Fast and Accurate classification of network threats in IDS using Distributed Machine learning techniques

MSc Internship
Cyber Security

Ritesh Naresh Gohil
Student ID: x18205836

School of Computing
National College of Ireland

Supervisor: Mr. Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ritesh Naresh Gohil
Student ID:	x18205836
Programme:	Cyber Security
Year:	2020
Module:	MSc Internship
Supervisor:	Mr. Vikas Sahni
Submission Due Date:	17/08/2020
Project Title:	Fast and Accurate classification of network threats in IDS using Distributed Machine learning techniques
Word Count:	6670
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature:	Ritesh Naresh Gohil
Date:	17th August 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Fast and Accurate classification of network threats in IDS using Distributed Machine learning techniques

Ritesh Naresh Gohil
x18205836

Abstract

An incremental rise in the size of the data has placed a significant impact on the security of data. The advancement in Intrusion detection system can enhance the network security by monitoring and analyzing the large network data. Due to pattern recognition and abnormal behaviour detection capability of machine learning, it became very popular among the researchers to reduce the fraudulent activities. As the network system generates huge volume of data, analysing and detection of attacks in a timely manner is a challenging process. Even, after achieving the results with good accuracy using traditional machine learning approaches. This method lacks to handle large volume of data, due to the non-scalable nature and limited resource capability. Therefore, an efficient, flexible and scalable solution is required for detecting multiple network attacks in a timely manner. In this work, we are proposing a distributed machine learning solution using hadoop and spark framework for anamoly based detection of network attacks. As the network attack can be classified into the multiple categories, it becomes a multi-class classification problem. In this work, we will use logistic regression, random forest and naive bayes as the classification algorithm and compare the result using both traditional and distributed approaches by utilizing the precision, recall, f1-score, training time and accuracy as the performance measures.

1 Introduction

The recent evolution in the technology and the advent of numerous machines, the size of the data has become vast than before. Due to the presence of such huge data, the monitoring process gets very difficult. Since the user count has been hiked to huge amount it has also led the way to the rise of cyber threats and malware attacks. These threats and attacks have the potential to cause great damage and loss of valuable information to the system. These cyber-attacks will also hinder the growth of your organisation and cause major drawbacks in the economic and social growth. Without any proper detection and blocking mechanisms, these cyber threats tend to grow more each day adversely impacting the development of technology and businesses. On the internet, there are many ways to infect a network with malware. Along with the evolution of technology, it has been evident that the malicious software is also getting upgraded into a more smarter version that the existing firewall and protection mechanism find it difficult to identify and stop its presence if found on any network. They are not effective enough and it leads to increased cyber-attack attempts. This paves way for a void in the domain of cybersecurity leaving the network and systems without proper shielding from cyber-attacks. There is

a need for a better detection mechanism that can protect the network from the attack of malware. As a result of many existing detection methods, we can identify and group the present cyber-threats and malware attacks that are aimed to the network and the system.
In recent days, the intrusion detection systems are termed as the best response mechanism to all kind of cyber threats in safeguarding the network. It is the best method of cybersecurity that one can rely on. When an unrecognised software or user attempts to interfere into any other network or a system without any formal notice or knowledge of the original user is termed as an intrusion. Such a process can lead to severing the valuable information or any manipulation of data in the network with the permission of the user. These losses can sometimes be unrecoverable. The IDS is a kind of software that monitors every activity of the network thoroughly. It runs the action mechanism throughout the network searching for even the least clue of detecting the presence of malicious software. It analyses every possible hint that can be identified in the network to find out any backlinks. If any malicious presence is detected to be present on the passive network then the IDS alerts the user by sending in the notification and secures the data and information from the found malware. But when the network is performing and is in an active state the working of the intrusion detection system will be very different in the function than what is described above. In an active network, the intrusion detection system tries to block and turns off the attempt that is made to penetrate the user's network and then activates the firewall mechanism to stop any further attempt of intrusion that is made. Moreover, to create an enhanced protection model, a more hands-on technique must be implemented which must have the capacity to detect the presence of malware as well as prevent it from corrupting the network domains. A device that is capable of doing such work is called the intrusion detection system mostly popularly known as the IDS. The working mechanism of intrusion detection is shown in Figure 1.

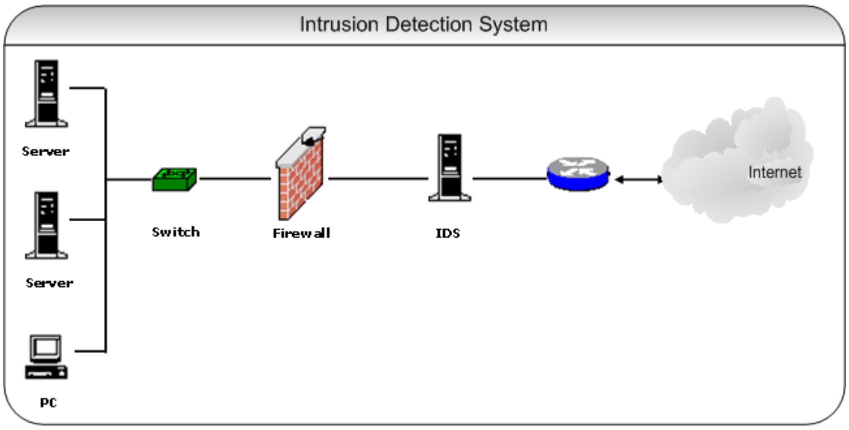

Figure 1: Intrusion detection system

In recent days, there are a lot of new enhanced techniques of IDS that is based on the machine learning protocols. Since there is the active involvement of the enormous amount of data in the process of identification and detection of malware in the network. Due to the limited computing capability and non-scalable nature, the traditional methods of machine learning fail to work well. With the use of the traditional machine learning methods, the desired accuracy and the enhanced prediction rate is nearly impossible to achieve. This is where the efficient usage of big data comes into the picture. It can be the best method to solve the problem of analysing huge data and detecting the presence of malware or intruders in the network system. In this research paper, a comparative analysis is made between the traditional machine learning techniques and the distributed machine learning algorithms. In the first type, the sample algorithms are modelled in a single machine and the latter multiple nodes or clusters are used. To process large data using distributed machine learning techniques, spark framework is used and to store large amount of data in the distributed manner, Hadoop HDFS is used. Apache Spark with ML library enables to deploy machine learning model in distributed manner. Some of the algorithms that are used in the comparative analysis are Logistic regression, Multinomial naïve Bayes and Random forest. All these techniques are implemented and the outcomes are taken into consideration for the analysis. The comparison between the traditional machine learning and the distributed machine learning methods are done by calculating the rate of precision, accuracy, recall, Training time and F1-score for each algorithm.

1.1 Research Question

Here are the following research questions

- How efficiently the Intrusion detection system can classify the network attacks by using distributed machine learning techniques ?
- Which classification algorithm is more suitable for intrusion detection system and why ?

The following research method is determined to build an efficient intrusion detection system using big data techniques for detecting the multiple network attacks from large volume of data.

2 Literature Review

There has been a lot of previous research methodologies regarding the use of traditional machine learning techniques. Numerous authors have put forth their proposals in the same above-mentioned domain but least of the concern is shifted to the area of big data. Only a handful of research proposals are found in the detection of the intrusion system using the technical advancements of big data. Many of the research techniques are found to be dealt with the CIDDS-001 dataset when the network is in both active and dynamic state while making out the detection process of the presence of malicious software. The NIDS evaluation methods are used to analyse all the factors like Precision rate, Accuracy, the notification of false positives and the rate of detection. There are 11 datasets that are used in the detection process. It classifies and groups all types of cyber-attacks based on their signatures and the patterns of its operation. This comparative process of both the models is examined after analysing all the positive and negative outcomes with a detailed description as mentioned in the research model [1]. The traffic patterns taken from the dataset for evaluation contains information about all the external servers and OpenStack data. From this in-depth research, it is established that the usage of CIDDS-001 dataset if found to be more accurate and effective in generating the desired results than any other datasets used by the existing researchers in their research. On a wider scale Almseidin also proposes a more enhanced model of the malware detection system working on a real-time basis when applied onto a network [2]. In this research model, the dataset called KDD is used. All the algorithms that are needed to be imported into the dataset are well analysed before uploading to avoid the occurrence of any common error. For the analysis, various types of classifies are taken into consideration and they are DT, MLP, Bayes and RT and usual RF. 21 types of cyber threats are defined and groups to be detected. The DOS identifies the registered attacks very clearly. The accurate of the proposed model is found out to be around 80 per cent and the normal error is less than 5 per cent.

The process of testing has nearly 80000 types of threats that are recorded for the detection process. The research also confirms that machine learning techniques must be enhanced to handle such kind of enormous data. They characterized all highlights in different layers of datasets of the systems, for example, HTTP Flood and SID-DoS and so forth [3]. They performed a near investigation to arrange and distinguish the systems utilized in the calculation. Their experiment utilizes different procedures of AI calculations like MLP, Multinomial Naive Bayes and Random forest. All the methods are compared and the outcomes are imprinted to demonstrate the practicability of every procedure. Among all the techniques utilized for the examination model, the MLP convention holds higher exactness and the execution procedure is relatively superior to different models. Noh assesses all the criteria of the Flags that are available inside every TCP to decide and discover the connection between the TCP bundles and the flags. To explore more data about different sort of interruption, the author utilizes the system traffic analysis procedure to distinguish the source and season of such an attack [4]. The previously mentioned model looks at the flags present and the rate of TCP present in the framework. When the current flags and parcel rates are affirmed this method checks for any malware in the framework. The detection process is finished by the count of paces of TCP hails and executing the activity rules utilizing the machine learning procedure. At that point, the warning is gotten concerning any interruption on the off chance that it had occurred.

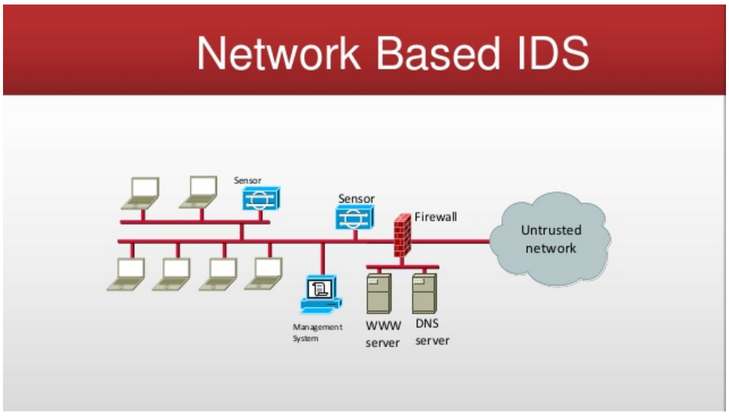

Figure 2: Network based IDS

Saad [5] in his paper keeps forth another investigation by utilizing some of the pattern Traffic of botnets. These standards are examined inside and out and afterwards approved for the analysis. It takes the data of different botnets that are accessible in the use and afterwards investigations it totally to see whether there is any interruption that occurred in it. Sourcing it through the botnets are truly outstanding and proficient strategies that are accessible. There are all the more testing and most current models of botnets accessible so the analysis is done in detail here [5]. A portion of different strategies is utilized here to distinguish the presence of malware in online frameworks. A portion of the normal procedures that are assessed in the paper is flexibility and early discovery. The analysis is done which recognizes the presence of malware utilizing the order and control generally called as (CC). These strategies distinguish the malware in the frameworks through the example of traffic practices and the distinction in the example through deep analysis. The datasets and the aftereffects of the assessment are recorded and demonstrated plainly in their research paper. Sharma utilizes the method of deciding the malware component by investigating the PCAP record with the assistance of a machine learning technique called a DT. To recognize and call attention to the presence of malware in the system a WEKA classifier model has been utilized [6]. The model is formulated with the assistance of the all-out estimation and investigation of all the bundles that are found in the source of a framework. Das in their strategic model clarifies that the SVM (Support Vector Machines) and RST (Rough set Theory) are the most ideal methods to discover the presence of any cyber threats if any occurred. At the point when the information is prepared, the strategy for RST is made to test the data and rearrange the bundle size. It helps in including the size of the information bundle in this manner it is basic for the recognition to calculate and to discover any technique of malware. The chose highlights of the RST are then at last prepared to the SVM for additional examination. The results are then broken down with the aftereffects of PCA (Principal Compound Analysis) which shows that the procedures of SVM and RST are relatively better with more prominent precision. Likelihood approach, Sequential and Paxson's standard are likewise done in this system to check for the location where the interruption has happened [7].

Solankar [8] groups different sorts of interruption model in their paper. Strategies like SVM, Bayes and DT calculations are utilized in their work [8]. There is additionally an additional option of another methodology called Weka to depict and investigate the relative outcomes. Each strategy is dissected inside and out to furnish with the exact working of each procedure. From this itemized investigation, it is obvious from the creator's work that the SVM procedure is more powerful and seen as more exact when contrasted with different strategies of interruption discovery. Tama [9] in their work, give out a concise system for finding out the presence of malware by utilizing the algorithms of machine learning. The creators looked and shortlisted 35 top models that are proposed by other authors from the web on the subject of malware detecting utilizing different calculations. Two databases were made to store the gathered papers as per the methods utilized. They were gathered dependent on the system utilized, qualities and some of the other strategies. The combined strategy was most appreciated among the analysts that contain complex subtleties [9]. The paper comprises of a survey of the shortlisted papers which is present in the domain of malware detection using machine learning methods. Naganhalli in their paper expresses the traffic patterns of different information suppliers that are considered. Factual techniques are utilized to look at the progression of traffic when the data of the traffic information is recovered. Bayes strategy for traffic estimation is likewise examined in detail and illustrative data is given with respect to that. It utilized on the subject of back contingent probabilities which can distinguish the interruption instrument that can happen whenever on the system. The results of the exploration depict the strategy were the parcels are orchestrated by their traffic streams which holds the precision of up to 81 per cent. All the models are brief in their research [10].

More hypotheses are discovered to be identified with applying both the calculations together to recognize the procedure of malware attacks in the frameworks. Liu in their proposal keeps forward the model of the logical arrangement as the principle system. The principle strategy that is utilized in the estimation of interruption instrument by investigating certain inquiries from the data that is recovered from the dataset. The paper proposes a thought of executing the correct technique for cybersecurity for the frameworks that manage more logical documentations [11]. An intricate presentation is given about the current calculations that manage a portion of the measurements utilized and different IDS frameworks. The dataset utilized for the examination is assessed for different data altogether. Their paper is introduced in an unremarkable composing style which set forth the idea of a standard size structure which clarifies different methods of understanding the issues that can possibly emerge in the IDS frameworks. So distant from this paper, unmistakably the utilization of both machine and profound learning conventions are relatively powerful. Showcasing systems are the most powerless against any sort of malware assaults or less exactness which may prompt the disaster of losing complete income and their place in the worldwide market. Hijazi in their paper think of a model which improves the frameworks that control the recognizable proof instrument of interruption in the frameworks. It likewise helps in the improvement of recognition pace of unidentified cyber-attacks that are available in the framework. Some different methods like Deep neural system and model methodologies are utilized for the assessment [12]. Better exactness was gotten from the model for distinguishing the endeavours that are made for the interruption. By utilizing profound learning calculation for the creation of a superior IDS framework the ideal adaptability and exactness are accomplished. Livadas utilizes the customary calculations to distinguish the modified traffic that goes in the botnets of the IRC frameworks. This identification functions as a bit by bit process. The initial step is to separate between the traffic that streams in the IRC frameworks and non-IRC frameworks. The following stage is to discover the IRC frameworks and the pace of traffic that is available in the botnets. For this procedure, strategies like Bayer's strategy and classifiers technique are utilized [13]. It shows the affectability and the degree of exactness in the assurance of such systems. Much bigger arrangements of information are utilized here and the framework gives a precise and dependable correlation between the traffic go. Alenezi gives a great deal of proof of verifications and studies in his examination. The identification of interruption in the DoS is additionally clarified. Point by point presentation of IDS is given in the exploration paper expressing the difficulties and attributes of interruption in the DoS are watched. Different kinds of interruption techniques are recorded and clarified in this paper. Three sorts are clarified in detail. They are DoS based interruption, Network flooding type and General DoS calculation. The focal points and weaknesses are clarified with some run of the mill model models [14]. The CUSUM identification strategies have more in addition to than the numerical and measurable techniques that are broadly rehearsed to distinguish the interruption instrument. The pre-owned techniques in this paper are nonparametric and don't rely upon some other boundaries. It is likewise adaptable to get to different interruption instruments.

3 Research Methodology

Intrusion detection system is an effective tool to monitor the network activities and it is mainly used for the detection of various network attacks on the system. Our proposed Work uses the distributed machine learning method in order to correctly classify the network attacks. The overall proposed framework will be described in multiple subsections. The flow diagram of proposed work is shown in Figure 3. Every step in proposed framework is important in order to achieve the better results.

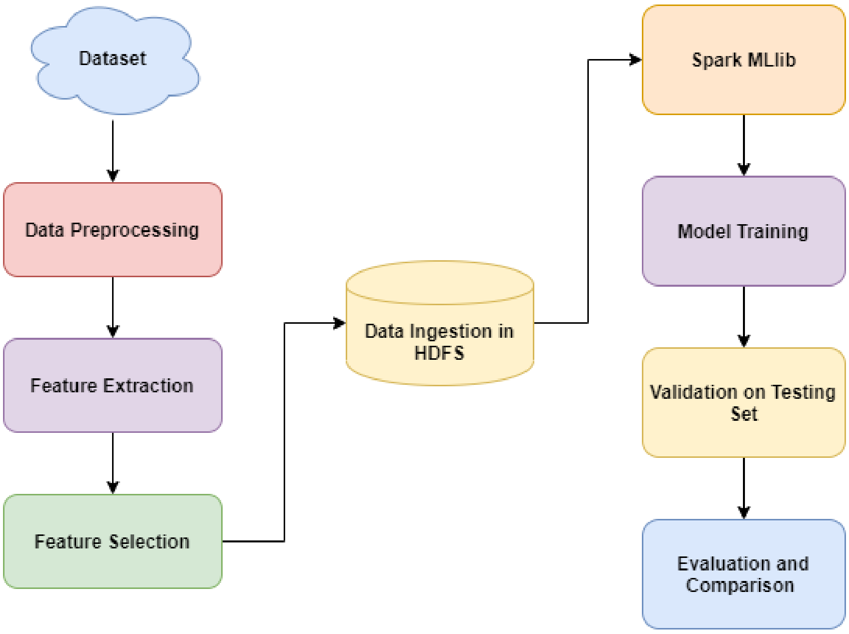

Figure 3: Proposed Methodology For Intrusion Detection System

3.1 Dataset Collection

A large volume of network data is required for predictions. In this work, the network data has been taken from KDD Cup 1999 dataset. Where the main challenge was provided to build a predictive model, which can perform the binary classification between intrusion attack and normal connection. The dataset contains the 2 million connection records and consists of 24 types of different attacks information. The dataset contains 42 different features which will used for our predictive analysis. Each feature either contains the discrete or continuous values. Dataset also contains some derived features. There are 23 different types of attacks available in the data, which can be mapped in 4 categories. Denial of Service (DoS) attack, User to root attack (U2R), Remote to local attack (R2L) and probing attacks.

3.2 Data Visualization

Every feature in the dataset has either discrete or continuous values. We will visually analyze the dataset and its features in order to gain a better insight about the network data. There are 43 features available in the dataset, discussing about each of the feature may not be a feasible option. We will discuss about some of the important and basic features of dataset. Protocol type is a discrete feature in the dataset, its value can be either tcp, udp or any other. On analysing it has been found that 57.85% of the protocol type in network data are TCP, whereas 38.19% protocol in the network data is UDP. Other protocols are rarely used which only contains the 3.97% as shown in Figure 4. Flag is another discrete feature in the dataset which represent the error status of the connection. The flag can either be normal or may contains some kind of error status related to network connection. After visualization it has been observed that 76.44% of the flags are normal whereas, the remaining 23.56% of the flag values has different error status as shown in Figure 4.

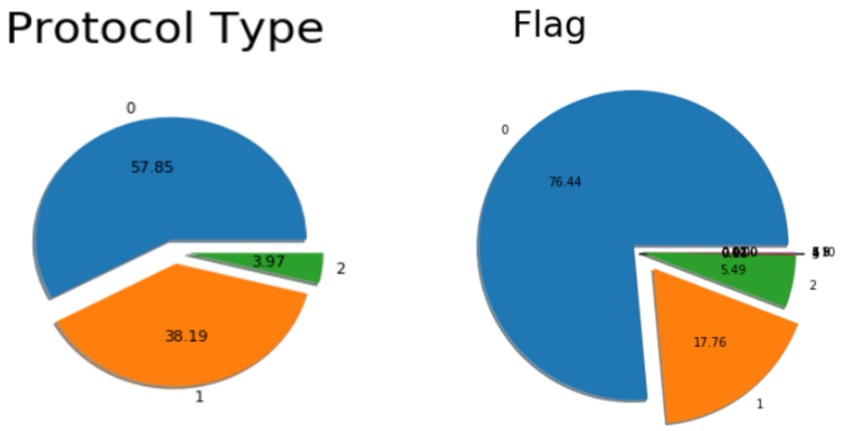

Figure 4: Discrete Features

There are some traffic features in the dataset such as count and srv_count which are continuous in nature. The count feature indicates the many connection to the same machine as live connection in past two seconds. Whereas, the srv_count illustrate that several connections to the same service as the ongoing connection in the past two seconds. The value for both the feature lies between 0 and 500 as shown in Figure 5 .

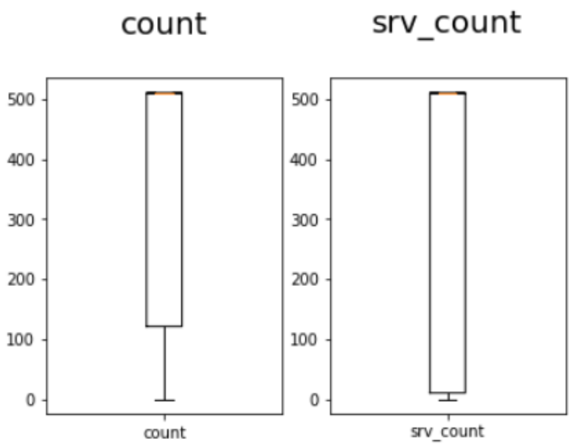

Figure 5: Continuous Traffic Features

There are 5 different categories in the target feature, which mainly represent the attack type. Type 'Normal' represents the safe connection, whereas the other 4 categories are the collection of various network attack. The count of target features is shown in Figure 6

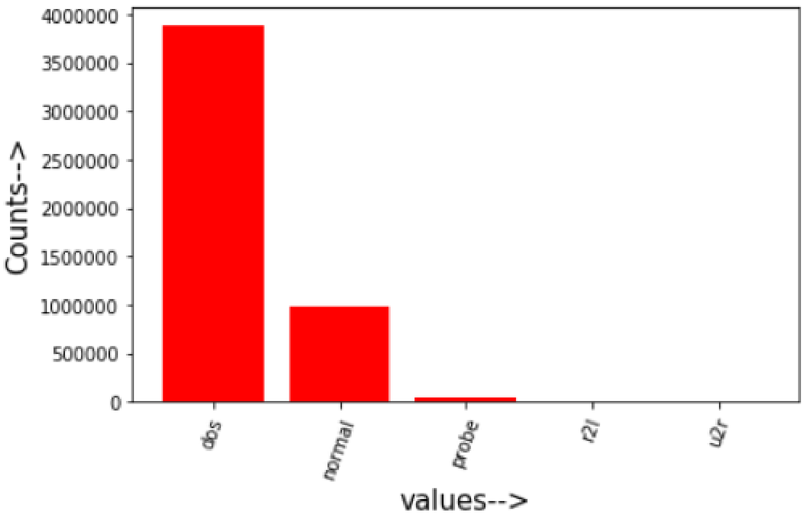

Figure 6: Target Features

3.3 Data Preprocessing

Data pre-processing is very much required in order to achieve the better results. A noisy data may effect the accuracy of the model to a great extent. After collecting and analyzing the dataset, we have applied necessary data pre-processing techniques. Where, we have removed all the null values from the dataset. As the data contains the 24 different attack type information and all these attack fall into the 4 major categories, it has been properly mapped. Dataset has been generalized using min and max scaler. The continuous features of data has been generalized with min-max scaler. Whereas, the discrete feature has been encoded with label encoder.

3.4 Feature Extraction and Feature selection

Features plays an important role for correctly classifying the target features. Highly correlated features reduces the accuracy of model. Therefore, to overcome this problem feature selection becomes a necessary step for our analysis. It is very necessary to remove the highly correlated features. After calculation of correlation values it has been found that, features such as num_root, srv_serror_rate, srv_rerror_rate, dst_host_srv_serror_rate, dst_host_serror_rate, dst_host_rerror_rate, dst_host_srv_rerror_rate and dst_host_same_srv_rate are highly correlated and hence not have been used for our predictions. The feature which have 1 or less than 1 unique values also has been ignored or neglected. After performing the various operation for feature selection we have 35 continuous and discrete features, which we will use for our analysis.

3.5 Data Ingestion in HDFS

For performing distributed computation the data should be available in all the datanodes of the cluster. Therefore, we will utilize the an important component of hadoop ecosystem called as HDFS (Hadoop distributed file system). All the servers connected to hdfs performs the operation in master-slave fashion. Where master node is mainly used for management of the cluster and slave nodes are mainly used to store the data in a distributed manner. Therefore, slave nodes are also called datanodes and master node is called namenode. HDFS is flexible, fault tolerant and scalable solution to store the large volume of data in a efficient manner while maintaining the replication. The sqoop can be used to transfer the data either from local machine or from sql servers to hdfs.

3.6 Spark MLlib

Spark MLlib is a scalable machine learning library, which is compatible with most of the commonly used programming languages such as python, R, scala and java. Due to in-memory computation capability of spark it is faster than hadoop mapreduce technology. Spark MLlib contains many algorithms and utilities which includes classification algorithms, regression algorithms, clustering mechanisms, ML pipeline construction, feature transformation capabilities etc. To efficiently store and process the large amount of network data HDFS and spark with MLlib can be considered as a best combination.

3.7 Model Training and Validation

Network attacks can be classified into various types. Therefore, for accurately classifying the network attacks multi-class classification algorithms should be used. In this work, we are implementing Logistic regression, random forest and naive bayes algorithm for both conventional and distributed approaches. In conventional machine learning the data will be stored in a local hard-disk. Whereas, for distributed machine learning the data will be stored in a distributed manner in HDFS. The dataset is divided into training and validation set with the ratio of 80:20. The final results are mainly based on the validation set.

3.8 Model Evaluation

The machine learning models can be evaluated using the classification performance. Various classification measures are used to evaluate the model performance. Following metrics will be used to measure the classification performance.

3.8.1 Accuracy

Accuracy is most commonly used measure to evaluate the classification performance. The correctly classified network attacks divided by total number of predictions made will calculate the accuracy. The accuracy formula is :

$$Accuracy = (TP + TN)/(TP + FP + FN + TN)$$

3.8.2 Precision

Precision score is mainly concerned about the false positive rates, it is the ratio positive predicted samples that are actually true. A low precision stipulates high false positive rate. The formula for precision is :

$$Precision = TP/(TP + FP)$$

3.8.3 Recall

Recall is the mainly used to calculate the false negative values. It is a fraction of type of network attacks, identified by system. A low recall score stipulates high false negative rate. The formula for recall is :

$$Recall = TP/(TP + FN)$$

3.8.4 F1-Score

F1-Score is mainly based on the calculated precision and recall values, it is weighted average of precision and recall score which considers both false positive and false negative values. The formula for f1-score is :

$$F1score = 2 * ((precision * recall)/(precision + recall))$$

3.8.5 Training time

Machine learning models mainly learns from the data and the time to train the model can be different for every algorithm. We will calculate the training time of every model to evaluate the performance.

4 Design Specification

In this work, we are mainly focusing on the big data solution for large volume of datasets. Scalability, parallel computation, flexibility and distributed nature are the main keys to choose the big data solution for our project. As the vast amount of network data is generated, detection of correct attack type is a quite challenging task. Therefore, we have configured a cluster for hadoop and spark framework consist of 3 nodes. Our cluster comprises of a single namenode and two datanodes. The secondary namenode is configured in the namenode itself. To maintain the high availability of data, we have kept the replication factor as 2. The Architecture used for HDFS is shown in Figure 7.

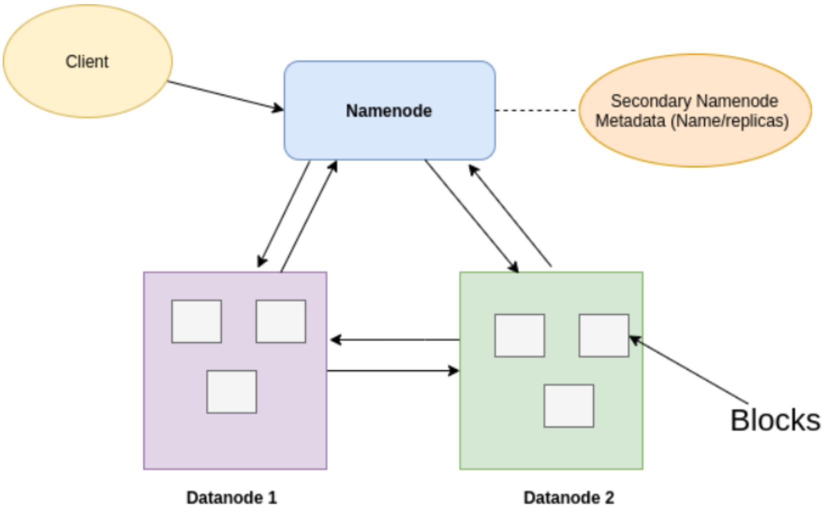

Figure 7: HDFS Architecture Used for Proposed framework

For distributed computation all the data will be stored in HDFS in form of blocks. We have configured the block size as 64MB. The namenode will be used to manage the cluster whereas, the data node will be mainly used to store the data. In case of Spark, the namenode of HDFS will work as a master node of spark and all other datanodes will be considered as a slave node, which will share their computation power for task execution. The spark architecture used for proposed framework is shown in Figure 8.

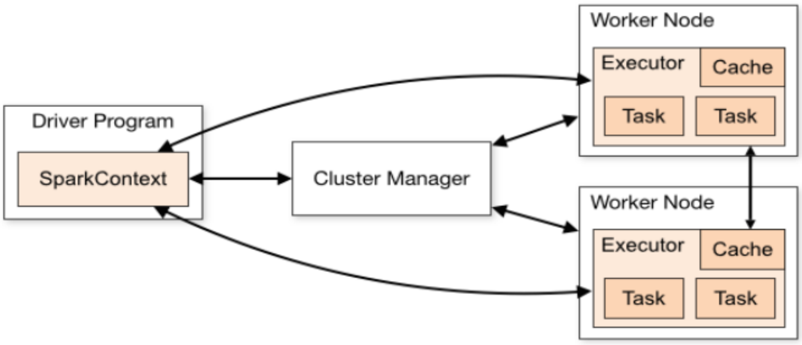

Figure 8: Spark Architecture Used for Proposed framework

5 Implementation

As described in the design section, i have used 3 virtual instances running on vmware workstation. The base operating system is windows. Whereas, the virtual instances is using the base operation system as Ubuntu 18.04. Following specifications are used for implementing the proposed framework.

- Workstation : VMWare
- Number of Instances : 3

- Main memory (RAM) : 4GB (For all nodes)
- Number of Cores : 2 (each node)
- Hard disk : 40GB (Each node)
- Operating system : Ubuntu 18.04 (Debian)
- Framework : Spark
- Programming Language : Python, Pyspark
- Libraries Used : Pandas, MLlib, numpy

For conventional machine learning methods, A local disk with single node is utilized. There is no extra hardware requirements for implementing conventional machine learning algorithms. The architecture used for distributed computation is explained in the Section 4.

6 Evaluation

In the section we will evaluate the model performance with respect to the classification metrics described in the Section 3. We will discuss about the results obtained after performing various experiments. Comparison the frameworks, models and algorithm on single metrics is not an efficient way of analysis. Therefore, we are using five different measures to evaluate the performance. There are 2 different methods conventional machine learning and distributed machine learning used in this work. For each of the method we are using 3 different algorithms named as Logistic regression, Naive bayes and random forest. In order to show the effectiveness for large data, we are using the different subsets of data ranges from small to large dataset. It includes 1000 rows, 10000 rows, 100 thousand rows and 1 million samples. In each experiment we will compare the model and methods with respect to the metrics.

6.1 Experiment 1/ Precision, Recall and F1-Score Comparison

We have discussed about the precision, recall and F1-Score metrics in Section 3. In order to gain the better insight about false positive and false negative rates for each individual machine learning model precision, recall and f1-score metrics are very helpful. We will show the PRF (Precision, recall and F1-score) value for each subset of data for both conventional and distributed approaches. The PRF Values for 1000 samples is shown in Figure 9. From the Figure 9, it has been found that logistic regression has the highest precision, recall and F1-score value for both traditional (Conventional) and distributed approaches for 1000 number of rows. Whereas, Random forest algorithm has the lowest Precision score for traditional approach and naive bayes has lowest recall and f1-score. The highest precision value is obtained using distributed logistic regression approach which is 0.983. The highest recall value is 0.946 obtained from naive bayes algorithm and highest f1-score obtained of 0.952 from random forest algorithm using distributed ML approach.

On increasing the number data samples to 10,000. We have found a increase in the score of precision, recall and f1 for both distributed and conventional machine learning

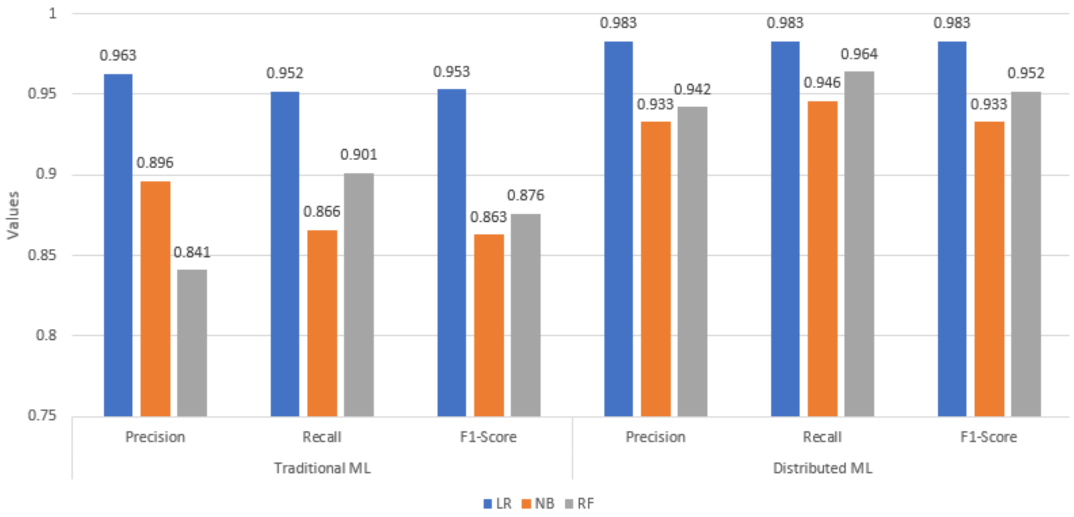

Figure 9: Precision, Recall and F1-score for 1,000 network connection samples

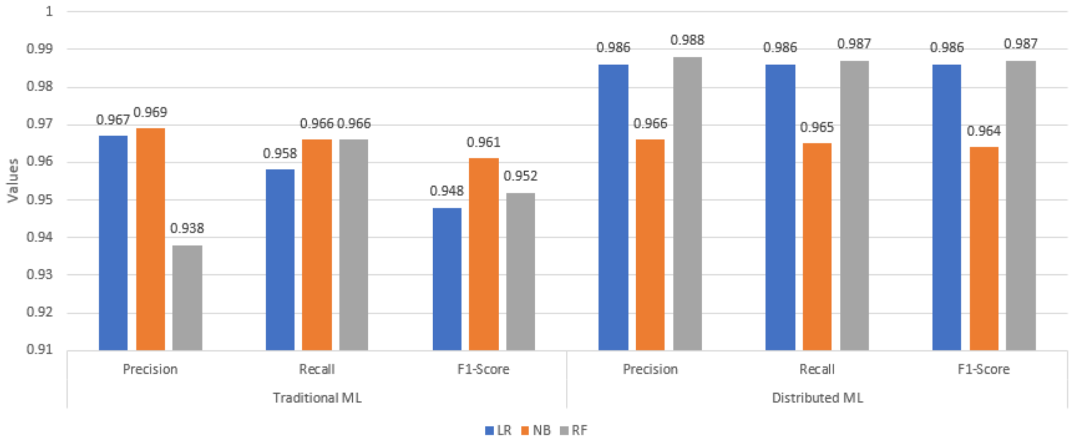

Figure 10: Precision, Recall and F1-score for 10,000 network connection samples

approaches.The random forest algorithm provides the highest precision value in case of distributed approach, where it provides the lowest precision values for traditional ML approach. It means in traditional approaches, random forest algorithm is dealing with more false positive values.The PRF score for distributed approach is high for all the algorithms as compared to traditional approach as shown in Figure 10.

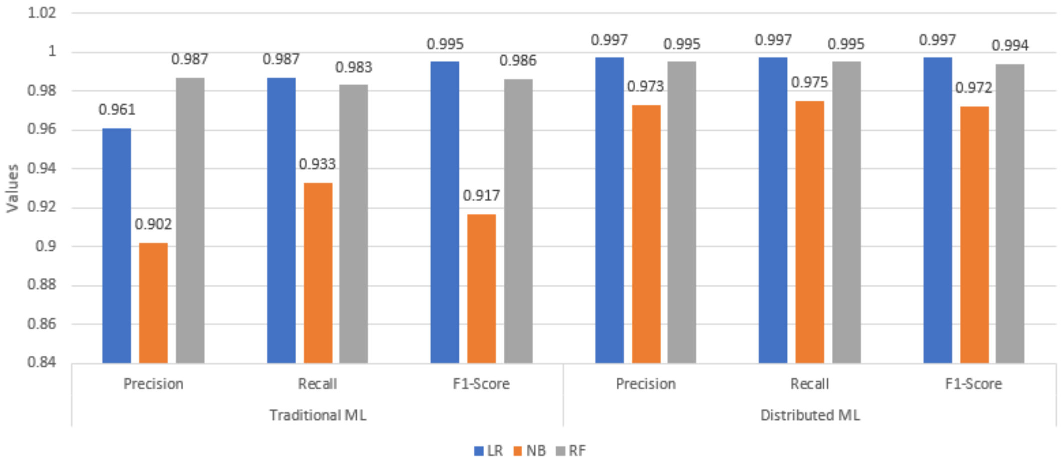

Figure 11: Precision, Recall and F1-score for 100,000 network connection samples

On analysis of PRF value on subset of 100 thousand network connection samples as shown in Figure 11, we have obtained highest precision,recall and f1-score for distributed approaches. Whereas, in case of conventional machine learning approach the highest precision values has been obtained using random forest algorithm, which is still less than precision value of distributed random forest. The lowest precision value has been provided by naive bayes algorithm of 0.902 by traditional approach.

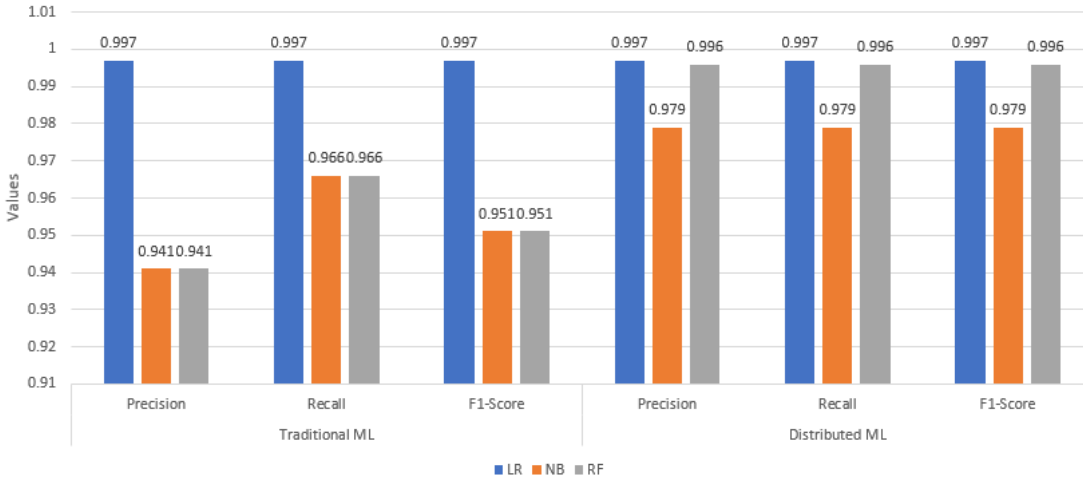

Figure 12: Precision, Recall and F1-score for 1 million network connection samples

Next experiment was performed over 1 million samples the obtained precision, recall and f1-score are shown in Figure 12. On analysing the graph we have found that precision, recall and f1-score obtained by logistic regression are highest and almost same with the value of 0.997. The lowest precision, recall and f1-score has been obtained by naive bayes and random forest algorithm.

6.2 Experiment 2 / Accuracy Comparison

Accuracy is an important measure to represent the effectiveness of any machine learning model. Therefore, in our experiment we are considering the accuracy as an important metric. The Evaluation of accuracy is performed based on the algorithm for both traditional and distributed approach. Based on the previous studies we have found that, logistic regression has given a significantly the good results as compared to other algorithms. Therefore, the first experimental algorithm used is logistic regression. The line graph has been used to represent the accuracy for both distributed and traditional approaches.

The change in accuracy on different subset of dataset for logistic regression is shown in Figure 13. The line graph shows that on increasing the size of dataset, there is improvement in the accuracy of the model. The highest accuracy has been achieved is 99.73% by distributed approach using logistic regression whereas, using the traditional approach accuracy of 99.17 % has been achieved for 1 million network connection samples. The accuracy obtained by logistic regression for both distributed and traditional approach is good but in respect to comparison distributed mechanism provide the better results. Same experiment above has been applied for naive bayes algorithm as shown using line graph in Figure 14. For distributed approach we have found a increase in accuracy as increase in the size of the data. For 10,000 number of samples the accuracy obtained from both the approach is same. Later, the distributed machine learning has provided the highest accuracy of 97.99%. Whereas, the conventional machine learning approach a highest accuracy of 96.66% which is 1% less than the distributed approach. The accuracy achieved by naive bayes approach is less than logistic regression algorithm. There is significant difference in accuracy around 3% between both the algorithms.

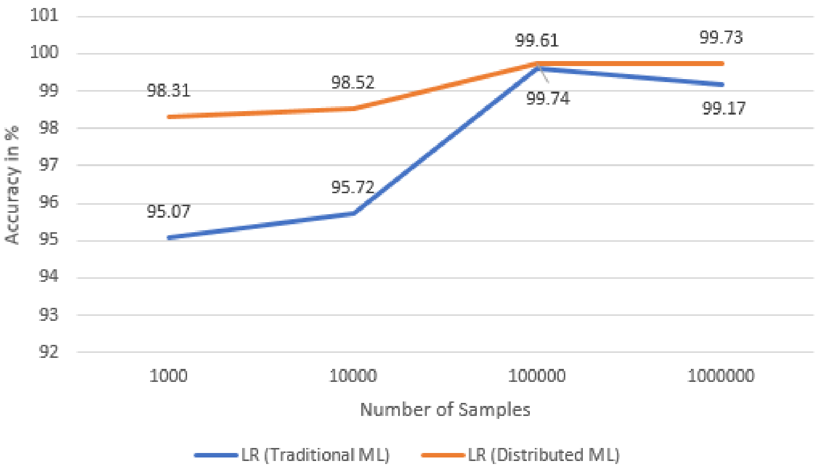

Figure 13: Accuracy Comparison for Logistic regression Algorithm

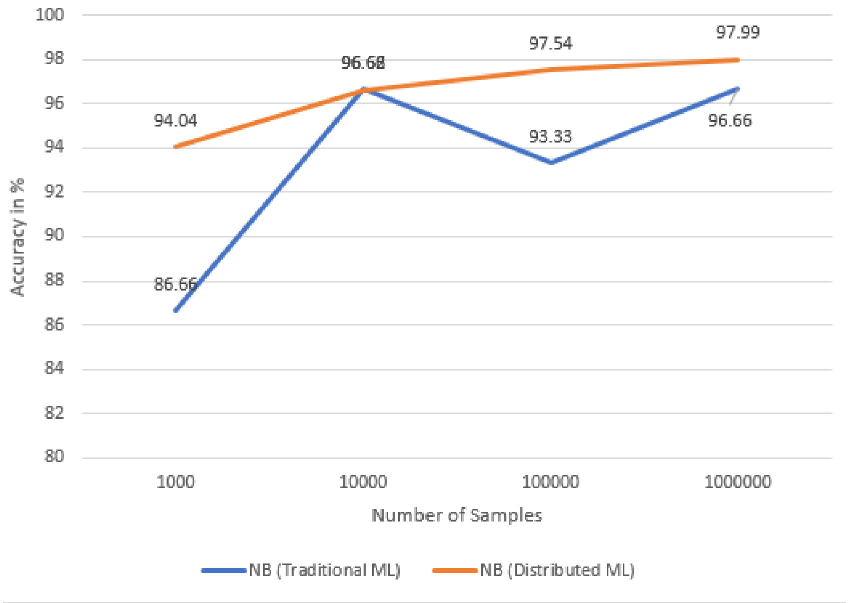

Figure 14: Accuracy Comparison for Naive Bayes Algorithm

Accuracy for random forest has been labelled in the graph as shown in Figure 15. A highest accuracy achieved by random forest is 99.67%. Which is almost similar to accuracy achieved by logistic regression for distributed machine learning approach which is 99.73%. We will conclude our analysis after evaluating the training time as well.

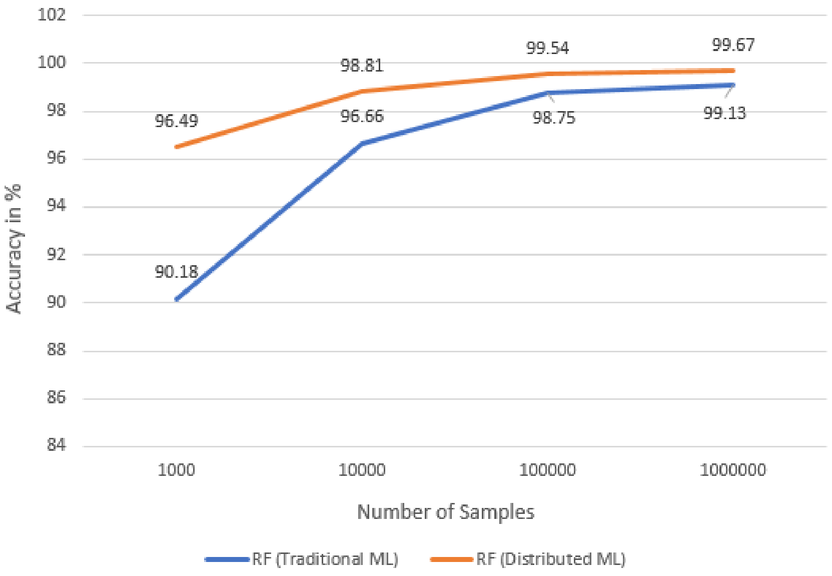

Figure 15: Accuracy Comparison for Random Forest Algorithm

6.3 Experiment 3 / Training Time Comparison

Training time is an important measure because if training time of model is too high, it may delay the results. Therefore, for choosing the optimal and best algorithm for intrusion detection system, we are using training time as a metric. The unit for training time is in seconds. The algorithm with minimum training time, high accuracy, precision, recall and f1-score is considered as the best algorithm. In order to have a better analysis of training time with respect to increase in size of data we are plotting horizontal bar graphs with respect to subset of data.

The bar graph in Figure 16 shows that the training time of model by distributed approach for every model is very high as compared to tradition approach of 1000 samples of data. Time taken by traditional approach for logistic regression, naive bayes and random forest are 0.08, 0.003 and 0.43 seconds. Where for distributed approach it takes the time of 13.09, 6.67 and 7.23 seconds. If we compare the models based on the algorithms naive bayes trains the model in minimum time for 1000 rows. For 10,000 rows approximately same training time has been noted down for all the algorithms of distributed approach. Whereas, the time has increased in traditional approach still is it very very less as compared to distributed approach as shown in Figure 17. The naive bayes again trains the model in minimum time for both traditional and distributed approaches. The lowest training for training 10,000 rows is 0.56 seconds by naive bayes using traditional approach. Where for distributed ML the training time by naive bayes is 7 seconds. Logistic regression algorithm has the highest training time for distributed approach.

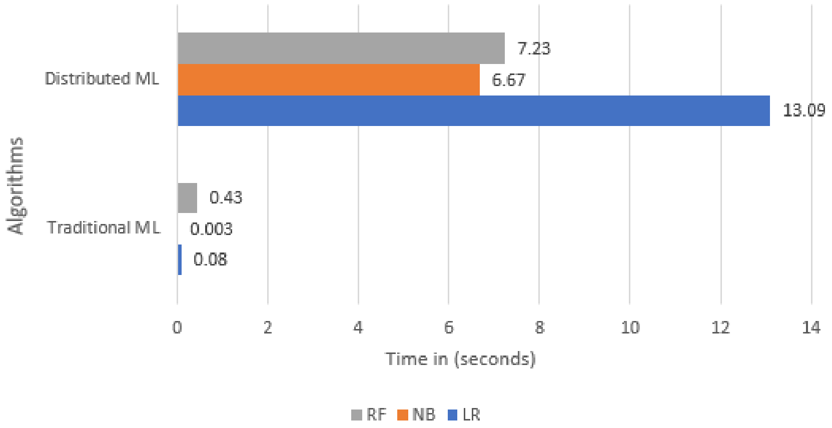

Figure 16: Training Time Comparison for 1,000 samples of data

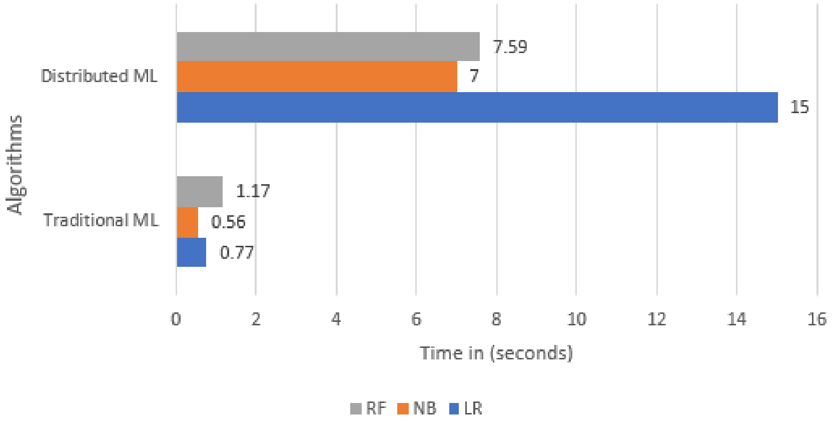

Figure 17: Training Time Comparison for 10,000 samples of data

On Further increase the row count to 100 thousand it has been observed that in some cases random forest with distributed approach is performing well. Whereas, in some cases like logistic regression still the traditional machine learning takes the least time to train the model. The lowest training time of 4.31 seconds has been by taken naive bayes again which is minimum as compared to all the other algorithms. In case of distributed approach also naive bayes wins the race with minimum time of 7.64 seconds as shown in Figure 18.

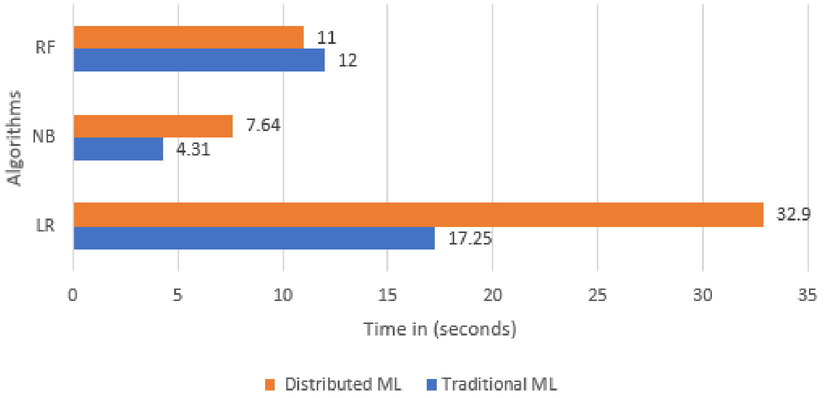

Figure 18: Training Time Comparison for 100,000 samples of data

On considering the highest volume of training data with the samples of 1 million following observation can be made. After increasing the data size to 1 million rows there is significant impact in the training time of algorithms for traditional and distributed approaches. This impact can be seen in the graph shown in Figure 19. On increasing the training data the time for both the approaches has increased but in terms of comparative analysis, the distributed approach takes the least time as compared to the traditional approaches for all the algorithms in proposed work. The training time taken by traditional approach for logistic regression, naive bayes and random forest are 129.39, 13.24 and 194 seconds. Whereas in case of distributed approach, the training time for logistic regression, naive bayes and random forest is 63.44, 8 and 14 seconds which is comparatively very less. From the graph shown in Figure 19, it can be observed that on increasing the datasize the training time for traditional approach has increased exponentially. while for distributed approach training time has not increased to exponential rate and comparatively, it is less than traditional approaches for all the algorithms.

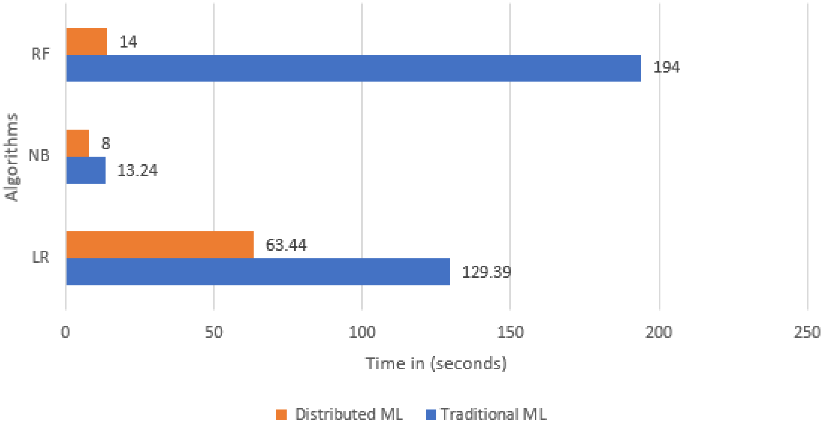

Figure 19: Training Time Comparison for 1 million samples of data

6.4 Discussion

After performing various experiments and analysis, we can conclude that traditional approaches for machine learning performs really well for small subset of data, specially in terms of training time. Although obtained accuracy and PRF value by traditional methods is quite less as compared to the distributed approaches. In terms of performance comparison between the algorithms, logistic regression algorithms provides the outstanding performance in terms of precision, recall, accuracy and f1-score. Although, the time taken by logistic regression to train the model is very high as compared to other algorithms such as naive bayes and random forest. It can also be concluded that, the time taken by naive bayes algorithm is very less for any size of data as compared to logistic regression and random forest, this applies for both distributed and conventional machine learning approach. Almost, for every model we have observed that on increasing the training size of data, the accuracy of the model increases significantly. In terms of accuracy for 1,000 samples naive bayes performs very poorly with accuracy of 86.66%.

7 Conclusion and Future Work

After performing the various experiments, analysis and discussion we can conclude that traditional machine learning approaches are optimal solution, when the volume of data is not high, it takes very less time to train the model and provides the fair performance in terms of accuracy and PRF value. Due to limited capacity of resources conventional(traditional) machine learning approaches may fail, to process the large amount of data or may take indefinite time to train the model. To solve, such challenges distributed, scalable and parallel computation methods are required. Therefore, we propose a big data solution to handle the large data efficiently. Hadoop HDFS not only stores the large amount of data in a distributed manner, it also maintains high availability and flexibility to store the data in a efficient way. Due to distributed, scalable and in-memory computation capability of spark it process the large volume of data in the minimum time. Training time of spark MLlib for small dataset is high, as these solutions are not prepared for operating with small amount of data. In terms of algorithms, a highest accuracy of 99.73% has been achieved using logistic regression algorithm, nearly accuracy of 99.67% has been achieved by random forest algorithm using distributed approach for 1 million rows. In terms of training time naive bayes algorithm took 8 seconds for 1 million samples using traditional approach, in case of distributed approach it took 13.24 seconds. Hence, we can say that logistic regression algorithm should be used as it can accurately classify the network attacks. Hadoop and spark are the efficient tool to handle and process the large volume of data. Still in the future work, more methods such as online learning algorithm can be explored and results can be compared with our proposed model.

References

[1] T. Devi and S. Badugu, "A Review on Network Intrusion Detection System Using Machine Learning," Jan. 2020, pp. 598–607.

[2] M. Almseidin, M. Alzubi, K. Szilveszter, and M. Al-kasassbeh, "Evaluation of machine learning algorithms for intrusion detection system," Sep. 2017, pp. 000 277–000 282.

[3] M. Alkasassbeh, G. Al-Naymat, A. B.A, and M. Almseidin, "Detecting Distributed Denial of Service Attacks Using Data Mining Techniques," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 1, 2016. [Online]. Available: http://thesai.org/Publications/ViewPaper?Volume=7&Issue=1&Code=ijacsa&SerialNo=59

[4] S. Noh, C. Lee, K. Choi, and G. Jung, "Detecting Distributed Denial of Service (DDoS) Attacks through Inductive Learning," vol. 2690, Mar. 2003, pp. 286–295.

[5] S. Saad, I. Traore, A. Ghorbani, B. Sayed, D. Zhao, Wei Lu, J. Felix, and P. Hakimian, "Detecting P2P botnets through network behavior analysis and machine learning," in *2011 Ninth Annual International Conference on Privacy, Security and Trust*. Montreal, QC: IEEE, Jul. 2011, pp. 174–180. [Online]. Available: http://ieeexplore.ieee.org/document/5971980/

[6] N. Sharma, A. Mahajan, and V. Mansotra, "Machine Learning Techniques Used in Detection of DOS Attacks: A Literature Review," 2016, library Catalog: www.semanticscholar.org. [Online]. Available: /paper/Machine-Learning-Techniques-Used-in-Detection-of-A-Sharma-Mahajan/884a0f78f397d01bace956e11c49ed07721c266a

[7] V. Das, V. Pathak, S. Sharma, M. Srikanth, G. K. T, A. V. Vidyapeetham, and T. Nadu, *Network Intrusion Detection System Based on Machine Learning Algorithms.*

[8] P. Solankar, S. V. Pingale, and R. Parihar, "Denial of Service Attack and Classification Techniques for Attack Detection," 2015, library Catalog: www.semanticscholar.org. [Online]. Available: /paper/Denial-of-Service-Attack-and-Classification-for-Solankar-Pingale/7b92f7aef4db055c04e9c0a1340dd8146fe01d7c

[9] B. Adhi Tama and K. H. Rhee, “Data mining techniques in DoS/DDoS attack detection: A literature review,” *INFORMATION, Japan*, vol. 18, pp. 3739–3747, Aug. 2015.

[10] N. S. Naganhalli and D. S. Terdal, “Network Intrusion Detection Using Supervised Machine Learning Technique,” *International Journal of Scientific & Technology Research*, vol. 8, no. 9, pp. 345–350, Sep. 2019. [Online]. Available: http://www.ijstr.org/paper-references.php?ref=IJSTR-0819-21701

[11] H. Liu and B. Lang, “Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey,” *Applied Sciences*, vol. 9, p. 4396, Oct. 2019.

[12] A. Hijazi and J.-M. Flaus, “A Deep Learning Approach for Intrusion Detection System in Industry Network,” Feb. 2019.

[13] C. Livadas, R. Walsh, D. Lapsley, and T. Strayer, “Usilng Machine Learning Technliques to Identify Botnet Traffic,” Dec. 2006, pp. 967–974.

[14] M. A. a. M. J. Reed, “Traceback of DOS Over Autonomous Systems,” 2013, publisher: Academy & Industry Research Collaboration Center (AIRCC). [Online]. Available: https://core.ac.uk/display/25807498