Security analytics of large scale streaming data

Lighari, Sheeraz Niaz

DOI (link to publication from Publisher):
10.5278/vbn.phd.eng.00047

Publication date:
2018

Document Version
Publisher's PDF, also known as Version of record

SECURITY ANALYTICS OF LARGE SCALE STREAMING DATA

BY
SHEERAZ NIAZ LIGHARI

DISSERTATION SUBMITTED 2018

AALBORG UNIVERSITY
DENMARK

SECURITY ANALYTICS OF LARGE SCALE STREAMING DATA

by

Sheeraz Niaz Lighari

AALBORG UNIVERSITY
DENMARK

Dissertation submitted

.

Dissertation submitted: June, 2016

PhD supervisor: Associate Prof. Dil Muhammad Akbar Hussain,
Aalborg University

PhD committee: Associate professor Daniel Ortiz-Arroyo (chairman)
Aalborg University

Professor Dr. Engr. Syed Hyder Abbas Musavi

Associate professor Sadiq Ali Khan
University of Karachi Pakistan (UoK)

PhD Series: Faculty of Engineering and Science, Aalborg University

Department: Department of Energy Technology

ISSN (online): 2446-1636
ISBN (online): 978-87-7210-220-7

Published by:
Aalborg University Press
Langagervej 2
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

Printed in Denmark by Rosendahls, 2018

ABSTRACT

This thesis, mainly involves the security investigation of large scale streaming data. To set up the ground, we began the examination of batch data utilizing KDDcup99 dataset. The KDDcup99 is widely used dataset by research community for threat detection. The project was started to analyse the KDDcup99 dataset using apache spark as a tool for security investigation, at that point it is examined by utilizing diverse machine learning algorithms like Support vector machine, Logistic regression, Naïve Bayes, Decision trees, Random Forest Tree which are cases of supervised algorithms. The project further utilized KMeans as an algorithm for unsupervised anomaly detection. The basic idea to analyze with various algorithms was to analyze them for finding the best algorithm for security analysis of extensive scale dataset. All these algorithms are incorporated as a part of spark machine learning library. In this project, we have further proposed a novel strategy for security investigation using hybrid model of rule based and clustering algorithm.

In the second part of the project, we investigated the streaming data created from the KDDcup99. A lot of work has been done in anomaly detection to the batch data yet recognizing oddities from streaming data that still remains an accessible issue. In streaming data, the tasks related to find out the anomalies has become challenging with the passage of time because of the dynamic changes in data, which are produced by different methods applied in data streaming infrastructures. During the time spent on anomaly detection, above all else, it was first required to know the method for finding the normal conduct of information and afterward it was anything but difficult to know the dynamic conduct or change in the information. In this unique situation, clustering is an extremely noticeable strategy. The use of clustering technique is exceptionally basic to analyze the static information. Yet in the field of data mining, it is a key issue to analyze the streaming data. Therefore, the main focus of our project is to analyze the streaming data. For that, we are applying streaming form of KMeans clustering algorithm. The algorithm is analyzed both on single and distributed environment.

Besides as a use case to analyze the streaming data. We are exploring the latency value created by the sensors introduced in the Web Servers of Smart Metering Infrastructure. In our examination, Kafka Producer generates the sensor latency value. Kafka Consumer based on the Spark Streaming framework then analyzes the value. Moreover, in this project, we are also using the Gaussian distribution model to perceive the peculiarities in the sensor data. This model is widely used for anomaly detection.

TABLE OF CONTENTS

Table of figures

CHAPTER 1. INTRODUCTION

1.2. MAIN IDEA OF THE PROJECT

There are number of Systems working in Smart Grid environment like Supervisory Control and Data Acquisition, Smart Meters, Sensors, Synchrophasors and Electric Vehicles. The data generated by these systems is very large which comes in the category of the big data. Big data is expressed in terms of Volume, Velocity and Variety (3Vs), where volume says about size of data, normally it starts from terabytes and velocity talks about speed of data generation which is very high in case of big data, Whereas variety expresses different formats of data, e.g: text, signals and images. The big data generated by Smart Grid (SG) systems is facing various attacks like other systems, which are using information and communication infrastructure (ICT). The traditional methods and techniques are not good at securing such a big amount of data generated by SG. It needs some special and advanced techniques. These advanced techniques comes with the field of big data analytics, which becomes become big data security analytics in perspective of security. The security threats can be known or unknown. The known threats can be detected by looking at the signature or pattern of the attacks but it is difficult to detect unknown threats or attacks. The detection of unknown attacks using big data analytics is relatively new and advanced approach towards the big data security analytics.

The big data generated by different SG systems is normally stored at a big data repository located at utility centers. The utility can use the big data for various functions like demand response and energy forecasting and threat detection etc. There are two types of big data available in SG systems, the data at rest and data in motion. The big data analytics is possible at two levels, analytics with the data stored in the repository (data at rest) or data during streaming (data in motion). Different applications have different requirements for example, in energy forecasting data can be useful for analysis after collecting from different sources (data in rest). But for security purposes, it is required to analyze the data at real-time (data in motion). The data like logs and events generated by Smart Grid devices e.g. Smart Meters can be used for security analytics. The Smart Meters reports the energy consumption

every 30 Minutes [1]. Thus a huge number of logs and events generated by every Smart Meter in millions of Smart Meters which is a good source of security analytics at real-time.

The project mainly involves the security analysis of the large scale network data of Smart Grid systems but for setting up the ground, we started the analysis using KDD cup 99 dataset. The KDD cup 99 dataset is widely used data for threat detection by research community. In the project, KDD cup 99 dataset will be fed to spark engine for the security analytics, then it will be analyzed using different machine learning algorithms like Support vector machine, Logistic regression, Naïve Bayes, Decision trees, Random forest which are examples of supervised algorithms and we have used k-means as a unsupervised algorithm. The basic idea to analyze with different algorithm was to compare them for finding the best algorithm for security analysis of large-scale dataset. These algorithms are integrated part of spark machine learning library. In this project, we have also proposed a novel method for security analytics using hybrid model of rule based and clustering algorithm. After the analysis, finally, we visualized the results using different graphs.

The aforementioned analysis is based on the dataset, which is stored in a data repository like disk. It is called as offline or batch analysis. In second part of the project, we generate the stream of KDD cup 99 and then analyze it. This type of analysis is called online or streaming data analysis. We analyze the KDD cup 99 using both supervised and un-supervised algorithms for threat detection.

Then after analyzing the KDD cup 99, we continue our analysis on Smart Metering system. Mainly, it involves the analyses of streaming data generated by the sensors which are installed in the web servers of Smart Metering system to report latency value. The project is further depicted in figure 1.

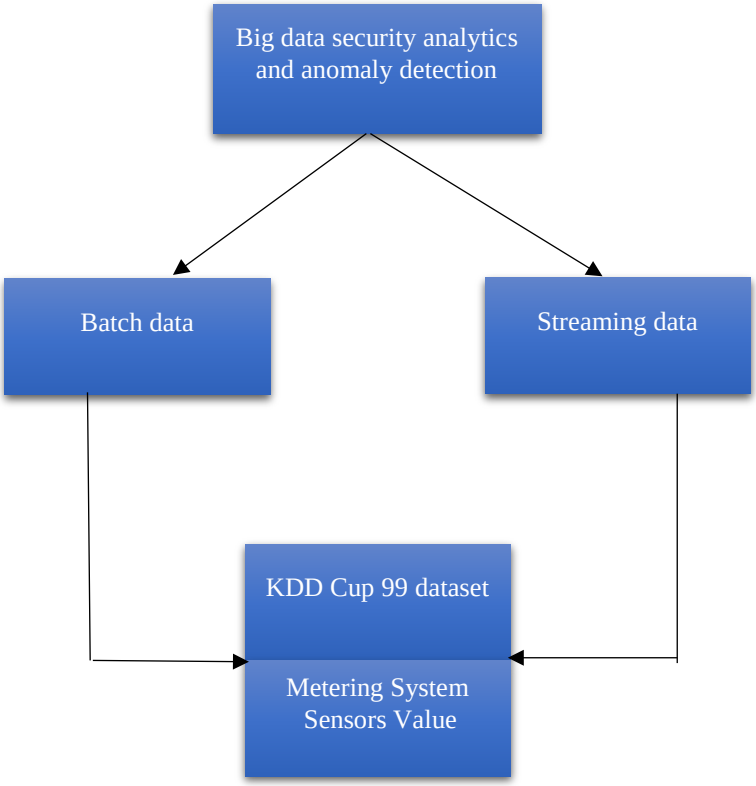

Figure 2 Project description

1.3. MOTIVATION OF THE PROJECT

Nowadays, almost every electric system is modernized with the help of advanced information and communication techniques. This modern system is called as Smart Grid. The Smart Grids collect and report electric consumptions via modern ICT system using Smart Meters. In Smart cities, almost every house is installed with Smart Meter, as a results in every city, they produce very huge amount of data every day. This huge amount of data exceeds the limits of normal size and becomes gigantic, and therefore need special and advanced techniques for analysis. The tools dealing with big data very much fits for Smart Meter analysis. The Smart Meter data can be used for energy forecasting and anomaly detection. Most of the work is related to energy forecasting and very few on anomaly detection, hence we directed our research

focus to anomaly detection. In smart meters system, the data of two types one is actual data which is the consumption data and other is the network data. We are focusing in our project on the network data.

1.4. SCOPE OF THE PROJECT

There are various networks like computer, cloud, telecommunication, social , health and smart grid networks, all they are producing huge amount of data. This huge amount of data needs special consideration for data analysis as ordinary methods are not appropriate for big data analysis. The proposed work can be utilized in any network but mainly its focus is on Smart Grid system. The Smart Grid systems produce very large amount of data which eventually analyzed for different purposes. The analysis can produce the right information which is used for decision making, if the data is not compromised. As in other networks, the traditional techniques and methods are also not working properly in case of Smart Grid systems to secure such a huge and rapid generation of data. Hence, our method of incident detection will act as a final and efficient layer of defense for the big data security of Smart Grid systems.

1.5. AIMS OF THE PROJECT

Main objective of our project is to contribute the research in the field of big data for anomaly detection based on the Smart Grid network data as the use case. For fulfilling the above object, first we started our analysis from the KDD cup 99 dataset; a well-known dataset for intrusion detection. The dataset was analyzed as both batch and streaming data. The project was than geared to the anomaly detection sensor data latency value installed in the web servers collecting the Metering infrastructure data.

1.6. PROJECT IMPLEMENTATION APPROACH

Our project involve both batch and stream analysis. In case of batch processing, we follow below three steps:

1. Loading the dataset
2. Processing or analysis
3. Visualization

1. **Loading the dataset:**
   KDD Cup 99 dataset is widely used dataset for network intrusion detection. It will be loaded to the spark engine for analysis.

2. **Processing and analyzing**
   For security analytics, clustering algorithms can be used. The apache spark has very good machine library for clustering algorithms.

3. **Visualization:**
   Visual reports will be displayed using different statistical graphs.

In case of stream analysis, the data is fed to the spark streaming engine by stream data generator as an integrated tool. In our project, we are using apache kafka to generate the stream of KDDCup99 dataset to set an example of large scale stream data security analytics as depicted in figure 2.

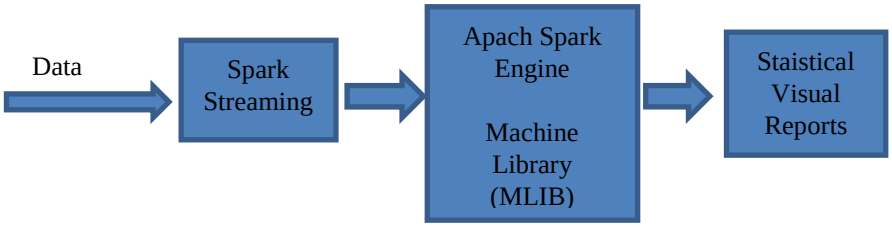

Figure 2. Data Stream analysis

1.7. STATE OF ART

Firstly, I provide some references about the work based on encryption and authentication techniques for the Smart Grid systems. The Public

Key Infrastructure (PKI) is considered as the best key exchange technique for Smart Grid systems [3]. PKI is secure type of key exchange algorithm but it is not only requirement of the Smart Grid system because PKI compromises efficiency with security. Reference [4] presents the identity based encryption for secure authentication. In identity based encryption an identity number is used to generate a unique key which encrypts the packet with a kind of signature. Authors in [5] proposes the scheme of combined integrity and confidentiality. The scheme exhibits the cascading failure of Smart Meters security, if one Smart Meters key is leaked, the neighbors are also compromised. Reference [6] suggests the concept of anonymous readings where every meter has two type of readings, its own reading and reading of other meter. This is a good method to hide the identity of the customer but it makes the extra burden on the communication. Reference [7] presents the authentication and encryption scheme between Smart Meter and home appliances (Home Area Network) which is based on PKI. Reference [8] proposes the attack tree approach for directing the device security test.

After proper authentication and encryption, the second layer of defense is intrusion detection system (IDS). AMI does not have an efficient Intrusion detection system [9]. Basically, there are three main types of Intrusion detections, signature-based, anomaly-based and specification-based [9]. The signature-based IDS checks the signature or pattern of the threat, anomaly-based is designed to the deviation of the system from its original behavior and specification-based IDS looks at the specified constraints. The complete IDS system must be able to detect both known as well as unknown attacks. The IDS uses both signature and anomaly-based techniques can sense both known and unknown attacks. From the limited amount of work done for designing the efficient IDS for Smart Grid, the key requirements for IDS of AMI are discussed in [10]. In [11], the IDS for Home Area Network (HAN) proposed which detects the threat on both signature and anomaly. Reference [12] suggests the specification based IDS which can only detect the known attacks. Reference [9] proposes the model for distributed signature and anomaly-based IDS for Neighborhood Area Network (NAN) of AMI. The design proposed by [9] detects the intrusion based on some rules.

It is very challenging to make an IDS for the systems generating big data (BD). The authors[13] gives survey and propose the Hadoop-based intrusion detection for BD of telecommunication systems. The authors in [14] indicates the shortcomings of the traditional tools and methods to manage and analyze big data due to its size, heterogeneity and speed of generation. It also recommends the Hadoop-based map-reduce for minimizing the size of big data. The authors in [15] presents the security analytics of the logs produces by web server using Hadoop-based technologies. The researchers in [16] [17] [18] [2] recommends the BDA (Big data Analytics) methods and tools as the advanced and latest approach to perform the security analytics. The authors in [19] presents the BDA techniques for analyzing the data generated by Phasor Measuring Units (PMUs) or synchrophasors which reports the measurement of electrical signals. The reference [18] proposes the java Hadoop-based architecture which use mysql for mining the threats. The proposed approach is not suitable for real-time analytics. The authors in [20] also propose the hybrid IDS based on snort and Hadoop which is also made for off-line analytics. The reference [1] recommends the stream processing and real-time analytics for captured data which includes Typical Load Profies (TLPs), logs and events generated by different devices in the complex architecture of the Smart Grid.
The reference [21] provide the anomaly detection based on the context of the data using clustering. The authors in [21] are presenting the real-time analytics on the sensor data like data streaming with temporal information, customer profiles and audio, video and images captured by sensors. The authors in [22] propose the BDA architecture for monitoring the network data of enterprise networks but it performs the off-line analysis.

The comprehensive literature survey confirms the recommendations of BDA techniques for security analytics. Most of the work comprises of the off-line big data analytics. Some work as in [22] gives real-time analytics on sensor data. The analytics on network data is presented in [15][22] which is done using off-line techniques. In order to have the early detection of threats in Smart Grid systems, it is required to analyze the network data at real-time. Therefore, my project is aimed to perform the real-time security analytics on network data which is produced by the sensors installed in the network of metering system.

1.8. RELATED WORK

Most of the anomaly detection problems are solved the KMeans Clustering as we reviewed in the literature. However, some of the problems in KMeans centroids are reported in [23]. The authors in reference [24] analyses the anomaly detection in streaming data. In the reference, anomalies are defined with the method of clustering score.

Mainly the distributed environments are useful to process the large scale datasets. The examples of these environments are Hadoop and Mapreduce [25]. They are primarily designed for the batch systems, and therefore not suitable for the real-time systems. Likewise, HBase[26] and Bashreduce[27] are also suitable for the batch processing. There are some other frameworks like apache Mahout [28], it also provide the support of Kmeans but only for the batch mode. Furthermore, anomaly detection on batch data using Hadoop and Mapreduce is provided in the reference [29] and [30]. Inherently, they are designed for batch mode, hence, they are not appropriate for the streaming mode of processing.

There are different tools available for the real-time processing like apache storm, apache s4, and apache spark. Among these the apache spark outperforms, therefore is the main working tool for research [31]. The anomaly detection technique based on apache spark is referenced in [32]. Most of the work in literature is focusing on the accuracy of the anomaly detection model but in our work, we are also paying attention to the processing of anomaly detection.

Now, we look at some anomaly detection or intrusion detection systems. The authors in reference [33] is explaining an intrusion detection system working on the signature identification and it evolves the rule-based identification. The authors in the reference[34], suggests an intrusion detection system for the cloud networks. The cloud networks are one of the producers of big data. The intrusion detection system in the reference [34], is based on the distributed environment in which systems are co-operating with each other to perform the intrusion detection. Moreover, the authors in the reference [35] performs anomaly detection based on fuzzy means and signature rules.

The authors in the reference [36] studies the kddcup99 dataset using apache spark, their main work is based on the supervised algorithms but in our work, we are studying both supervised and unsupervised algorithms.

The most of the work related to the streaming data involves the networks like computer networks, cloud networks, telecommunication networks, and social networks but less of the work seen on the Smart Grid systems for stream data processing, therefore we are providing the work focusing on the Metering System. In metering system, we are detecting the anomalies in the latency values installed in the web servers which are collecting the consumption information from the meters.

1.9. MAIN CONTRIBUTION OF THE PROJECT

In this project, we contributed both in anomaly detection of large-scale batch data and streaming data. The contributions are listed as following.

3.1. ANOMALY DETECTION OF LARGE SCALE BATCH DATA

Our first contribution in this section of project is about to find out the best performing algorithms for anomaly detection of large scale batch data. For this, we compared different parameters of both supervised and un-supervised algorithms. This part of the thesis is contributed in Paper A.

3.2. ANOMALY DETECTION OF LARGE SCALE STREAMING DATA

In this part of project, we are mainly contributing the anomaly detection of streaming data both on single and distributed mode. The various factors were analyzed in this paper like accuracy rate, anomaly detection time, true positive rate, and false positive rate. The analysis was performed using streaming version of KMeans. This part of the paper is contributed in Paper D.

Furthermore, as a use case for anomaly detection of streaming data, we provide the analysis of sensor data. The sensors are connected to the webservers receiving consumption information from metering system. The sensors are reporting the latency values and we detecting any abnormal latency value affecting access of information from the webservers. This process involves the use of Gaussian distribution model. This part of the paper is contributed in Paper E.

Moreover to aforementioned contributions, we have also provided a survey to setup the monitoring system for Metering system based on the big data analytics. This part of the paper is contributed in Paper C.

The thesis contribution involves the list of following papers:

3.3. LIST OF PAPERS CONTRIBUTED IN THE PROJECT

Paper A: Testing of algorithms for anomaly detection of big data using apache spark

In this paper, we performed the evaluation of algorithms for anomaly detection at large scale batch data. The comparison is done between both supervised and unsupervised algorithms. The supervised algorithms involve Support vector machine, Logistic regression, Naïve Bayes, Decision trees and Random forest. In case of unsupervised algorithm, we performed analysis based on KMeans. The evaluation of algorithms is based on three factors of Accuracy rate, Training time and Prediction time. Moreover, whole process of analysis is performed on the batch data including KddCup99 dataset.

Paper B: Hybrid model of rule-based and clustering analysis for big data security

In this work, we propose the application of rule-based technique after the clustering method for anomaly detection of large-scale batch data. The reason behind the proposal is based on normalizing of big data by the clustering method. Once, it is normalized to small dataset then its analysis is easy and will take less time for anomaly detection.

Paper C: Reviewing the security surveillance of AMI using big data analytics

In this paper, we surveyed the sections of Metering Systems requiring the implications of big data methods and procedures. We also reviewed the possibilities of utilizing different big data tools to monitor the metering system. In this paper, We also

discussed the opportunities and constraints in using the big data technology in the field of Smart Grid.

Paper D: The efficient way of detecting anomalies in the large scale streaming data

In this paper, we are proposing the methods and techniques by which we can detect anomalies efficiently from the large scale streaming data. The technique involve the updating of cluster centres after arrival of new batch in the stream of data. This is applied by using streaming version of KMeans. In order to validate the efficiency of algorithms we have tested it both on single and distributed environment. The parameters used in the evaluation are accuracy rate, anomaly detection time, true positive rate, and false positive rate. Furthermore, in this work, the stream of data produced from KddCup99 dataset

Paper E: Anomaly detection of sensor streaming data based on the Gaussian distribution model

In this paper, we perform the anomaly detection on large scale streaming data. The streaming of data is produced by the sensors, which are installed in the webservers connected to the metering system. The sensors produce the latency values in accessing the data from webservers. In this paper, we analyze any deviation from the normal latency value using the Gaussian distribution model.

1.10. THESIS OUTLINE

Thesis is divided in two parts, the first part involves the introduction, background and conclusion and future work and the part include the list of papers as depicted in the following figure 3.

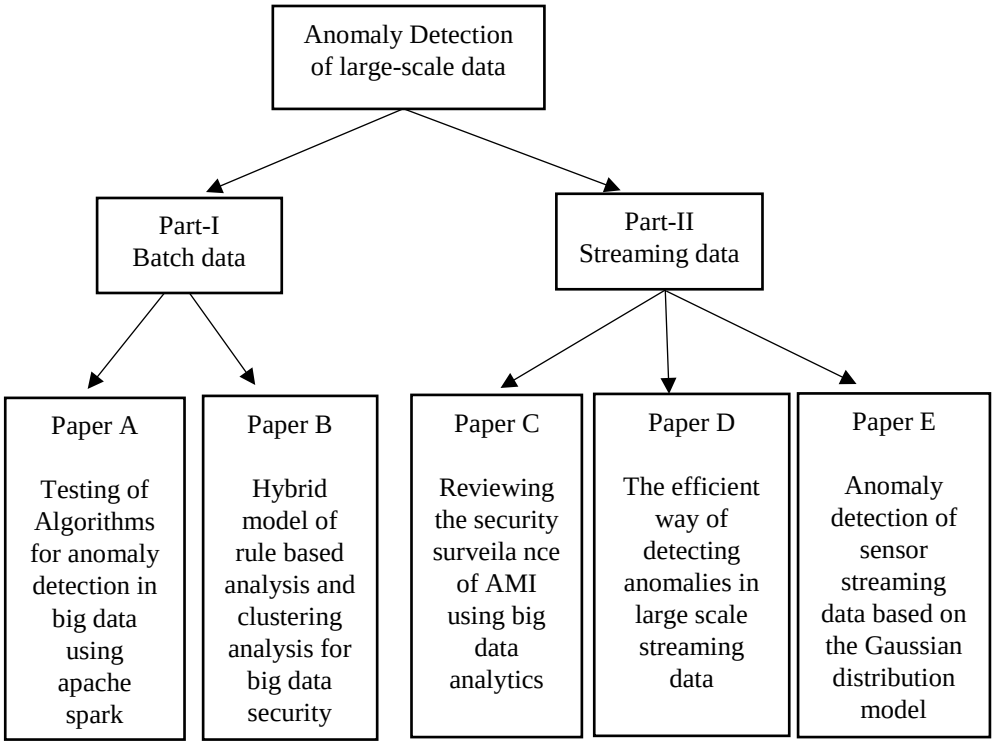

Figure 3. Thesis outline

CHAPTER 2. REAL TIME STREAMING DATA ANOMALY DETECTION FRAMWORK

2.1. BACKGROUND

Online anomaly detection tools [32,37,38,39] purpose is to catch the anomalies in the real-time systems. These anomalies may seem in various shapes like malevolent network intrusions [38, 40], irregular patterns, malicious contagions, and over-utilization of the resources etc. In order to detect anomalies, the real-time systems needs to captivate the stream of data for analysis. Hence, it is very challenging work due to large volume, high velocity, and may be the complexity in the data streams.

It is observed that the real time anomaly detection systems are dire needs of Smart Grid, telecommunication, cloud, and social networks. Real-time anomaly detection systems of these systems handles huge amount of data which is called as Big data. There are two types of big data processing systems; batch processing and stream processing systems. Hadoop[25], MapReduce[27], Hbase[26], Mahout[28], and Google bigtable[41] are inclined to batch processing, whereas apache Spark[42], apache storm[43], and apache s4[44] are the stream processing systems. Among them the apache spark performs the low latency processing because of its feature of in-memory computation. In addition to low latency, it makes the processing units called as mini-batches the smallest as they can be. It also maintains the states of processes to recover them in case of the failure. Spark has the exceptional element of DAG(Directed Acyclic Graph) which has the features of cyclic data flow and in-memory handling, which makes it quicker than other different frameworks. In case of in-memory computing, Spark is 100 times faster than Hadoop and whereas on disk, it ten times faster than Hadoop[42]. Furthermore, in case of stream processing tools, it is also faster than apache storm and apache s4 [45]. Spark also offers the easy configuration for the cluster computing.

Despite of the aforementioned challenges in real-time anomaly detection we have contributed by the following way:

First we devise a method to produce the real-time data using Kafka Producer [46] from Kddcup99 dataset. Than this stream of data is sent to the Kafka Consumer for analysis [46]. The Kafka supports the assured transfer of the messages in appropriate order. Than Kafka producer sent the messages to specific topics which are read by Kafka consumer in the order they are saved in the topics. Furthermore, Kafka cluster can be modeled with the features of low latency in processing and a fault-tolerant system. As a case study, we are producing sensor values reporting the latencies in the Web Servers collecting the information from the Metering system.

Second, we proposed models and its implementations to detect the anomalies from streaming data. The first streaming model that detects the anomalies from the streaming data. The streaming data produced in this case is from the Kddcup99 dataset. This model is, based on Streaming KMeans algorithm. Our second model for anomaly detection of streaming data, is based on the Gaussian distribution model. This model performs anomaly detection on latency values produced by the sensors.

2.2. APACHE SPARK

Apache spark is an open source big data project. Its development started in the year of 2009 in UC Berkeley lab. The reason to initiate the spark project was the inefficient MapReduce jobs and the spark was designed so that it process the algorithms fast. The spark also came with the concept of in-memory data storage and the idea of efficient fault recovery. In conclusion, spark product became more faster than MapReduce jobs. Initially, spark was started to develop in UC Berkeley lab, now it is contributed by some large organizations like Yahoo, Databricks and Intel [47].

Apache spark is a framework based on the cluster-computing concept and written in Scala language. It appeared to be the fast processing engine for large-scale parallel processing. Basically, spark improves the concept of MapReduce model by working in-memory. The in-memory

concept of spark helps him to process the data more fast than the in-disk data processing which was used by the MapReduce model.

The most important libraries of spark project are Spark MLlib, Spark Streaming, Spark SQL and GraphX. All these components can be used on one project. These all components are scheduled, monitored and distributed to spark cluster by spark core as in the figure 3.1 [47]. Spark supports its libraries in many languages like scala, java, python, and R. Spark can be executed over Hadoop clusters, therefore it can also be integrated with other big data tools.

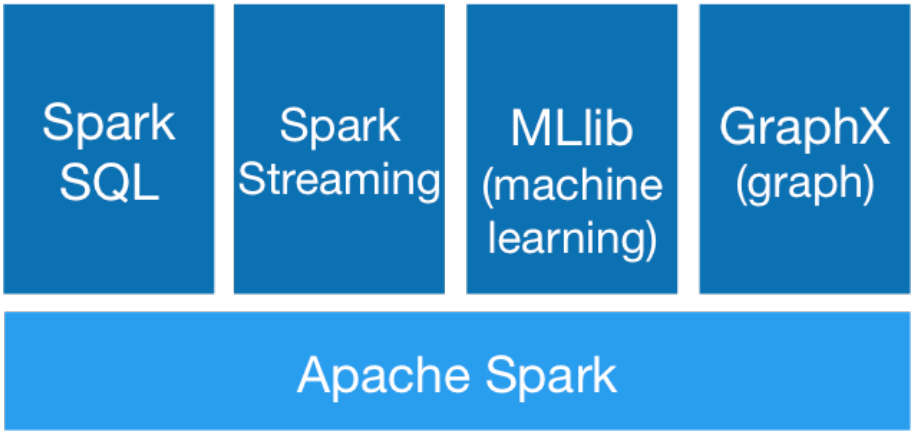

Figure 2.1 Spark framework[48]

2.3. BASIC TERMS AND CONCEPTS IN SPARK

To understand the working of spark we need to be familiar with following terminologies related to spark.

Job: It is a computation, which executes in parallel manner. It takes input from any input source and then process it to produce the results.

Task: Jobs are divided in different stages and stages in tasks. Each task is allotted to one partition which then processed by one executor.

Executor: It is a actual machine process which is executing the task. Its number depends on the time taken by one job. The figure 3.2 shows reduction in processing time with the increase in the number of executors [47].

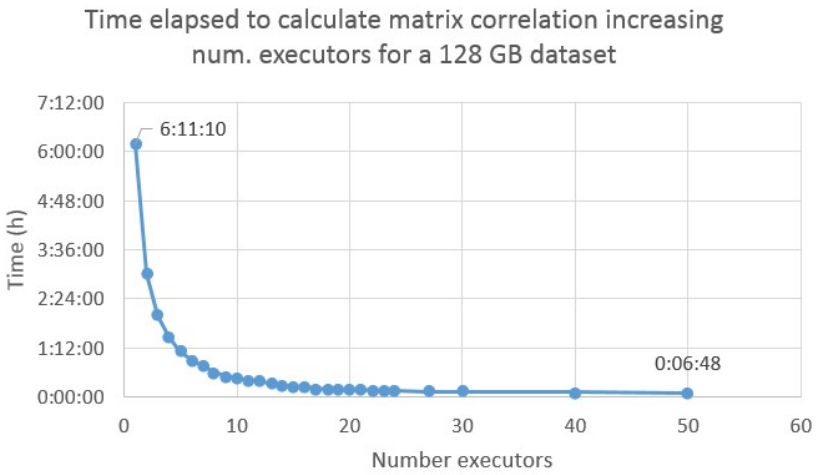

Figure 2.2 Number of executors [47]

Driver: It is a program, which runs the job on spark.

Master: It is a machine which is running the driver program.

Slave/Worker: It is a machine which runs the executor program

The figure 2.3 as in the following shows the overall structure of the spark program.

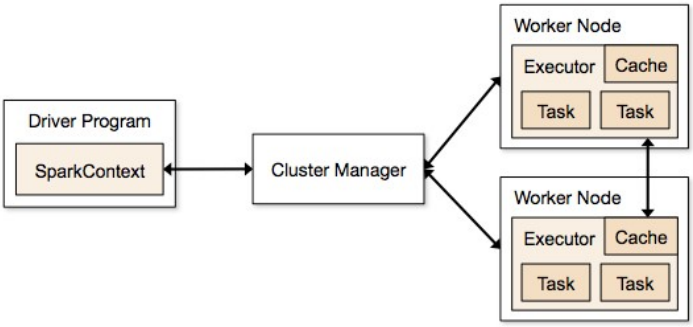

Figure 2.3 Spark Program [47]

2.4. RDD (RESILIENT DISTRIBUTED DATASET)

RDD is a main concept in spark. It is a collection of objects, which is read-only and stored in memory in special case of spark or in disk. There are two methods to create RDDs:

(1)It can be created by loading the external dataset.

(2)It can be created by executing the parallelize method.

One of the main characteristics of RDD is that it is a fault tolerant [49] which means that the lost partition can be easily recovered from the information contained in the RDD graph. An other important property RDD contains which is called the lazy evaluation. It means the operation are executed immediately by spark, rather it waits for other operations to be requested, so that it can group them together. The RDDs perform two types of operations as in figure 2.4.

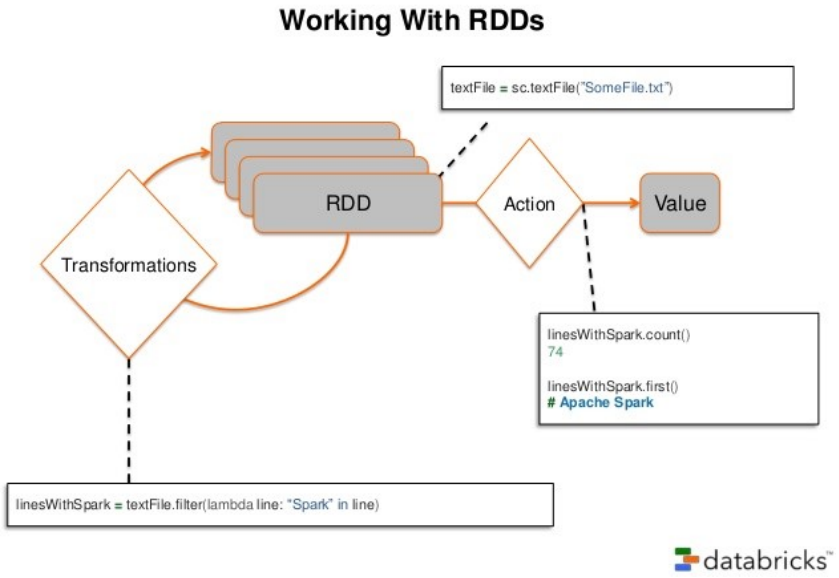

Figure 2.4 RDDs [50]

In figure 2.4, the process starts by producing the RDD by loading a text file. Then it produces an other RDD by executing the filter transformation.

In figure 2.4, the process start with loading of the textfile which creates the first RDD of the program. It then performs a filter transformation, which in result return a new RDD. In last, the program executes two actions of counting the file and displaying its first line.

2.5. SPARK DIRECTED ACYCLIC GRAPH (DAG)

Spark DAG model is the counterpart of MapReduce model which has two stages of map and reduce while spark model can be of any stages as in the figure 2.5.

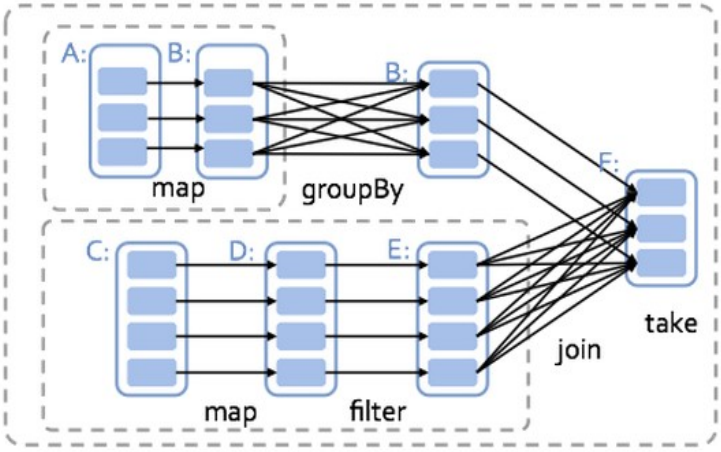

Figure 2.5 Spark DAG model [51]

The figure 2.5 explains the way spark operates. In first stage, it performs the map operation and in second stage, it executes the operations of map and filter. In last, it combines both stages at third stage to produce the final results.

2.5. MACHINE LEARNING LIBRARY (MLLIB)

The core of the spark integrates many useful tools. The machine learning library is one of them. Although it is a part spark core, hence it can be used by other spark libraries. Mllib applies their algorithms on the cluster composed by spark and it operates on the datasets distributed along the clusters. Apparently, MLlib works its traditional machine learning data types like vectors and labeled points but these data types are eventually converted to RDDs [52].

2.6. SPARK STREAMING

For stream analytics, we use Spark Streaming engine. It is used to process the streaming or online data with high throughput. It is very scalable and fault tolerant streaming engine [52].

The figure 2.6 shows that the data ingested to spark streaming by different ways like HDFS, Kafka and Flume etc. After processing of data, it can be stored in database or HDFS, and or visualized using any dashboard like kibana.

Figure 2.6 Spark Streaming [53]

Actually, spark streaming engine divides stream of data into small units called batches. The spark then process these batches one by one and also produce results in the form of batches as shown in the figure 2.7.

Figure 2.7 Spark stream processing [53]

2.7. SPARK STREAMING TECHNIQUES

There are two methods to process streaming data.

(1)Process each record as separate entity.

(2)Combine records and form a minibatch and then process

batches. The collection of minibatches is called as DStream and each minibatch is represented by RDD in Spark as shown in figure 2.8.

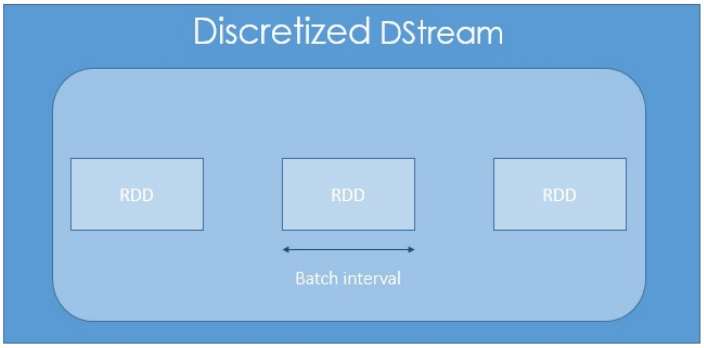

Figure 2.8. Spark DStream [52]

Each mini-batch is allotted a time slot which is called time interval. In this interval, the input of DStream is coverted to RDD. Each RDD represents the mini-batch and mini-batches are collection of records.

2.7.1. OPERATIONS IN SPARK STREAMING

DStreams consist of RDDs and RDDs can perform transformations to different RDDs in spark. Spark has made streaming transformations easy by leaving similar type of RDDs as in general. One of the main transformations are map(), flatMap(), filter(), count(), union(), reduce(), join(), repartition(), countByValue(), countByKey(), reduceByKey(), updateStateByKey(), and transform().

2.7.2. WINDOWS OPERATIONS IN SPARK STREAMING

Windows operation is very important concept in spark streaming. It basically can be explained with two terminologies depicted in figure 3.9 which are Window duration and Sliding window. Window duration indicates the time taking by spark streaming to obtain the data and Sliding window is the place where spark collects all DStream data and it use the window concept of open and close. It opens when start collecting and closes when it is full and need to start processing. The window operations include window(), countByWindow(), reduceByWindow, reduceByValueandWindow(), and reduceByKeyandWindow().

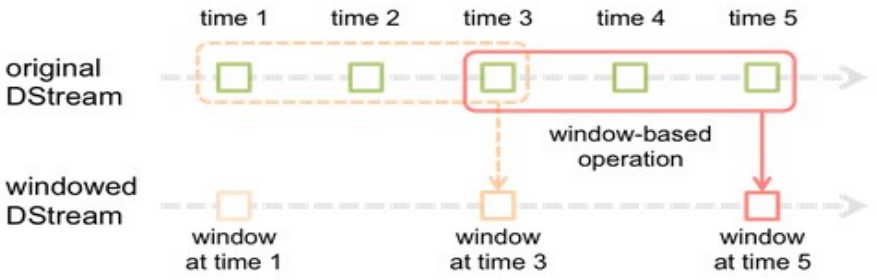

Figure 2.9. Spark window operation[52]

2.8. LANGUAGES SUPPORTED BY SPARK

	Scala	**Python**	**Java**	**R**
Easy to use	Yes	Yes	No	Yes
Own libraries	No	Yes	Yes	Yes
Performance	Yes	Yes	Yes	Yes
Own Spark documentation	Yes	No	No	No
Own Spark libraries	Yes	No	No	No

Table 2.1 Spark languages comparison

2.9. APACHE KAFKA

Kafka is a dispersed, apportioned, imitated confer log service. It gives the feature of messaging system. As said, this framework is brilliant when we have different information sources. In the following section, we discuss the features of kafka which makes it a well known messaging system. The architecture of Kafka was designed to reduce the complexity posed by other related systems. Its architecture mainly consists of Producer/Consumer messaging system as shown in figure 2.10.

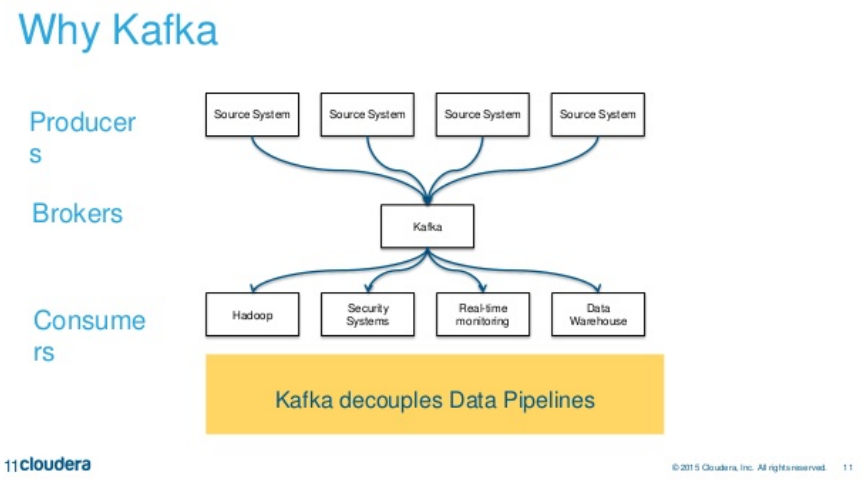

Figure 2.10. Apache Kafka Producer/Consumer architecture

Let's now discuss the fundamentals of Apache Kafka: The main components of Kafka are Topics, Brokers, Producer and consumer as shown in the following figure 2.11

Figure 2.11 Kafka architecture

Topics

Messages of kafka are divided into entities called as topics. In kafka, the messages are sent and received from the topics. Producer writes the message to topics and the Consumer reads the message from the topics. As the Kafka comes in the category of distributed systems, therefore it processes in the shape of clusters and the clusters consists of various nodes. Each node is called the Kafka Broker.

The kafka topics are alienated in number of partitions as in the figure 2.12. The partitions play an important role in parallelizing the topics and delivering them to multiple brokers. Every message in a partition is distributed in number of units called as offsets. These offsets are unchangable sequence of messages. The consumer specify the starting offset to initiate the reading up-to the entire message.

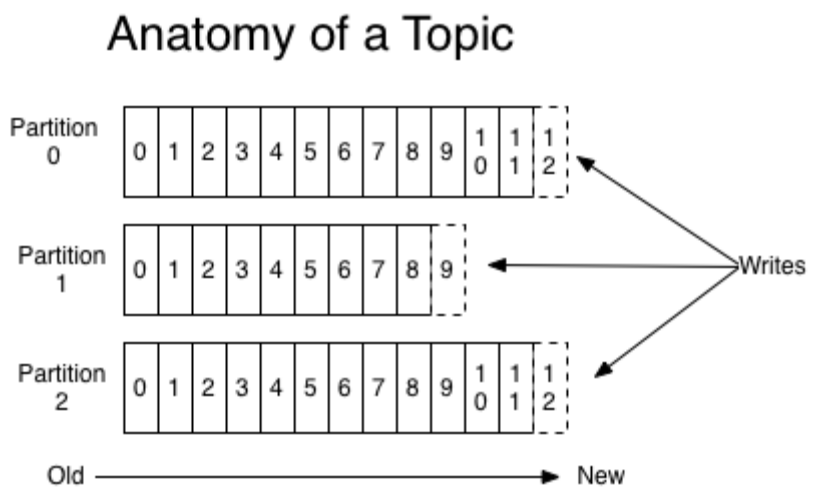

Figure 2.12 Kafka topic [54]

Broker

The partitions are delivered to brokers and every broker consists of number of partitions as shown in the figure 2.13.

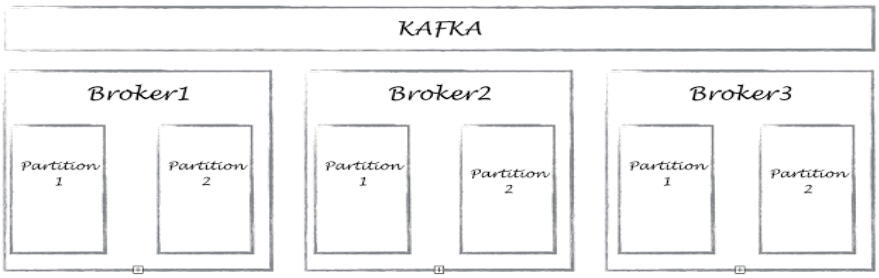

Figure 2.13 Kafka broker [55]

Producer

It is an important element of apache kafka. It is responsible of generating messages and publishing them to the topics. It also decides the number of partition to which messages are sent.

Consumer

The messages produced by kafka producer are read by Kafka consumer. In case of Publish-Subscribe models the messages are distributed to all consumers. Every message is published to a topics and than it is read by the consumer or consumer group which subscribes it.

CHAPTER 3. MACHINE LEARNING ALGORITHMS

3.4. SUPERVISED ALGORITHMS

In these kinds of algorithms, the dependent variable (result) which is anticipated from given independent variable which are likewise called as predictors. In this procedure, we make a function which changes a given input to anticipated output. The algorithm is trained with given arrangement of data until the point when it gets the needed rate of accuracy. Some of the supervised algorithms which are used in this project are Support vector machine (SVM), Logistic regression, Naïve Bayes, Decision trees and Random forest.¨

3.5. SUPPORT VECTOR MACHINE (SVM)

This method is sort of classification. In this method, every datum item is plotted in space of n-dimensions where n denotes to number of features and here, each feature value is the estimation of particular facilitate. For instance, we take two features like tallness and hair length. These features are plotted in two dimensional space. Each point in the diagram contains two co-ordinates and they are called as help vectors as in figure 3.1.

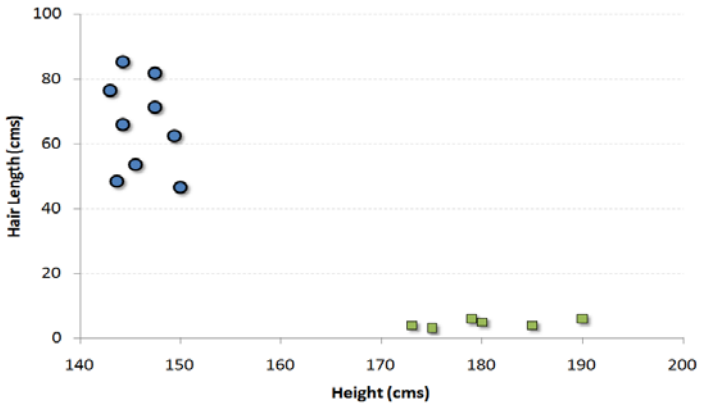

Figure 3.1 Support Vector Machines [56]

Now, we make a line between these two groups. The line is drawn in such a way that each closest point to the line must be at maximum distance as in figure 3.2. The black line is at maximum distance to the closest points of the group. SVM will classify each testing data to its related side according to its features.

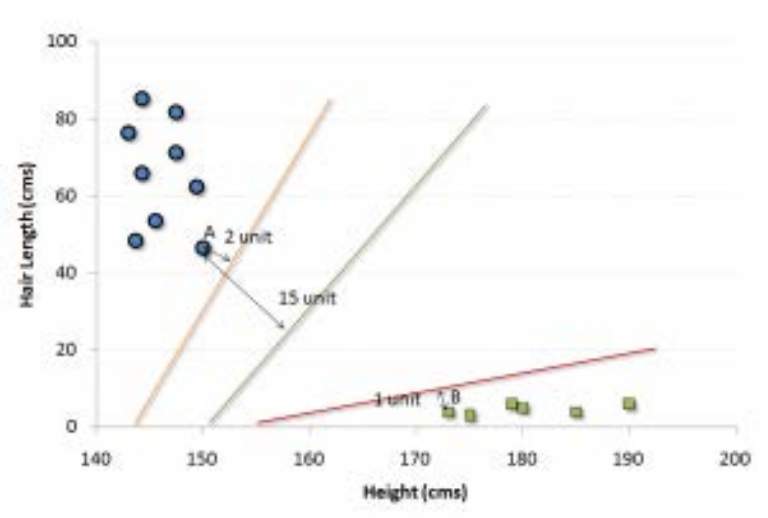

Figure 3.2 SVM classifier [56]

3.6. LOGISTIC REGRESSION

It is also a sort of classification algorithm. The outcomes of this algorithms are discrete values like zero or one, true or false , yes or no. These values are assessed by an arrangement of given free factors. It really utilizes logit function to anticipate the likelihood of occurrence of an event. As stated, it predicts the likelihood or probability, in this manner its outcomes are dependably between 0-1. Algorithm regression is exceptionally useful method uncommonly in attack or spam recognition [57].

Figure 3.3 [58] delineates the case of algorithm regression where x-axis demonstrates the evaluated probability of infected hosts and y-axis demonstrates the known values in the training data. In this procedure, some threshold is set for-instance: host with probability of above than 4 are delegated contaminated.

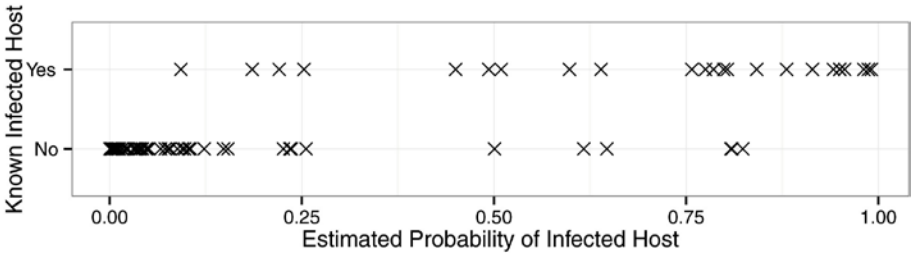

Figure 3.3 Logistic regression [58]

3.7. NAÏVE BAYES

It is sort of classification method and it depends on Bayes hypothesis. It says in regards to autonomy between predicators which implies one component in a class not identified with other element. For instance: A man with tallness 6 inch and weight 80Kg, where naive Bayes will treat each component autonomously, however they are features of same substance. Naïve Bayes model is very good at large datasets and it is also easy build. Figure 3.4 depicts the example of Naïve Bayes.

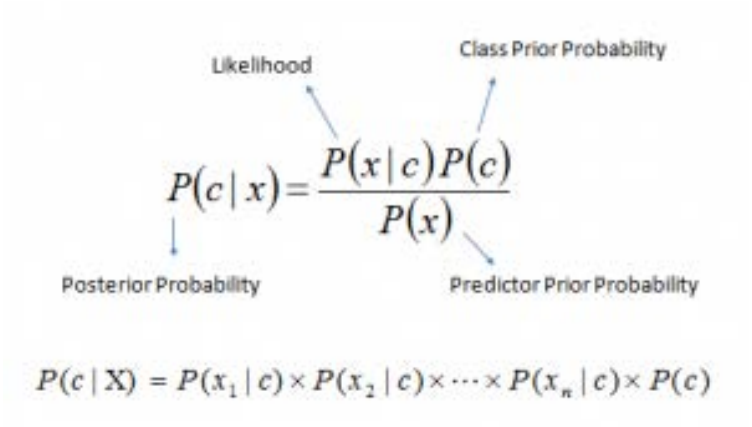

Figure 3.4 Naïve Bayes

3.8. DECISION TREE

This is a sort of supervised algorithm. “It is a predictive modeling technique from the fields of machine learning and statistics that builds a simple tree-like structure to model the underlying pattern” [59]. Decision trees are one example of a classification algorithm. Classification is a data mining technique that assigns objects

to one of several predefined categories [60]. This algorithm performs on both dependent and independent variables. It distributes the samples into two or more similar groups. This division is done on the basis of independent variables.

Example: The figure 3.5 depicts the decision tree example which describes rules generated from learning algorithm. The new network data is constructed on the basis of previously trained results.

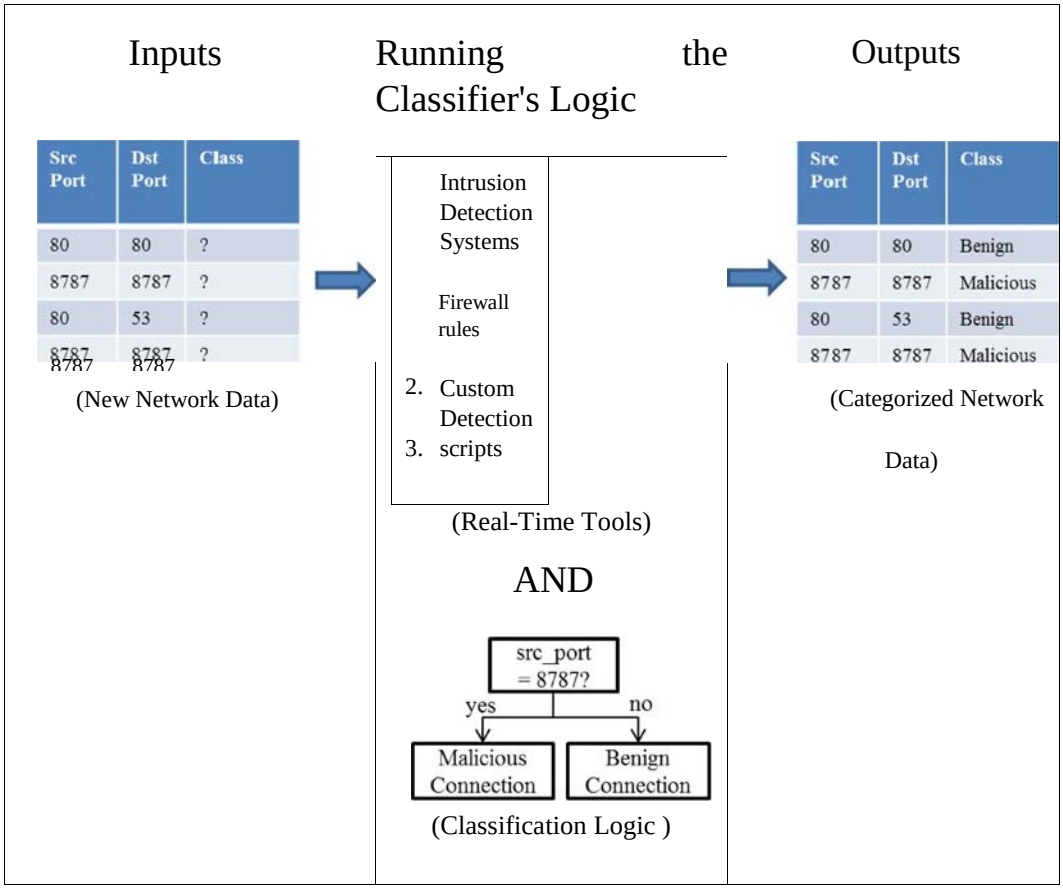

Figure 3.5 Decision Tree [61]

3.9. RANDOM FOREST

It is type of a supervised algorithm. Random Forest makes multiple decision trees and combine them in order to do more correct predictions. It can be utilized for both classification and regression. But in our case, we take it as the classification algorithm. The Random Forest can be better described as in the figure 3.6.

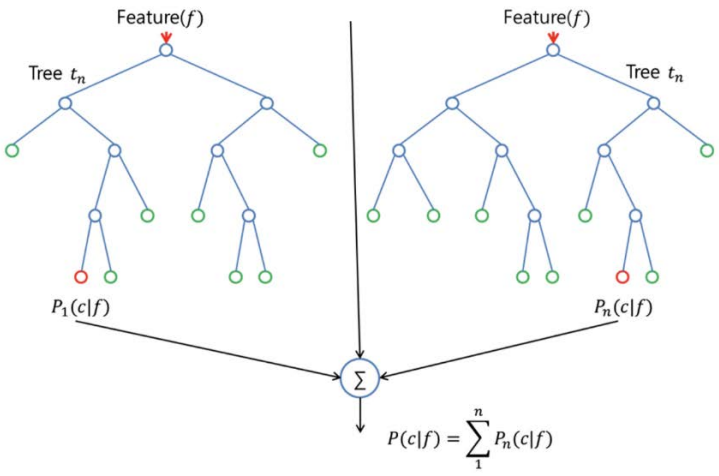

Figure 3.6 Random Forest [62]

3.10. UNSUPERVISED ALGORITHMS

The supervised algorithms are trained in advance about accurate inputs and outputs but in some situations, we do not know about the exact output. In this regard, we are lucky to have methods which are called as unsupervised methods. Unsupervised methods do not require to be trained about target values, hence they don't exist. However these methods can find out itself the structure and likely outputs of the inputs. In our project, we are using unsupervised technique using clustering. In clustering, we are using one of the well-known clustering algorithm called as Kmeans.

3.11. KMEANS CLUSTERING

One of the unsupervised methods is clustering. Generally, they are used to find out the related groups of data. These related groups are called as clusters. In clustering, the most of the work is done using Kmeans. At first, it finds out k clusters in the data, where k is a given input. K is the main parameter of the algorithm and its correct value depends on the data. Finding the exact value is one of task of this research.

In Kmeans, we need to find out the distance between data points. We are using Euclidean distance formula [63] to measure the distance between data points. The formula is given as follows:

$$\boldsymbol{d} = \sqrt{(\boldsymbol{x_2} - \boldsymbol{x_1})^2 + (\boldsymbol{y_2} - \boldsymbol{y_1})^2}$$

In Kmeans, the main point to the cluster is center. All other points in the cluster are called as feature vectors, they are represented in number format and they are also called as vectors. All the points are said be in the Eucleadian space. Central point is called as the centroid. The centroid is calculated by computing the mean of all the points in the given cluster.

The Kmeans start working by selecting some data points and these data points are treated as initial cluster centroids. Then each data point is allotted to the closest cluster. In last, a new centroid is calculated by computing the mean of all data points. This procedure is repeated as in the following flow diagram of the Kmeans Figure.3.7.

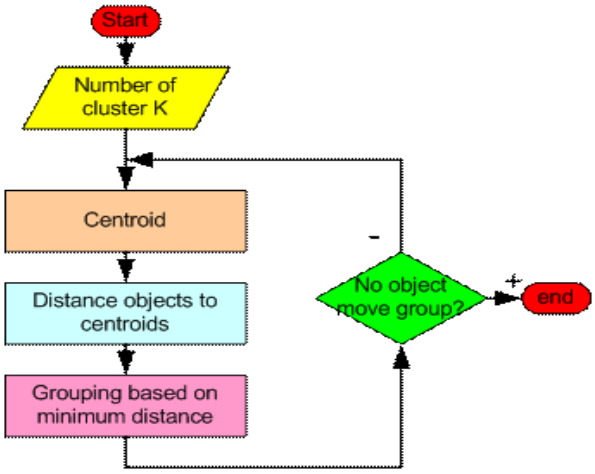

Figure 3.7 KMeans [64]

3.12. STREAMING KMEANS

The objective of k-implies is to segment an arrangement of information focuses into k bunches. The KMeans algorithm repeats between two stages. In the first stage, given an underlying arrangement of k bunch focuses, we discover which cluster every datum point is nearest to. At that point, we figure the normal of every one of the new groups and utilize the outcome to refresh our cluster centers. At every one of these means, we are making the centers inside each cluster increasingly like each other By emphasizing between these two stages over and over, we can for the most part focalize to a decent arrangement.

In the streaming setting up figure 3.8, our data get to batches, with possibly many data items per batch. The easiest extension of the typical k-means algorithm is always to commence with cluster centers , usually arbitrary locations, because we haven't yet seen any data , and for every new batch of data items, does the similar two-step procedure explained above. Then, we utilize the new centers to reiterate the task on another batch. Streaming Kmeans works on the concept of forgetfulness.

If source of the data is continuous , the same three clusters permanently, the aforementioned streaming algorithm will converge to an identical solution as though k-means was run offline on the whole collected data set . In fact, in cases like this the streaming algorithm is similar to a popular offline k-means algorithm, "mini-batch" k-means, which consistently trains on arbitrary subsets of the data to avoid loading the complete data set in memory.

However, imagine if the resources of data are changing as time passes? How do we prepare our model to indicate those changes?

For this environment, we have expanded the algorithm to aid forgetfulness, permitting the model to adjust to changes over time. The main strategy is to include a fresh parameter that amounts the relative need for new data versus previous data.

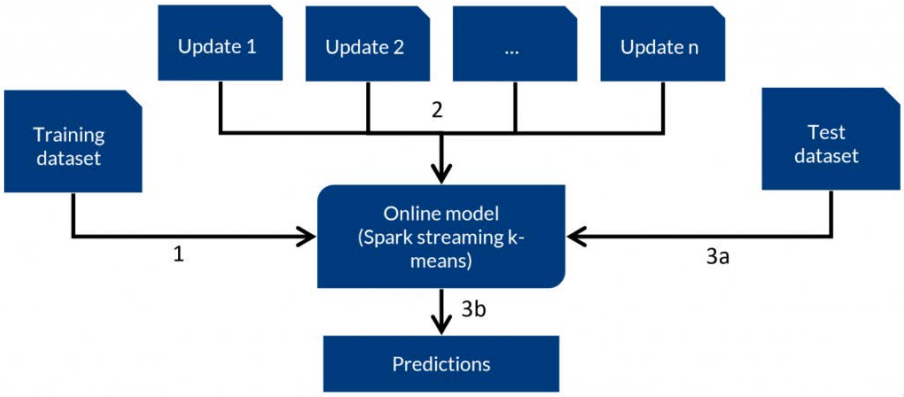

Figure 3.8 Streaming KMeans [65]

3.13. STATISTICAL METHODS

3.14. GAUSSIAN DISTRIBUTION MODEL

Gaussian distribution or Normal distribution [66] is a to a great degree standard probability dissemination that approximates the direct of various typical wonders. An information gathering is known as "normally distributed" when a huge segment of the data adds up to around its mean in a symmetric way. Values end up being less and less slanted to happen the more remote they are from the mean

Right when a metric is typically appropriated as in figure no.3.9 it takes after some entrancing law:

The mean and the middle are the comparative when both are equivalent to 1000 for this condition. This is a result of the superbly symmetric "bell shape".

The standard deviation, called sigma (σ), symbolizes how far the typical conveyance is disseminated in the mean.

In the Gaussian circulation display 68% of all qualities comes in class of [mean-σ, mean+σ], 95% in [mean-2*σ, mean+2*σ], and 99.7% in [mean-3*σ, mean+3*σ]

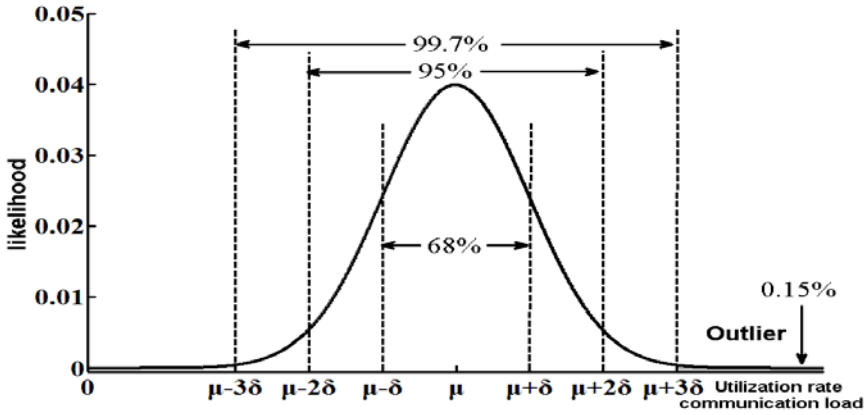

Figure 3.9. Gaussian distribution model [67]

CHAPTER 4. CONCLUSION AND FUTURE WORK

4. CONCLUSION AND FUTURE WORK

The work in this project incorporated the anomaly detection of large scale data. It was done in two stages. In first stage, we distinguish the anomalies from the batch data and in second stage, we applied anomaly detection models on the streaming data.

The large scale examination of batch data comprised of techniques for anomaly detection on premise of supervised and unsupervised machine learning algorithms. In the primary segment of batch analysis, we played out the anomaly detection on both supervised and unsupervised algorithms. In this work, we analyzed three parameters of Accuracy, Training time, and Prediction time between the supervised algorithms and unsupervised algorithms. In the second area of group investigation, we proposed a hybrid model comprising of the anomaly detection of batch data utilizing unsupervised algorithm KMeans clustering and if-else rules. The upside of utilizing this sort of model was that we diminished of the dataset utilizing clustering, and after that it was simple and efficient to additionally identify the remaining peculiarities in the dataset.

The second phase of our project is about the anomaly detection of streaming data. This is additionally isolated in two sections, in the initial segment we distinguish inconsistencies from the streaming data utilizing KddCup99, which is generally utilized dataset for anomaly detection in the computer networks. The investigation of this part depends on Streaming adaptation of the KMeans called as Streaming KMeans. In the second part, we recognize the anomalies from the streaming data produced from the sensor data which is identified with the inactivity estimations of the web servers gathering the utilization data from the metering information. This investigation depends on the Gaussian distribution model which is broadly utilized model for irregularity discovery.

The greater part of the work in our project comprises of the anomaly detection of vast scale information in light of the KddCup99 dataset. For batch examination the data was recovered from KddCup99 dataset document and for stream investigation, the data was produced with the assistance of Kafka Producer from the KddCup99 dataset. As a use case, we analyzed the data created from the sensors.

In this project, our concentration was anomaly detection of network data. The network may made out of computer systems or metering systems. In future, we are planned to recognize the utilization information delivered by the Smart Meters. The investigation will be founded on both the supervised and unsupervised algorithms.

Moreover in future, we will incorporate the database framework comprising of vitality information and apache spark ecosystem for the examination using database queries. To acquire this, the Energy Management System gather the information from metering system. At that point this information will be recovered and investigated with the SQL queries. The batch analysis and investigation is additionally a typical strategy yet analyzing the streaming meter information utilizing sql queries is a challenging task because it is on the grounds that the databases and their capacities are not customary intended for streaming data processing.

REFERENCES

[1] Damminda, Xinghuo, " Advanced Analytics for Harnessing the Power of Smart Meter Big Data", IEEE 2013

[2] Rasim and Yadigar, "Big promises for information security", 8th international conference on application of ICT IEEE 2014

[3] Metke AR and Ekl RL, "Security technology for smart grid networks", Transactions on Smart Grid IEEE 2010

[4] So HK, Kwok SHM, Lam EY, Lui K, "Zero-configuration identity-based signcryption scheme for smart grid", First International Conference on Smart Grid Communications(SmartGridComm), IEEE 2010

[5] Yan Y, Qian Y, Sharif H, "A secure and reliable innetwork collaborative communication scheme for advanced metering infrastructure in smart grid", Wireless Communications and Networking Conference(WCNC) IEEE 2011

[6] Efthymiou C, Kalogridis G, "Smart grid privacy via anonymization of smart metering data" First International Conference on Smart Grid Communications (SmartGridComm) IEEE 2010

[7] Aravinthan V, Namboodiri V, Sunku S, Jewell W, "Wireless AMI application and security for controlled home area networks", Power and Energy Society General Meeting IEEE 2011

[8] McLaughlin S, Podkuiko D, Miadzvezhanka S, Delozier A, McDaniel P, "Multi-vendor penetration testing in the advanced metering infrastructure", Proceedings of the 26th Annual Computer Security Applications Conference 2010

[9] Nasim, Jelena, Vojislav, and Hamzeh, "A framework for Intrusion detection system for advanced metering infrastructure, Security and communication networks", published online 28 november 2012 in wiley online library

[10] Berthier R, Sanders WH, Khurana H, "Intrusion detection for advanced metering infrastructures: requirements and architectural directions" First International Conference on Smart Grid Communications (SmartGridComm) IEEE 2010

[11] Jokar P, Nicanfar H, Leung V, "Specification-based intrusion detection for home area networks in smart grids", International Conference on Smart Grid Communications (SmartGridComm) IEEE 2011

[12] Berthier R, Sanders WH, “Specification-based intrusion detection for advanced metering infrastructures” ,17th Pacific Rim International Symposium on Dependable Computing (PRDC), IEEE 2011

[13] Hae-Duck, WooSeok, Jiyoung, and Ilsun ,“Anomaly teletrafic intrusion detection system on Hadoop-based plateforms: A survey of some problems and solutions”, 15th international conference on network-based information systems IEEE 2012

[14] Shankar and Siddarth,“Big data analysis using apache hadoop”, International conference on IT convergence and security IEEE 2014

[15] Jakrarin and Krerk, “Applying Hadoop for log analysis towards distributed IDS”, ACM 2013

[16] Tariq and Uzma,“Security analytics: Big data analytics for cybersecurity, a review of trends techniques and tools”, 2nd international conference on information assurance IEEE 2013

[17] Alvaro, Pratyusa, and Sreeranga,“Big data analytics for security”, Security and Privacy IEEE 2013

[18] Bhawna and Kiran,“Big data analytics with Hadoop to analyze the targeted attacks on enterprise data”, International Journal of Computer Science and Information Technologies (IJCSIT) 2014

[19] Mukhtaj, Maozhen, Phillip, Gareth and Junyong ,“Big data analytics on PMU measurements”, 11th international conference on fuzzy systems and knowledge discovery IEEE 2014

[20] Prathibha,Dileesh,“Design of hybrid IDS using snort and hadoop”, International journal of computer applications 2013

[21] Michael and Miriam, “Contextual anomaly detection in Big sensor data”, IEEE 2014

[22] Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia, ”Learning Spark- Lightning-Fast Big Data Analysis”, oreilly 2015

[23] Kaufman, Rousseeuw, “Finding groups in data”, Wiley

[24] Assent, Kranen, Baldauf, Seidl, “Any time outlier detection on streaming data”, Springer

[25] Hadoop. http://hadoop.apache.org/

[26] HBase . https://hbase.apache.org

[27] BashReduce. https://rcrowley.org

[28] Mahout. https://mahout.apache.org

[29] Yu and Lan, ”A scalable non-parametric anomaly detection framework for hadoop”, ACM

[30] Gupta, Sharma, Chen, and Jiang, “Context aware time series anomaly detection”, SDM 2013

[31] Zaharia, Chowdhury, Das, Dave, Ma, Mccauley, Franklin, Shenker, and Stoica, “Fast and interactive analytics over Hadoop with spark”, USENIX 2012

[32] Solaimani, Iftekhar, Khan, Thuraisingham, Ingram, “Spark-based anomaly detection over multisource vmware performance data in real-time”, USA 2014

[33] W. Lee , J. Stolfo , “A framework for constructing features and models for intrusion detection systems,” ACM Trans. Inf. Syst. Sec., 2000, 3, (4), pp. 227–261

[34] Z. Tan , UT. Nagar, X. He, P. Nanda, RP. Liu , S. Wang, J. Hu , “Enhancing big data security with collaborative intrusion detection,” IEEE Cloud Computer, pp. 27–33, 2014

[35] S. Bridges, B. Vaughn, “Fuzzy Data Mining and Genetic Algorithms Applied to Intrusion Detection,” Proceedings of the National Information Systems Security Conference (NISSC), Baltimore,MD, October, 2000

[36] Govinda, Manish, “A framework for fast and efficient cybersecurity,” Conference on Advances in Computing & Communications, ICACC 2016, 6-8 September 2016, Cochin, India

[37] Mustafa, Haque, Khan, Baron, and Thuraisingham, “Evolving stream classification using change detection”, 2014 10th IEEE International conferenceon collaborative computing USA

[38] Yao, Sharma, Golubchik,and Govindan, “Online anomaly detection for sensor systems”, 2010

[39] Lee, Stolfo, Chan, Eskin, Fan, Miller, Hershkop, Zhang, ”Real-time data mining based intrusion”, IEE 2001

[40] Abuaitah, ”Anomalies in sensor network deployments: Analysis, Modeling and Detection”, PhD thesis 2013,Wright state university

[41] Chang, Dean, Ghemavat, Hsieh, Wallach, Burrows, Chandra, Fikes, Gruber, “Bigtable: A distributed structured system for big data”, ACM 2008

[42] Spark. http://spark.apache.org

[43] Storm. https:// storm.incubator.apache.org

[44] S4. https://incubator.apache.org/s4

[45] Zaharia, Das, Li, Shenker, Stoica. “Discretized streams: an efficient and fault tolerant model for stream processing on large clusters”, USENIX 2012

[46] Kafka. http: // kafka.apache.org

[47] Book: Learning spark lightning fast analysis

[48] https://spark.apache.org/

[49] Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., J. Franklin, M., Shenker, S. and Stoica, I. (2012). Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing

[50] https://es.slideshare.net/tsliwowicz/reversim2014

[51] http://blog.cloudera.com/blog/2014/03/apache-spark-a-delight-for-developers/

[52] Book: Pentreath, N. and Paunikar, A. (n.d.). Machine learning with Spark

[53] https://spark.apache.org/docs/1.3.0/streaming-programming-guide.html#linking

[54] https://sookocheff.com/post/kafka/kafka-in-a-nutshell/

[55]https://www.happiestminds.com/blogs/apache-kafka-building-intelligent-systems/apache-kafka/#.WyzZJqczY2w

[56]https://www.analyticsvidhya.com/blog/2015/08/common-machine-learning-algorithms/

[57] Book: Research Methods for Cyber Security By Thomas W. Edgar, David O. Manz

[58] Book: Data driven security_analysis, visualization and dashboards

[59] PredictionWorks: Data Mining Glossary. (n.d.). PredictionWorks. Retrieved February 11, 2011, from http://www.predictionworks.com/glossary/index.html

[60] Tan, P., Steinbach, M., & Kumar, V. (2005). Introduction to data mining . Boston: Pearson Addison Wesley

[61]https://www.sans.org/reading-room/whitepapers/detection/decision-tree-analysis-intrusion-detection-how-to-guide-33678

[62] https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd

[63] https://en.wikipedia.org/wiki/Euclidean_distance

[64] https://iitrdsg.wordpress.com/2016/06/15/k-means-clustering-explained/

[65] https://www.inovex.de/blog/online-offline-machine-learning-network-anomaly-detection/

[66] https://anomaly.io/anomaly-detection-normal-distribution/

[67] Dan-dan CHEN*, Zhi-qiang LI, Tian LI, Ming-xing TENG and Feng XIE, “Big data based intrusion detection of Smart Meters”, 2017 2nd International Conference on Computer, Network Security and Communication Engineering (CNSCE 2017)

ISSN (online): 2446-1636
ISBN (online): 978-87-7210-220-7

AALBORG UNIVERSITY PRESS