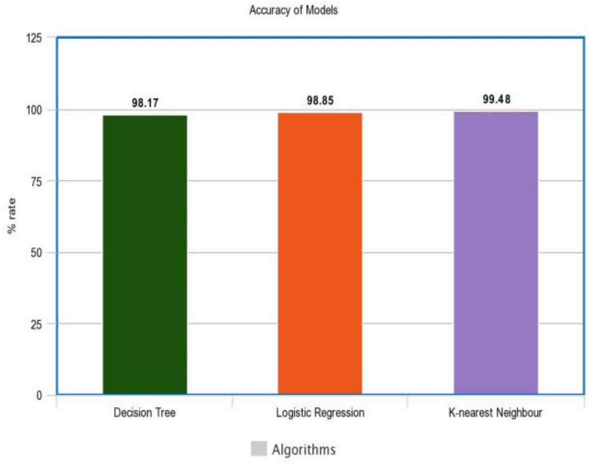

Fig 2 Comparison of the accuracy of models

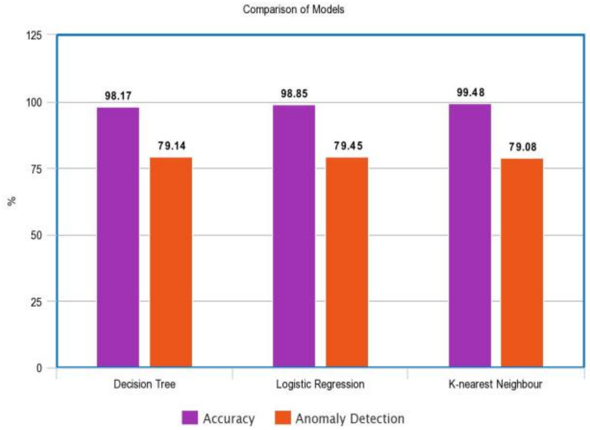

Fig 3 Comparison of accuracy and anomaly detection rate

Fig 2: It is a bar graph that represents the comparison between the accuracy of the three models.
From this, it is clearly established that the k- nearest neighbor model has the highest accuracy rate amongst all three models.
However, the model is slow so logistic regression model is the most accurate.
Fig 3: It is a multiple bar graph that represents the relative comparison of the accuracy and anomaly detection rate of all the three models.
From this, it can be seen that logistic regression model has the highest accuracy rate and anomaly detection rate amongst all three models
Hence, a Logistic Regression model is the best out of all the three models.

7. CONCLUSIONS

We hope to achieve some reasonably accurate models that will perform classification on the data set under consideration. For this, we will leverage the stream processing capabilities of KSQL. We have also established that the model built on Logistic Regression is the one with the best accuracy and anomaly detection rate.

8. ACKNOWLEDGEMENT

I take this opportunity to express my sincere thanks and deep gratitude to all those people who extended their wholehearted cooperation and have helped us in completing this project successfully. I would also like to thank MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY for providing me the opportunity to do a major project on my desired interest. I also would like to acknowledge our gratitude to other faculty members who are not only the source of inspiration but a constant motivation. Despite all efforts, I have no doubt that error and obscurities remain that seen to afflict all research projects and for which we are culpable. I would also like to thank all the faculty members of Information Technology department for their effort of constant cooperation of and, which have been a significant factor in the accomplishment of my major project report. Last but not the least, I would like to place a word of appreciation on record for all those who directly or indirectly supported me.

9. REFERENCES

[1]Alex Handy,'Confluent Brings SQL Querying to Kafka Streaming Data',available on: Https://thenewstack.io/confluent-brings-sql-querying-kafka-streaming-data/
[2]Apache Kafka,'Kafka Streams',available on :https: //kafka. Apache.org /documentation/streams/
[3]Ashara Banu Mohamed,Norbik Bashah Idris,'A Brief Introduction to Intrusion Detection System', available on:https://link.springer.com/chapter/10.1007/978-3-642-35197-6_29
[4]Jay Kreps,'Introducing Kafka Streams: Stream Processing Made Simple',available on:https://www.confluent.io/blog/introducing-kafka-streams-stream-processing-made-simple/
[5]Jiong Zhang,Mohammad Zulkernine,'Anomaly based Network Intrusion Detection with unsupervisedoutlierdetection',availableon:https://pdfs.semanticscholar.org/14d2/4108c42425281db7bc0bac57910ef60f7110.pdf
[6]KDD Cup 1999. Available on: http://kdd.ics.uci.edu/databases/kddcup 99/kddcup99.html, October 2007.
[7]KDD'99 datasets, The UCI KDD Archive, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, Irvine, CA, USA, 1999.
[8]Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani,'A Detailed Analysis of the KDD CUP 99 Data Set',available on:https://www.ee.ryerson.ca/~bagheri/papers/cisda.pdf
[9]Neha Narkhede,'Introducing KSQL: Open Source Streaming SQL for Apache Kafka',available on:https://www.confluent.io/blog/ksql-open-source-streaming-sql-for-apache-kafka
[10]Neil Avery,'Secure Stream Processing with Apache Kafka, Confluent Platform and KSQL',available on:https://www.confluent.io/product/ksql/
[11]Preeti Aggarwal,'Analysis of KDD Dataset Attributes',available on: https://ac.els-cdn.com/S1877050915020190/1-s2.0-S1877050915020190-main.pdf?_tid=873116ce-c82b-46a1-b358-7be0eb7c61b9&acdnat=1522052494_a14e4568eff3773e7396754099988339

[12]Rahul-Saxena,’K-NEARESTNEIGHBORALGORITHMIMPLEMENTATION’,available on:https://dataaspirant.com/2016/12/27/k-nearest-neighbor-algorithm-implementation-python-scratch/

[13]Rahul Saxena,’BUILDING DECISION TREE ALGORITHM’,available on:https://dataaspirant.com/2017/02/01/decision-tree-algorithm-python-with-scikit-learn/

[14]Saimadhu Polamuri,’Implementation of Logistic Regression’, available on:http://dataaspirant.com/2017/04/15/implement-logistic-regression-model-python/

[15]XuePing-Chen,’SQL injection attack and guard technical research’, available on:https://ac.els-cdn.com/S1877705811022764/1-s2.0-S1877705811022764-main.pdf?_tid=806e5620-9eb5-4e46-b146-f032e16813fc&acdnat=1522051350_9bfd1ca231eb73cccef20bf38ecc19b4.