Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 132 (2018) 253–262

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Intelligence and Data Science (ICCIDS 2018)

Network Intrusion Detection in Big Dataset Using Spark

Priyanka Dahiya[a*], Devesh Kumar Srivastava[b]

[a]CET, Mody University of Science and Technology, Lakshmangarh, Rajasthan,India
[b]IT, Manipal University Jaipur, Jaipur, Rajasthan,India

Abstract

Nowadays, huge amount of data is flowing every second; hence intrusion detection task became tedious.Hence, Intrusion detection systems require efficient and improved detection mechanism which could detect intrusive activities and serious threat to network security. Nowadays, huge amount of data is flowing every second; hence intrusion detection task became tedious. In our research work, we have proposed a framework in which a feature reduction algorithm is used for reducing the less important features and then applied the supervised data mining techniques on UNSW-NB15network dataset for fast, efficient and accurate detection of intrusion in the Netflow records using Spark. In this paper, we have used two feature reduction algorithms, namely, Canonical Correlation Analysis (CCA) and Linear Discriminant Analysis (LDA) and seven well known classification algorithms. In order to compare the performance of the proposed framework, five performance matrices such as accuracy, Specificity, Kappa, Mean Abs. Error, FPR, Precision, Recall, ROC Area and Training Time are used.

*Keywords:*Intrusion Detection System(NIDS); Apache Spark; UNSW –NB15 dataset, Receiver OperatingCharacteristics(ROC).

*Corresponding author
E-mail address:dahiyapriyanka814@gmail.com

10.1016/j.procs.2018.05.169

1. Introduction

Day by day we are becoming network and computer technology dependent. It raises the need of secure networks. We have to improve computer network security for data integrity, confidentiality and availability. But these cannot stop intrusion detection. Vulnerable computer systems and networks are required to secure just to prevent risk of unauthorized access and data theft. An Intrusion Detection System scans all packets on the network and attempts to classify the traffic as intrusive or non-intrusive. Intrusion detection is the process which begins where the firewall ends [1-2]. The paper is organized as follows: Section 2 presents the related work. In section 3, ResearchMethodology is discussed, In Section 4, we give a comparative study of these methods based on various metrics, Result is discussed in section 5 and finally Section 6 carries the conclusion.

2. Related work

S.Veetil et al. (2013) [3], they have presented their intrusion detection system that runs a Naive Bayes algorithm in a distributed manner on Hadoop. The classifier in their experiment used the Apache Hadoop and HStreaming APIs to detect intrusions in real time scenario.

Xun-Yi Ren et al. (2013) [4], presented an Intrusion Detection System model with feature multi-classification fusion based on hadoop. They have used Map forming new classification according to the classification centres. Then they have removed the duplicate values reforming a new detection model. They have used KDD CUP99 datasets and their results of testing huge dataset show that the fused classifier has more accuracy than mere classifier.

Junlong Xiang et al. (2014) [5], they have used Extreme Learning Machine (ELM) algorithm which achieve a relatively high Overall Accuracy and can decrease the time of the training phase. For large data or big they proposed a massively parallel algorithm for ELM, that is MR ELM and is a MapReduce variant of ELM.Their experiment results shows that MR ELM have a high speed performance.

In [6], Samuel Marchal et al. (2014) has proposed a solution to cope large data to analyze for security observing insights. They introduced a security observing architecture of networks for intrusion detection and prevention and forensic analysis. They mined different types of data like honey-pot data. It provides the big data solution in a distributed system. They proposed Data correlation schemes and calculated their work in Hadoop and Spark.

They introduced a new scalable NIDS design which collects and stores honeypot data, DNS data. Five big data frameworks were evaluated for security monitoring. After their performance analysis, they found that Shark and Spark were the on top performers in all scenarios and so they were suitable to implement the solution.

Sung-Hwan Ahn et al. (2014) [7], author believed that intelligent new threats are growing and earlier unknown attacks cannot be detected using existing pattern matching methods. They anticipated bigdata analysis solution which is a solution for detecting these kinds of unknown attacks.

In future works, author suggests the researches which must be done to classify the data by context of intrusion detection. This can lead to implement the data relation study methodology and unusual behavior detection approach. Author is hopeful for q*uantitative* and *qualitative* calculation of suggested model and performance estimation in future.

Kleber M.M. Vieira et al. (2014) [8], they proposed IRAS, an Intrusion Response Autonomic System, using Big Data techniques for data analytics for decision taking. Also, proposed a model for autonomic intrusion detection system based on the autonomic loop, known as MAPE-K (Monitor, Analyze, Plan, Execute and Knowledge Base).

MohiuddinSolaimani et al. (2014) [9], they performed experiments on a real-time, Chi-square test based anomaly detection framework using Bigdata tool Apache Spark.

Tamer F. Ghanem et al. (2014) [10], they proposed a hybrid approach for anomaly detection in large datasets using genetic algorithms and multi-start meta-heuristic method. The results show that accuracy of 96.1% which is better than other machine learning algorithms.

Sachin Kumar et al. (2015) [11], proposed a framework for analyzing accident patterns for different types of

accidents on the road. They used K modes clustering and association rule mining algorithm for this analysis. Total 11,574 accidents were analyzed which occurred on Dehradun (Uttrakhand, India) during 2009 to 2014. Six clusters (C1–C6) were taken in consideration taking K=6. Association rule mining was applied on all 6 clusters to generate rules. Trend analysis results also support their method that performs clustering prior to analysis helps in detecting valuable results.

Michael A Hayes (2015) [12], was evaluated for two real-world sensor datasets provided by a Canadian company Brampton. The framework was also evaluated against the open-source Dodgers dataset and R statistical toolbox.The proposed work identifies a contextual anomaly detection framework. It detects content and context both. The content detector determines anomalies in real-time, identifying false positives. Juliette Dromard et al. (2015) [13], they proposed to take advantage of Hadoop and spark in order to speed up an Unsupervised Network Anomaly Detector Algorithm (UNADA). The experiments proved that execution time can be improved 13 times allowing UNADA for large datasets processing.This paper is good step for detecting network anomalies in real time on large non sampled traffic. Yafei Wu et al. (2015) [14], the original anomaly detection algorithm HOTSAX works in sequential manner in standalone machines with limited computing capabilities and storage. In this paper they have mitigated this problem by proposing distributed anomaly detection algorithm using apache spark computing platform and hadoop HDFS storage. By this approach, they have mitigated the low memory problems of their algorithm.

It was found that limited research was done on anomaly detection using bigdata tools like apache spark on large intrusion datasets.

3. Research methodology

The dataset UNSW-NB 15 is collected to experiment on spark tool.

3.1.Description of UNSW-NB 15 dataset

For the evaluation of performance and effectiveness of NIDS, we require a comprehensive dataset which contains both normal and abnormal behaviors. Lot of research has been done using older benchmark data sets like KDDCUP 99 and NSLKDD but these data sets do not offer realistic output performance. The reason is that KDDCUP 99 has lots of redundant and missing records in the training set. So these datasets are not comprehensive representation of modern low foot print attack environment. UNSW-NB 15 dataset was created by the IXIA PerfectStorm tool in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS). It contains both real modern normal activities and synthetic contemporary attack behaviors [15]. UNSW-NB15 dataset is available in comma-separated values(CSV) file format. There are 175,341 records in training set and 82,332 records in testing set with all different 9 types attack and normal records. There are 49 attributes or features with 10 class values in this dataset. All records are divided in two major categories of the records - normal and attack. The attack category is again subdivided into 9 categories of attack types. Attack types are Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms.

Table 1: List of attacks UNSW NB-15 dataset

Category	Training Set	Testing Set
Normal	56000	37000
Generic	40000	18871
Exploits	33393	11132
Fuzzers	18184	6062
DoS	12264	4089
Reconnaissance	10491	3496
Analysis	2000	677
Backdoor	1746	583
ShellCode	1133	378
Worms	130	44
Total Instances	1,75,341	82,332

We have performed experiments on 2 sets of UNSW datasets to evaluate the performance of all classifiers.These classifiers have been evaluated on Apache spark. Number of instances in dataset-1 is 7410 in training and 823 in testing. In dataset-2, instances for training are 47342 and 5260 instances are for testing.Big dataset is of approx. 602 MB. The training data is 397.32MB and test data is 198.66 MB used in Spark. Small dataset is 100 MB. Training data is 66 MB and test data is 33 MB.

Table 2: Distribution of normal and attack instances in Dataset-1

Name: attack_cat Missing: 0 (0%)	Instances:7410 Attributes: 45	Distinct:10	Name: attack_cat Missing: 0 (0%)	Instances: 823 Attributes: 45	Distinct: 10
No.	Label	Count	No.	Label	Count
1	Normal	3433	1	Normal	367
2	Reconnaissance	287	2	Reconnaissance	32
3	Backdoor	52	3	Backdoor	13
4	DoS	356	4	DoS	41
5	Exploits	1000	5	Exploits	113
6	Analysis	61	6	Analysis	10
7	Fuzzers	516	7	Fuzzers	68
8	Worms	5	8	Worms	1
9	Shellcode	37	9	Shellcode	5
10	Generic	1663	10	Generic	173
Training Set			Testing Set		

Table 3: Distribution of normal and attack instances in Dataset-2

Name: attack_cat Missing: 0 (0%)	Instances:47342 Attributes: 45	Distinct:10	Name: attack_cat Missing: 0 (0%)	Instances: 5260 Attributes: 45	Distinct: 10
No.	Label	Count	No.	Label	Count
1	Normal	15077	1	Normal	1654
2	Reconnaissance	2833	2	Reconnaissance	339
3	Backdoor	490	3	Backdoor	53
4	DoS	3222	4	DoS	372
5	Exploits	9017	5	Exploits	1038
6	Analysis	539	6	Analysis	78
7	Fuzzers	4972	7	Fuzzers	509
8	Worms	35	8	Worms	1
9	Shellcode	295	9	Shellcode	23
10	Generic	10862	10	Generic	
Training Set			Testing Set		

3.2. Proposed framework

The proposed framework for intrusion detection is shown in the fig no.1

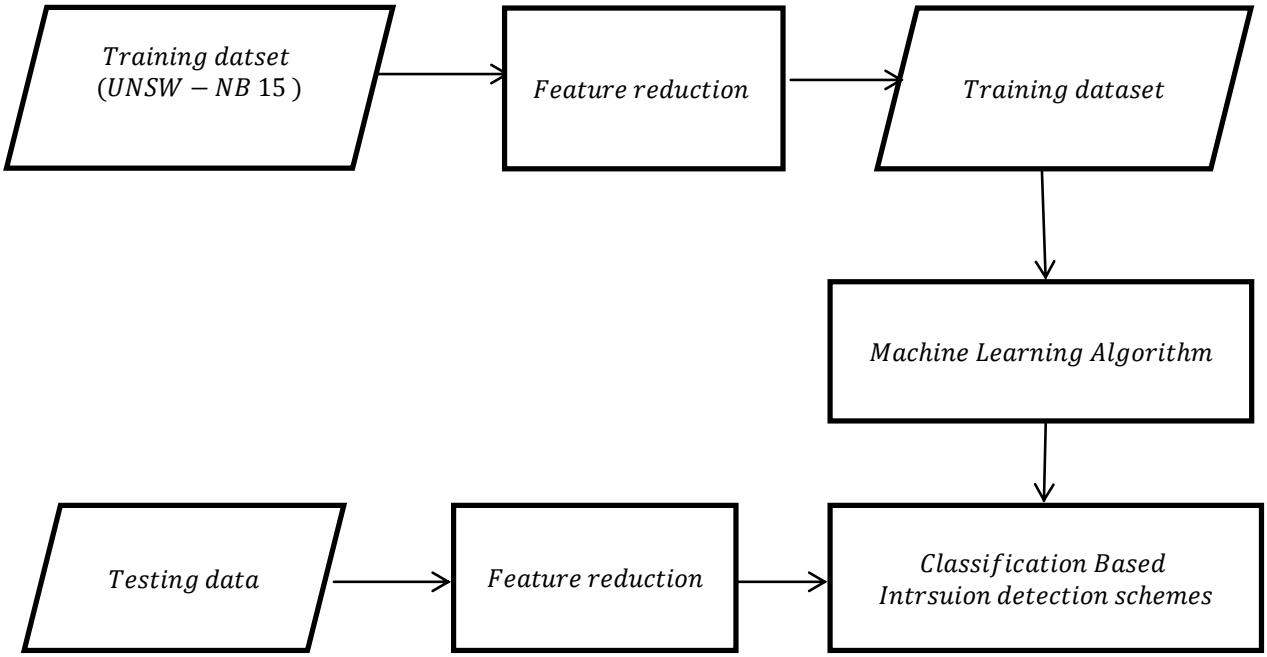

Fig. 1. Proposed framework for intrusion detection

3.3. Feature Reduction algorithms

In this section, we have used two feature reduction algorithms, namely, Canonical Correlation Analysis (CCA)and Linear Discriminant Analysis (LDA) that are used for evaluation of the proposed framework.

3.3.1.Canonical Correlation Analysis (CCA)

Canonical Correlation Analysis (CCA) connects two sets of variables by finding linear combinations of variables that maximally correlate. Two typical purposes of CCA are Data reduction and Data interpretations.Correlation basically tells the dependence among various attributes, so the attributes which are highly correlated, i.e. depended on other attributes can be removed. It saves a lot of time and gives the better results.

3.3.2. Linear Discriminant Analysis (LDA)

This method makes certain the maximal separately by making the most of the ratio of between-class variance to the within-class variance in any particular data. This method makes a decision region between the given classes; provide more class but not the location of the class. LDA provides the understanding of the scattering of the feature data. There are two methods to LDA, class dependent and class independent. However, here, the class dependent type as good discrimination is aimed.

3.4 Classification based Intrusion Detection Schemes

3.4.1. Naïve Bayes

Bayes' rule says that if you have a hypothesis H and evidence E that bears on that hypothesis, then

$$P(H|X)\frac{\mathrm{P(X|H)P(H)}}{\mathrm{P(X)}} \quad (1)$$

- P(H|X)is the posterior probability, or a posteriori probability, of H conditionedon X.
- P(H)is the prior probability.
- P(X|H) is the likelihood which is the probability of predictor given X.
- P(X)is the prior probability of X. "How are these probabilities estimated?" . P(H),P(X|H), and P(X) may be estimated from the given data.

3.4.2. Random Forest

Random Forest is used for the tasks: classification and regression. To reduce the over fitting risk it combine many of decision trees. The random forests can handle the definite features and does not require feature scaling.

Random Forest Algorithm: Random Forest is an ensemble method.Random forests can be built using bagging in tandem with randomattribute selection.Random forests are efficient on very large databases because it consider many fewer attributes for each split. They can be faster than either bagging or boosting. For each iteration, i (i =1, 2, : : : , l), a training set, of t tuples is sampled with replacement from T. That is, each training set is a bootstrap sample of T, so that some tuples may occur more than once in a training set, while others may be excluded. Let M be the number of attributes to be used to determine the split at each node, where M is much smaller than the number of available attributes. To construct a decision tree classifier, Fi , randomly select, at each node, M attributes as candidates for the split at the node. The trees are grown to maximum size and are not pruned.

4. Big Data Processing tools: Apache Spark

Apache Spark [16] is a tool for processing large amount of data. It is 100 times faster than MapReduce of Hadoop. Apache Spark has an advanced DAG execution engine that supports acyclic data flow and in-memory computing. It write applications quickly in Scala, Python, R and Java. It can be used with many other advanced languages. It can access different data sources from HBase, Cassandra, HDFS. Spark run be run with its standalone cluster mode, on Hadoop YARN, or on Apache Mesos or on EC2.The main components of Apache spark are of Spark core , SQL, Streaming, MLlib, GraphX. The key idea of spark is Resilient Distributed Datasets (RDD). Spark supports in-memory processing.

5. Result Analysis

5.1. Performance matrices

a) To measures for how "accurate" or superior your classifiers at predicting the class label of tuples. The classifier evaluation measures are accuracy (also known as recognition rate), sensitivity (or recall), specificity, precision, F1. The accuracy of a classifier on a given test set is the percentage of true positive and true negative from all correctly classifier.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

Here, TP=True Positive, TN = True Negative, FP= False Positive, FN= False Negative.

b) Recall (True Positive Rate). It is used to measures the proportion of attacks that are correctly identified as attack.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

c) Precision:Precision is the fraction of correctly classified attacks to all attack records.

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

d) Specificity (True Negative Rate). It is used to measures the proportions of not attacks that are correctly identified as not attacks.

$$Specificity = \frac{TN}{TN+FP} \quad (5)$$

e) Kappa: This is a measurement to find the performance of classifier as compared to how well it would have performed simply by chance.

f) Receiver Operating Characteristic (ROC) Curves: It is curve between TPR and FPR. Area under curve(AUC) gives the value of ROC. More the AUC and more will be the value of ROC.

$$False\ Positive\ Rate = \frac{FP}{FP+TN} \quad (6)$$

g) Training Time: measurement of time taken by a classifier to train a classifier.

5.2. Results

The results of smaller dataset-1 and larger dataset-2 are illustrated. Big dataset is of approx. 602 MB. The training data is 397.32MB and test data is 198.66 MB used in Spark. Small dataset is 100 MB. Training data is 66 MB and test data is 33 MB used in spark. Resultsare compared of small dataset and large dataset of UNSW NB-17datasets using Apache Spark. The performance of7different classifiers are evaluated on the basis of various parameters like accuracy, FPR, Training Time, precision, recall and ROC area.

Table 4: Performance evaluation using Canonical Correlation Analysis (CCA) based dimension reduction method on Dataset - 1 of (UNSW NB-15 dataset).

Classifier	Accuracy	Specificity	Kappa	Mean Abs. Error	FPR	Precision	Recall	ROC Area	Training Time (Sec)
Naïve Bayes	58.44	54.4	0.5408	0.0887	0.020	0.726	0.524	0.876	0.4
REP TREE	94.61	92.54	0.8242	0.0194	0.001	0.876	0.896	0.987	1.2
Random Tree	93.43	75.78	0.8941	0.0205	0.006	0.73	0.91	0.986	0.45
Random Forest	92.81	83.56	0.8569	0.0224	0.007	0.821	0.929	0.984	3.2
Random Committee	89.08	82.45	0.9134	0.0216	0.006	0.841	0.913	0.985	0.24
Bagging	95.53	79.57	0.8405	0.0234	0.01	0.816	0.906	0.984	1.16
Randomizable Filtered	87.76	92.5	0.7868	0.0213	0.01	0.89	0.912	0.985	0.88

It is clear from table 4 that when we used Canonical Correlation Analysis (CCA) based dimension reduction method on Dataset - 1 of (UNSW NB-15 dataset)then the Accuracy of REP Tree (94.61%), Random Tree (93.43%), Random Forest (92.81%) and Bagging (95.53%) algorithms was almost same. But Random Forest (3.2 Sec) and Bagging (1.16 Sec) took more training time so REP Tree (1.2 Sec) and Random Tree (0.45 Sec) were the winners in Accuracy and Training Time.

Table 5: Performance evaluation using Canonical Correlation Analysis (CCA) based dimension reduction method on Dataset - 2 of (UNSW NB-15 dataset).

Classifier	Accuracy	Specificity	Kappa	Mean Absolute Error	FPR(False Positive Rate)	Precision	Recall	ROC Area	Training Time (Sec)
Naïve Bayes	59.55	55.43	0.4373	0.0901	0.029	0.669	0.578	0.879	0.23
REP Tree	89.46	91.94	0.8343	0.0274	0.016	0.9	0.879	0.983	1.79
Random Tree	89.75	75.65	0.8342	0.0343	0.012	0.794	0.899	0.98	0.77
Random Forest	90.1	82.83	0.8655	0.0345	0.015	0.913	0.912	0.989	12.91
Random Committee	87.84	81.47	0.8365	0.0345	0.019	0.897	0.876	0.983	0.27
Bagging	88.45	78.56	0.8579	0.0310	0.02	0.879	0.889	0.982	1.55
Randomizable Filtered	87.10	91.3	0.8226	0.0325	0.019	0.898	0.882	0.985	1.02

It is clear from table 5 that when we used Canonical Correlation Analysis (CCA) based dimension reduction method on Dataset - 2 of (UNSW NB-15 dataset) then the performance of Random Tree (86.46%), J48 (86.17%) and Bagging (86%) algorithm decreased after reducing dimensions. But Random Forest (12.91 Sec) and bagging (1.55 Sec) took more training time so again Random Tree (0.77 Sec) was the winner in performance. This time Random Forest took 12.91 seconds time in training so we can say that Random Forest is slower to train when we use larger datasets.

It is clear from table 2 that when we used dataset-1 using spark then the Accuracy of Random Tree (93.01%) and Random Committee (93.08%) algorithms was equally high so both were the winners in Accuracy using spark.

Table 6: Classifier Evaluation Performance using LDA based Dimension Reduction on Dataset-1 of (UNSW NB-15 dataset).

Classifier	Accuracy	Specificity	Kappa	Mean Abs. Error	FPR	Precision	Recall	ROC Area	Training Time (Sec)
Naïve Bayes	57.93	55.43	0.4803	0.0881	0.014	0.767	0.559	0.899	0.9
REP TREE	88.12	92.93	0.8351	0.0345	0.024	0.869	0.871	0.984	2.54
Random Tree	92.5	74.66	0.8754	0.0307	0.016	0.903	0.902	0.991	1.4
Random Forest	87.75	82.38	0.8559	0.0325	0.018	0.889	0.887	0.984	3.6
Random Commite	92.16	80.43	0.877	0.0284	0.015	0.906	0.903	0.991	5
Bagging	89.73	76.56	0.8559	0.0325	0.018	0.889	0.887	0.984	19
Randomizable Filtered	83.02	92.32	0.7978	0.0447	0.025	0.85	0.84	0.978	6

Other side, Naïve Bayes is easy to train as it took just 0.8 Seconds but its accuracy is not good in our case. Naïve Bayes perform well on classification of textual data but our dataset was having more features with numerical data.
It is clear from table 3 that when we used based Dimension Reduction (LDA) on Dataset-1 of (UNSW NB-15 dataset) using spark then the Accuracy of Random Tree (92.5%) and Random Committee (92.16%) algorithms was equally high so both were the winners in Accuracy.

Table 7: Classifier Evaluation Performance using LDA based Dimension Reduction Dataset-2 of (UNSW NB-15 dataset).

Classifier	Accuracy	Kappa	Mean Absolute Error	FPR	Precision	Recall	ROC Area	Training Time
Naïve Bayes	55.5	0.4768	0.0893	0.013	0.763	0.555	0.896	1.76
REP Tree	93.56	0.7879	0.0335	0.023	0.833	0.832	0.905	7.92
Random Tree	86.46	0.827	0.0335	0.025	0.861	0.865	0.974	2.55
Random Forest	84.3	0.8018	0.0366	0.021	0.842	0.843	0.961	5.74
Random Committee	93.26	0.8007	0.0353	0.02	0.842	0.842	0.925	7.98
Bagging	86	0.822	0.0348	0.024	0.862	0.86	0.977	60.92
Randomizable Filtered	75.93	0.6955	0.0481	0.04	0.76	0.769	0.86	5.4

It is clear from table 7 that when we used dataset-2 Dimension Reduction (LDA) on Dataset-2using spark then the Accuracy of Random Tree (93.56%) and Random Committee (93.26%) algorithms was equally high in accuracy and training time is also less, so both are the winners.

6. Conclusion

The paper proposed framework was fast and effect for intrusion detection.We have used small and large dataset of UNSW NB-15 dataset for performance evaluation of the proposed framework by applying different feature extraction and classification algorithms.It is found that using CNN affects the accuracy on small dataset-1. But time taken is also reduced. But using the LDA not only increases the accuracy but also the time taken is not increase by large margin to train the data in case of small as well as large dataset. Random Tree are the winners based on various performance parameters and LDA is better feature reduction technique.

It is found that anomaly detection approach is more effective and fast using LDA and random tree algorithm outperforms. The accuracy of random tree algorithm is better than other algorithms. This approach properly classifies the data either as normal and various attacks. Accuracy is also improved by using feature reduction methods. It can be concluded that this approach is better, faster and more efficient when used on apache spark.

References

[1] Niva Das, Tanmoy Sarkar (2014) "Survey on Host and Network Based Intrusion Detection System." *: An International Journalon Advanced Networking and Applications*.

[2] Abhinav S. Raut, Kavita R. Singh (2014) "Anomaly Based Intrusion Detection-A Review"*An International Journal on Network Security*, Vol. 5,

[3] Sanjai Veetil, Dalhousie University (2013)"A Real-time Intrusion Detection System by Integrating Hadoop and Naive Bayes Classification," DSCI.

[4] Xun-Yi Ren and Yu-Zhu Qi (2013) "Hadoop-based Multi-classification Fusion for Intrusion Detection,"*Journal of applied science*.

[5] Junlong Xiang , Magnus Westerlund, Dušan Sovilj,Göran Pulkkis (2014) "Using Extreme Learning Machine for Intrusion Detection in a Big Data Environment"IEEE.

[6] Samuel Marchal, Xiuyan Jiangz, Radu State, Thomas Engel (2014) "A Big Data Architecture for Large Scale Security Monitoring" , *Springer*.

[7] Sung-Hwan Ahn, Nam-Uk Kim,Tai-Myoung Chung (2014) "Big Data Analysis System Concept for Detecting Unknown Attacks", IEEE.

[8] Kleber M.M. Vieira, Fernando Schubert, Guilherme, A. Geronimo, Rafael de Souza Mendes, Carlos B. Westphall(2014) "Autonomic Intrusion Detection System in Cloud Computing with Big Data", *Conference: The 2014 International Conference on Security and Management.*

[9] Mohiuddin Solaimani, Mohammed Iftekhar, Latifur Khan, Bhavani,Thuraisingham (2014) "Statistical Technique for Online Anomaly Detection Using Spark Over Heterogeneous Data from Multi-source VMware Performance Data," *IEEE*.

[10] Tamer F. Ghanem a,*, Wail S. Elkilani b, Hatem M.Abdul-kader(2014) "A hybrid approach for efficient anomaly detection using metaheuristic methods"*Journal of Advanced Research*.

[11] Sachin Kumar and Durga Toshniwal (2015) "A data mining framework to analyze road accident data"*Springer*.

[12] Michael A Hayes and Miriam AM Capretz (2015) "Contextual anomaly detection framework for big sensor data"*Springer*.

[13] Juliette Dromard, Gilles Roudi_ere, Philippe Owezarski (2015) "Unsupervised Network Anomaly Detection in Real-Time on Big Data," *Springer*.

[14] Yafei Wu, Yongxin Zhu, Tian Huang (2015) "Distributed Discord Discovery: Spark BasedAnomaly Detection in Time Series,"*IEEE*.

[15] Nour Moustafa, Jill Slay (2016) "UNSW-NB15: A Comprehensive Data set for Network Intrusion Detection systems.*" IEEE, Information Security Journal: A Global Perspective, Volume 25, Issue 1-3, 2016.*

[16] Apache Spark™ - Lightning-Fast Cluster Computing, http:// spark.apache.org/

[17] Trevor hastie, Robert tibshirani, Jerome friedman(2008.),"The elements of statistical learning ", Springer.