IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

Intrusion Detection in Fast and Streaming Network Data using In-Memory Stream Processing in Spark

N. Pramila[1], G. Ravi[2]

Research Scholar, Department of Computer Science, Jamal Mohamed College (Autonomous), Tiruchirappalli, Tamilnadu, India[1]

Associate Professor & Head, Department of Computer Science, Jamal Mohamed College (Autonomous), Tiruchirappalli, Tamilnadu, India[2]

Abstract: Due to the increased digitalization of information, a huge amount of data is being generated. Information richness in such data has attracted researchers to this data. The major problem existing in real time data is that it is fast and streaming making analysis on them difficult. Intrusion detection is a continuous process and depending on the size of the network and the number of transmissions being carried out in the network, the number of packets to be analyzed varies considerably. The packets being transferred tends to be fast, hence a mechanism to provide analysis in real time becomes mandatory. This paper presents a tree based technique to analyze network traffic and provide real time predictions with higher accuracy. It uses an ensemble of trees called the Random Forest classifier. Experiments were conducted on Hadoop platform using Spark. Spark, being a stream processing framework exhibits effective results in real-time.

Keywords: Classification; Anomaly Detection; Network Intrusion Detection; Hadoop; Spark; Random Forest.

I. INTRODUCTION

Complexity and size of the networks has been increased to a very large extent due to the high levels of interconnectivity options. This seems to be a promising scenario; however, it also has a downside. The more information is transferred online; the information becomes more vulnerable and has high probabilities of being exploited using vulnerabilities in the network. Intrusion detection in a network is one of the most important requirements of any administration system in a network. Current years have seen a huge growth in networking technologies, attacks occurring in networks and also attack avoidance techniques. Several techniques have been developed for counteracting intrusions, but new types of intrusions keep appearing every day.

The online crimes in general take the form of attacking a target system directly or stealing information during online transactions. In either of the type, a computer forms the base of the attack and this system is called the compromised node. Detecting these compromised nodes is a very important issue in intrusion detection. The compromised node has the ability to perform malicious activities like sniffing of packets, performing Denial of Service (DoS) attacks, transmitting viruses/worms and much worse, convert other computers into compromised nodes. All other systems within the network become vulnerable to attacks due to the presence of a compromised node.

Hence it becomes mandatory to black list these nodes and either remove them from the network or monitor its activities for malicious behavior and restore the system to its initial state.

Not only the intrusions, but also the methods to compromise these intrusions have shown a huge increase. Data mining algorithms such as Clustering, Classification and Association Rule Mining and several other variations of such algorithms have proved to be much useful in the case of detecting Intrusions.

One major problem in an IDS is that it also has high probability of generating false positive results. False positives refer to marking a legitimate or correct transmission as an intrusion. This in turn leads to blocking the transmission. The expectation of a good intrusion detection system is not only to detect intrusions but also to provide the least false positive rate.

II. RELATED WORKS

Pedro et al. proposed a study in which a summary of the existing network intrusion detection techniques have been discussed [1]. This paper also lists the available IDS systems that perform the required task, and also the major challenges in establishing the process. An organization of the related techniques and their behavioural nature are presented in detail.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

Dasgupta et al. presented an agent based intrusion detection system in [2]. It has been proposed to detect malfunctions, abnormal traffic, faults, intrusions, deviations, and also claims to provide appropriate recommendations for the user. The major advantage of the presented IDS approach is that it can simultaneously monitor the network activities at multiple levels, and hence can effectively provide correlation among the deviated values and security violations corresponding to the intrusions. Agents were implemented using the Cougaar framework, and each node is imposed with four agents for operating on the data. CIDS (Cougaar based Intrusion Detection System) has a modular design and is dynamic. Hence it has the flexibility to incorporate new detection, action and decision rules. It uses a swing based GUI which is used for monitoring network traffic. This technique has been proposed to majorly counteract probing attacks and DoS attacks.

Using an intrusion detection system that is accurate and fast is the current requirement. The ID method to be used should be faster such that the results returned by an IDS is actually usable in real time. Teodoro et al. [3], Sheyner et al. [5] and Ammann et al. [4] presented an attack graph based prediction system that operates in linear time with quadratic space requirements. A queue graph (QG) based approach is presented by this model, and uses the latest alerts for processing. This leads to the usage of only the recent alerts, hence all the past alerts need not be examined. This reduces the amount of processing to a large extent. Further, it helps to correlate alerts that are arbitrarily far away, hence can help predict intrusions effectively. A similar technique using Apriori for fraud detection is presented in [10]. A combination technique that uses PCA and PSO for detecting intrusions is presented in [11]. Other ensemble bsed techniques include [12-15].

Huang et al. presented a parallel Intrusion Detection mechanism in [6] using GPUs. It converts a commodity GPU hardware into a powerful pattern matching processor. Parallelized packet inspection is performed to provide a faster method for detecting intrusions. Giorgos et al. presented a regular expression based IDS using GPUs in [7]. The highly comparative nature of the regular expressions presents an overhead to the system. This requires frequent access to CPU and memory. GPUs tend to reduce this overhead by performing the tasks in parallel. This mode of operations claims to produce an increased speed up of about 48 times when compared to regular systems.

Liberios et al. presents a GPU based IDS, Suricata that provides active protection and improved network security [8]. Suricata performs effective traffic analysis. The implementations are carried out using the CUDA architecture (GeFORCE GTX 260). This technique also uses rule based computations. A similar method which uses a 12 core machine with 2 GPUs was proposed by Jamshed et al. in [9]. This method was named the Kargus claims to exploit the full potential of commodity hardware. Kargus was designed as a batch processing system and it claims to adapt its resource usage depending on the input rate. This leads to an excellent energy conservation mechanism in Kargus.

III.OUR APPROACH

The enhanced tree based intrusion detection system uses an ensemble of decision trees to perform classification. The intrusion detection data is initially segregated to two sections, divided in the ratio 7:3. These represent the training and the test data respectively. The training data is subject to the random forest algorithm.

This algorithm is a combination of tree predictors, and each tree depends upon the random vector generated. The strength of the individual trees and their correlation determines the generalization error. Map Reduce and spark based implementations of the Random Forest algorithm is used in a Hadoop cluster. Sequential Random Forest algorithm tends to take more time, due to the redundancy requirement of building trees.

Since building each tree is an independent operation, the basic algorithm is embarrassingly parallel in nature. This has made the Random Forest Classifier, the best candidate for the current classification process. The test data is passed to the trained classifier model and the accuracy is determined.

Size of the ensemble classifier is determined and the training data is split accordingly, such that every decision tree in the classifier is provided with at least 66% of the training data. The reason for such division is that every class should contain their representations in each decision tree of the ensemble classifier in order to perform efficient classification.

Each decision tree identifies a subset of m predictor variables from a list of M total predictors (m<M). The best predictor variable is identified from the set of m values and a split is performed on it.

This process is repeated for all the predictor variables and a decision tree is constructed. Pruning is avoided in order to reduce information loss. Every decision tree operates on the data provided to them, hence if the ensemble contains k independent decision trees, k different rules are finally generated.

Since a subset of data was used for training, none of the decision rules obtained at this stage are complete. These rules obtained at the intermediate stage are referred to as weak rules. All the generated rules are aggregated to obtain the final decision tree, called the strong classifier. This method works on the basic principle that several weak classifiers can be combined to form a strong classifier.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

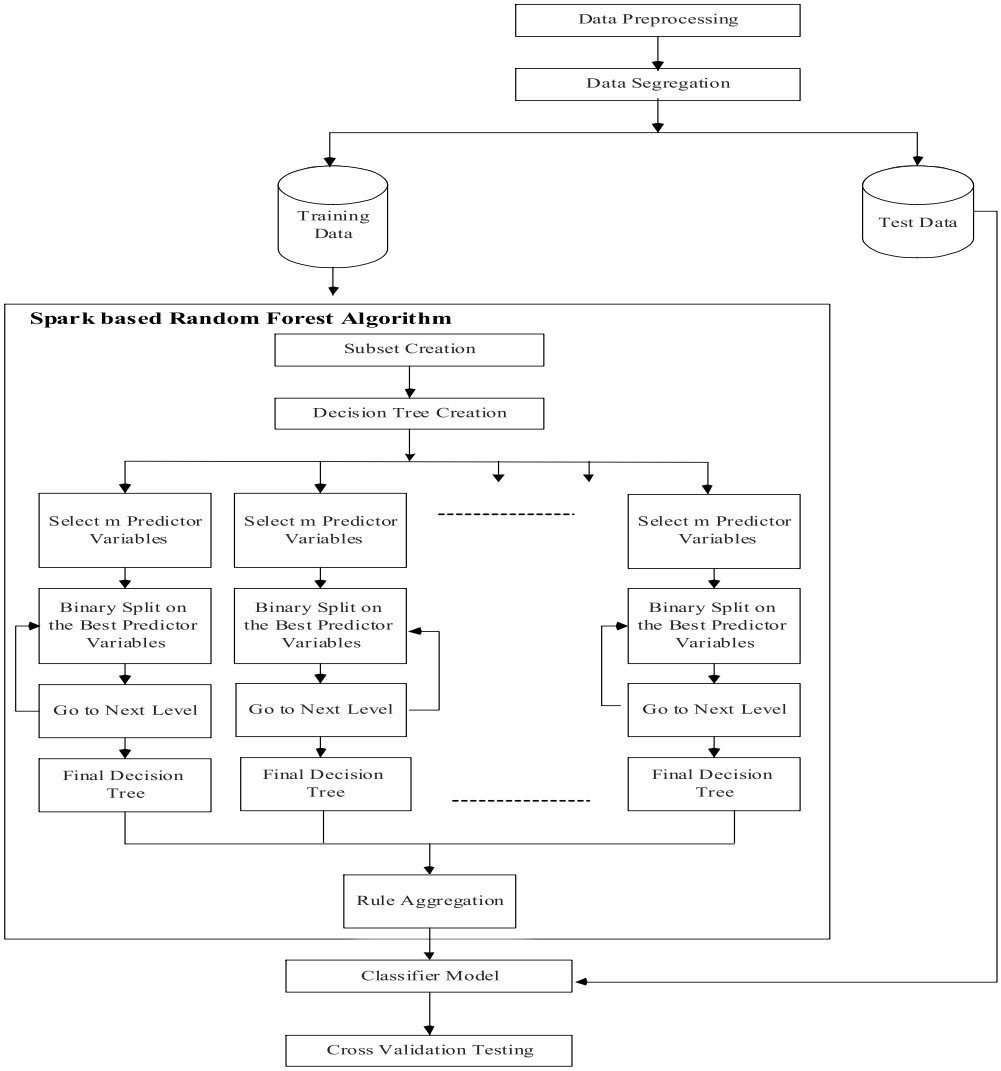

Fig. 1. Spark Based Random Forest Intrusion Detection: Architecture

IV. RESULTS AND DISCUSSION

Experiments were carried out in Cloudera VM using Hadoop 2 and Spark 1.4. Experiments were carried out on the KDD Cup 99 dataset and the results obtained from the random forest classifier was recorded and analyzed.

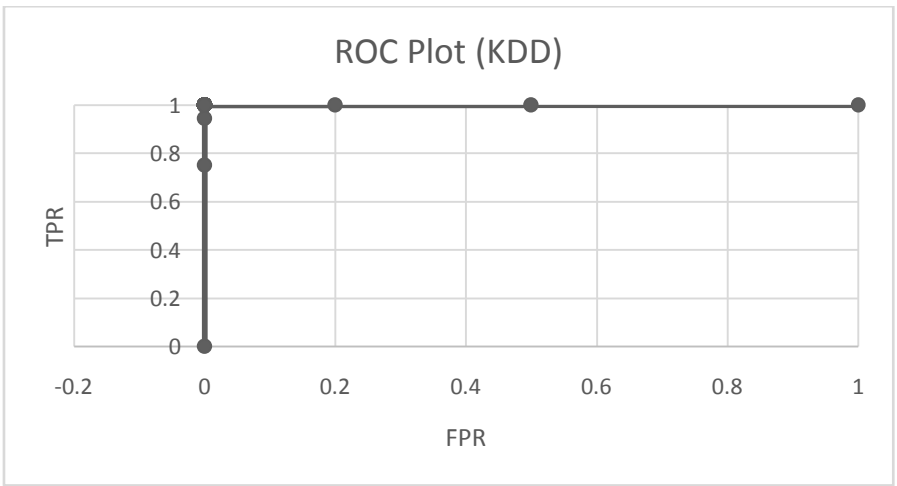

Fig.1 ROC Plot (KDD)

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

The ROC plot for KDD Cup 99 dataset is presented in Figure. It could be observed that most of the points are concentrated towards the top left, indicating the efficiency of operation of the algorithm.

Similarly, the PR plots (Figure) shows their concentration towards the top right indicating high precision and recall levels. This exhibits the efficiency of the working algorithm.

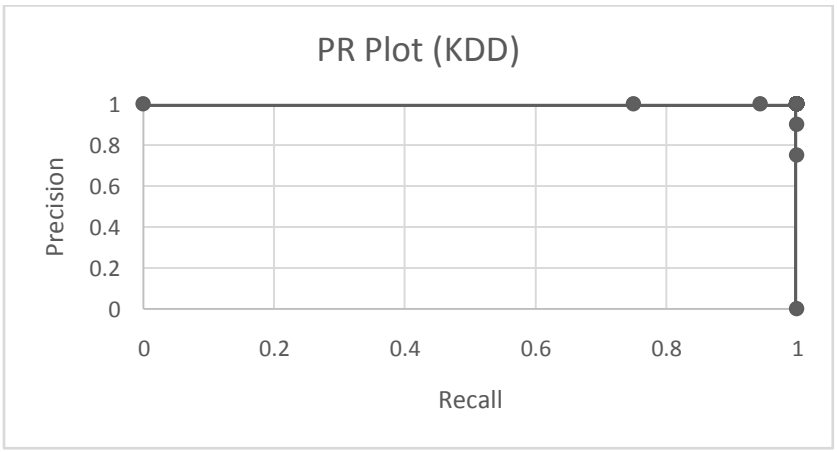

Fig. 2 PR Plot (KDD)

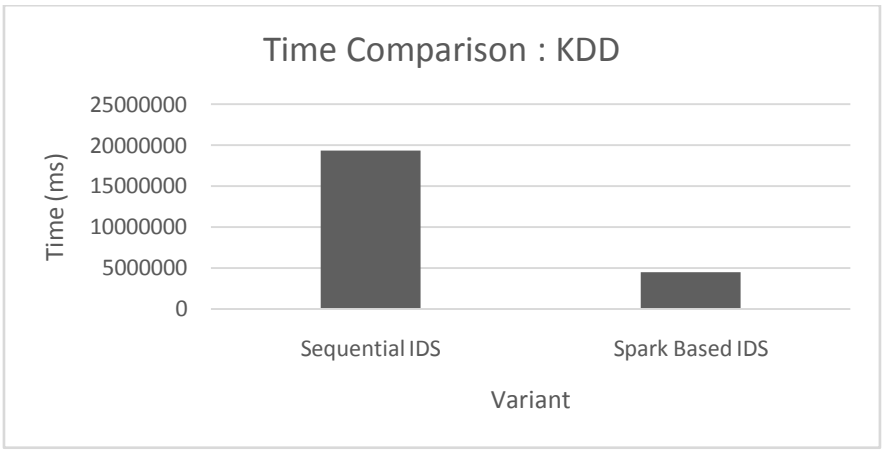

Fig 3 Time Comparison: KDD

A comparison is carried out on the basis of time taken between an intrusion detection system based on random forest, executed in a sequential environment and in parallel environment (Spark). It could be observed from the figure that the spark based classifier operates with 3X speed compared to its sequential counterpart. This proves the efficiency of the algorithm when working on real-time data.

V. CONCLUSION

The increase in ecommerce trades has made online intrusion detection a major necessity for any financial accessing system. This paper proposes an effective intrusion detection technique which also has the capability of handling Big Data. The proposed technique uses several Decision Trees that are made to execute in-parallel, forming a Random Forest. The Random Forest algorithm is considered as an ensemble of several decision trees to build the classifier rules for the problem. This makes the algorithm immune to imbalance and missing data.

The Random Forest algorithm was tested on datasets containing low to moderate imbalance and was found to scale well both in terms of accuracy and reliability. Future enhancements of this algorithm would include testing the algorithm's scalability level on datasets with huge imbalance.

REFERENCES

[1] G.T.Pedro, et al. "Anomaly-based network intrusion detection: Techniques, systems and challenges." Computers & security 28.1: 18-28, 2009.
[2] D. Dasgupta, et al. "CIDS: An agent-based intrusion detection system." Computers & Security 24.5: 387-398, 2005.
[3] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges." computers & security, 28(1), pp.18-28, 2009.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

[4] P. Ammann, D. Wijesekera, S. Kaushik. "Scalable, graph-based network vulnerability analysis," in: Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS'02), pp. 217–224, 2002.
[5] O. Sheyner, J. Haines, S. Jha, R. Lippmann, J.M. Wing. "Automated generation and analysis of attack graphs," in: Proceedings of the 2002 IEEE Symposium on Security and Privacy (S& P'02), pp.273–284, 2002.
[6] N.F. Huang, et al. "A GPU-based multiple-pattern matching algorithm for network intrusion detection systems." Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference on. IEEE, 2008.
[7] V. Giorgos, et al. "Regular expression matching on graphics hardware for intrusion detection." Recent Advances in Intrusion Detection. Springer Berlin Heidelberg, 2009.
[8] V. Liberios, A. Baláž, and BranislavMadoš. "Intrusion detection architecture utilizing graphics processors." Acta Informatica Pragensia 1.1, 50-59, 2013.
[9] M.A. Jamshed, et al. "Kargus: a highly-scalable software-based intrusion detection system." Proceedings of the 2012 ACM conference on Computer and communications security. ACM, 2012.
[10] K. Abdullah and A. Sami, "SysDetect: A systematic approach to critical state determination for Industrial Intrusion Detection Systems using Apriori algorithm." Journal of Process Control, 2015.
[11] W. Hui, G. Zhang, E. Mingjie, and N. Sun, "A novel intrusion detection method based on improved SVM by combining PCA and PSO." Wuhan University Journal of Natural Sciences 16, no. 5: 409-413, 2011.
[12] Shittu, Riyanat, Healing, A., Ghanea-Hercock, R., Bloomfield, R. and Rajarajan, M. "Intrusion alert prioritisation and attack detection using post-correlation analysis," Computers & Security 50: 1-15, 2015.
[13] L.Wei-Chao, S. Ke, and C. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," Knowledge-Based Systems 78: 13-21, 2015.
[14] C. Jaeik, T. Shon, K.Choi, and J. Moon, "Dynamic learning model update of hybrid-classifiers for intrusion detection." The Journal of Supercomputing 64, no. 2: 522-526, 2013.
[15] P.S. Nilkanth, and R.S. Bichkar, "Genetic algorithm with variable length chromosomes for network intrusion detection." International Journal of Automation and Computing: 1-6, 2015.