Hadoop Based Real-time Intrusion Detection for High-speed Networks

M. Mazhar Rathore, Anand Paul*, Awais Ahmad
The School of Computer Science and Engineering
Kyungpook National University, Daegu, Korea
rathoremazhar@gmail.com, *paul.editor@gmail.com,
awais@knu.ac.kr

Seungmin Rho
Department of Media Software
Sungkyul University, Anyang, Korea
smrho@sungkyul.edu

Muhammad Imran
College of Computer and Information Sciences,
King Saud University, Saudi Arabia
dr.m.imran@ieee.org

Mohsen Guizani
Department of Electrical and Computer Engineering,
University of Idaho, USA.
mguizani@ieee.org

***Abstract*—The rate of data generation is enormously growing due to the number of internet users and its speed. This increases the possibility of intrusions causing serious financial damage. Detecting the intruders in such high-speed data networks is a challenging task. Therefore, in this paper, we present a high-speed Intrusion Detection System (IDS), capable of working in Big Data environment. The system design contains four layers, consisting of capturing layer, filtration and load balancing layer, processing layer, and the decision-making layer. Nine best parameters are selected for intruder flows classification using FSR and BER, as well as by analyzing the DARPA datasets. Among various machine learning approaches, the proposed system performs well on REPTree and J48 using the proposed features. The system evaluation and comparison results show that the system has better efficiency and accuracy as compare to existing systems with the overall 99.9 % true positive and less than 0.001 % false positive using REPTree.**

Keywords—Machine Learning; Intrusion Detection; Network Threats, Big Data

I. INTRODUCTION

In this technological era, the internet speed has been crossing the limit of gigabytes and even into terabytes. The people from various domains, even with the lack of enough computer knowledge, are getting benefits by using internet services. The companies are gaining profit by managing their resources and transactions over the network. Thus, the possibility of cyber-attacks by stealing the personal and secret information is also increasing at the same rate. These threats might be the intrusion into the system, which can be defined as any illegal computer activity to get access for information gathering, eavesdropping, etc., passively or by doing harmful packet forwarding, packet dropping, or performing hole-attack, etc.

Even though, the IDS concept came very early in 1980's but, it is still a significant topic for researchers due to the continuous evolution and change in the structure of data, speed of networks, and the changing adaptation techniques of the intruders. The Technological advancement in cyber space increases the usage of ubiquitous networks, wireless sensors networks, and web technologies. Thus, the abundant use of technology results in an exponential increase in the network data traffic. According to one of the reports, 65% of UK houses were connected to the internet in 2008 [1], and it was increased to 80% in 2012 [2]. Moreover, in 2012 the overall computer generated data was estimated as 2.27 zettabytes and expected 8 zettabytes in 2015 [3] in which more than 90% contents were generated in last two years [4]. While, on the internet, this data is transmitted at a very high speed in various varieties. Therefore, in order to provide the safety from intruders in such high-speed data environment, an efficient system is needed, which keeps the requirement of handling high velocity of data in consideration while monitoring and analyzing network traffic at real-time. Such type of high volume data with high velocity, and of a different variety is usually termed as Big Data, which can be structured, semi-structured, and unstructured. In the era of Big Data, the IDS is efficient enough to process high-speed transmission at real-time without losing any vital flow packets.

To address the aforementioned challenges, the proposed system meets the need of efficiency with higher accuracy while running continuously in a parallel environment of Hadoop and introducing no extra overhead that degrades the performance of the transmission. The contribution of the proposed scheme is manifold, i.e., i) Hadoop-based architecture is proposed for IDS that detects any kind of threats accurately, efficiently, and with high speed using machine learning (ML) classifiers, ii) Feature selection mechanism is proposed that selects the nine best features among 41 features of KDDCUP99 dataset to detect abnormal behaviors in the network, iii) implementation of the

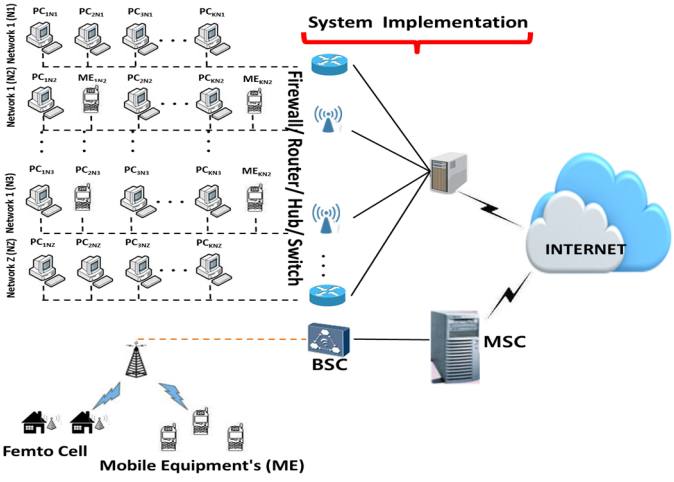

Fig. 1. Overview of the proposed system

978-1-5090-1328-9/16/$31.00 ©2016 IEEE

proposed IDS using Apache Spark on top of the Hadoop ecosystem in order to get real-time efficiency, and iv) Evaluation of the proposed method using various machine learning techniques on various intrusion datasets. The proposed system is compared with existing techniques with respect to accuracy and efficiency. The proposed system has higher accuracy and more efficient than existing systems. Therefore, it has the capability to work in high-speed Big Data environment due to its obvious advantages over the traditional system. The system can be implemented by capturing traffic from either switch, router, gateway or any other high-speed network device with high-speed capturing card. Abstract level model of the system is shown in Figure 1.

Rest of the paper is organized as follows. The related work including the existing IDS techniques and the background is presented in Section II. Details related to the proposed system are given in Section III. Section IV elaborates the implementation details, system evaluation, and comparison. Finally, the last section gives the concluding remarks.

II. RELATED WORK

Various types of techniques have been proposed for intrusion detection, such as rule-based, statistical analysis-based, machine learning-based, misused-based. These techniques can be host-based, network-based detection, or hybrid. A rule-based technique is proposed in [5] that is based on known ratio propagation model by describing power decay of message transmission rule. Puttini et al. [6] proposed Bayesian classification statistical method that is used to detect intrusion. Their main aim is to detect packet flooding that results into DoS. In [7], the estimated congestion at intermediary nodes is used as a decision-making mechanism to detect malicious behavior that caused packet dropping. The IDSs proposed in [8, 9, and 10] uses ML methods and classifiers to detect intrusion. They used kdd99cup datasets and introduced various parameters for ML classifiers to detect various attacks. Abbes et al. [11] also used ML approach for active IDS by analyzing different application protocols by defining a distinct adaptive decision trees for each protocol that to identify DoS attack, scans attack, and botnets. Wagner et al. [12], Khan et al. [13], and Wagner et al. [14] used support vector machine (SVM) for intrusion classification. Misuse-based detection can be either signature-based or rule-based. Using this type of techniques, signatures or patterns of the existing attacks are detected and then used for future profile e.g., "more than five attempts to sign in but failed". On the other hand, the authors identified some rule-based detection by identifying some rule for intrusion detection, such as interval rule, retransmission, integrity rule, delay rule, repetition rule, radio transmission range, etc. [15].

Network-based IDS monitored and made an analysis on each incoming packet of the network traffic and identified intrusions on the network. Francisco et al. [16] proposed Network Intrusion Detection System (NIDS) for the smart-sensor-inspired device. Similarly, specification-based technique, which is mainly concerned with detection attacks, such as DoS, replay attacks as well as compromised node in distance-vector routing protocols, such as DSDV protocol [17]. Similarly, host-based identifies any intrusion activity on a single node as a result of any event, such as changes in critical system files on the host, repeated failure access attempts to the host, unusual process memory allocations, etc. One of the host-based anomaly detection ADMIT is done by Sequeira and Zaki [18] by creating user profiles of a sequence of user or computers commands. Few IDS techniques are hybrid that use both network-based and host-based features [19] to detect threats.

Few researchers worked on the development of intrusion detection system in Big Data environment using Hadoop ecosystem. Prathibha and Dileesh [20] proposed hybrid intrusion detection system using Snort rules on Hadoop. However, neither they provided complete implementation details nor the accuracy and efficiency results. Similarly, Bandre and Nandimath [21] also provided the design consideration of Network Intrusion Detection System (NIDS) based on the Hadoop framework. They used General Purpose Graphical Processing Unit (GPGPU) to accelerate the performance. Even though, they only used seven parameters for IDS, but, on the other hand, they also did not provide the accuracy of the system. Their work only considered three attacks i.e., Denial of Services (DoS), Tcp_Syn, and ArpPoi. Jeong et al.[22] discuss few other works done by using Hadoop systems.

Even though there are various kinds of techniques proposed in the literature but most of them are still not efficient enough to process high-speed Big Data at a real-time. Others have a lack of accuracy. Therefore, based on the previous ML knowledge, the proposed system detects intrusions using the selected nine features with higher accuracy and efficiency.

III. PROPOSED SYSTEM

A. Datasets, tool, and Experiment Environment

Most widely used intrusion datasets are used as benchmarks in order to analyze the specific features of the intruder flows and for testing. We used DARPA dataset [23] as basic analysis datasets of size 5.5GB, in the form of network traffic, which contains various complex intrusions including probing, breaking into the system by exploiting vulnerabilities, installing DDoS software for the compromised system, and launching DDoS attack against another target. The KDDCUP99 [24] is the simplified, structured, labeled form of the DARPA dataset, characterized by 41 parameters. Therefore, we used CDDCUP99 dataset, of size 1.2GB, for system testing and evaluation. The dataset contains 24 specific types of intrusions in training files and extra 14 attacks added in testing dataset including DoS attack, remote to local attack (R2L), user to root attack (U2R) and probing attacks. In addition to these datasets, a small 40 MB dataset i.e., NSL-KDD dataset [25], is also used for testing purpose. NSL-KDD dataset resolved the issues from the KDDCUP dataset by removing the redundant and duplicate records by making it more reliable for the security analysts and researchers.

We implemented the system on Hadoop ecosystem in a single node environment on Ubuntu 14.04 LTS system with 4 GB RAM and core i5-3.20 GHz processor. However, For Heap usage in ML algorithms, only 2 GB RAM is allocated.

B. Features and Parameters Selection

KDD99 [24] dataset classify the intrusion by 41 flow parameters. But, to calculate 41 parameters for each flow in high-speed networks is impractical because of computational complexity. In addition, this list of parameters may consist of many irrelevant parameters, which cause a reduction in accuracy. Few techniques have been proposed to reduce the parameters list by selecting the significant parameters. One of

the techniques is proposed by Aljarrah called RF-FSR and RF-BER [26] that selects best 16 features among 41 of KDD features. Similarly, Kayacik [8] and Araujo [9] uses their reduction techniques to reduce this number to 15 and 14 respectively. However, still, the number is large to handle at real-time in a high-speed environment. Finally, Kantor [10] reduced the list to a minimum level at 6 features among those 41 features. Although the number is perfect in order to process high-speed Big Data. On the other hand, the accuracy of the system is compromised, especially in the case of any unknown future attack. There should be an equilibrium between the efficiency and accuracy. Therefore, we should select best minimum possible parameter list so that the accuracy and efficiency should be more. Considering this requirement, we used forward selection ranking (FSR) and backward elimination ranking (BER) [26] mechanism to select 6 best feature among 41 of them including features 1, 2, 3 and 16 and instead of parameters 6, 7 i.e., src_bytes, dst_bytes, we prefer to use the "number of packets" and "packet size mean". Furthermore, we observed from DARPA TCPDump traffic analysis that the packet size distribution of normal traffic and malicious flows for some applications differs. Therefore, three more features i.e., pkt_rate, pkt_sd_size, and range_pkt_size are added in the proposed parameter list. In FSR or BER parameter selections, the weight calculation is a key factor, which shows the importance of each parameter in the classification process. In our case, we used Enhanced Support Vector Decision Function (ESVDF) [27] to calculate the weights of all forty-one parameters. Afterward, the parameters list is sorted depending on their weights using Random Forest [28]. Using FSR, at start, only two features with highest weights are taken to form a set called Selected Features Set (SFS). Then, the current SFS is used for Classification model building and testing. The set in the current state is used for Intrusion classification using Random forest and evaluated with respect to efficiency and accuracy. Afterward, the parameters are added one by one in the SFS depending upon their weights and again the set is evaluated. In case, the newly added feature enhances the accuracy and efficiency performance, it is withheld in the SFS, otherwise, it is removed from SFS. While selecting with BER, initially, the parameter selection is performed in reverse by adding all 41 parameters in the SFS. Then they are removed one by one depending on the minimum weight. In case, the removal of the parameter reduces the performance, that parameter is again kept into the set. Finally, we used both FSR and BER working together to choose the best 4-6 features amongst all parameters. At the end, we finalized nine parameters including the BER and FSR selected parameters as well as the analysis-based features. The complete list of parameters with their details is shown in Table 1. These nine parameters are used with the ML classifier to detect the intruder's flows.

C. *Proposed Architecture*

The main objective of the proposed system is to process network traffic at real-time for intrusion detection with higher accuracy in the high-speed Big Data environment. Keeping in mind the objective of the system, the proposed architecture is designed, which can be implemented at any network device, such as at switch, router, and even at ISPs and telecommunication authorities' gateways and firewalls. Initially, the traffic is captured at the above mentions high-speed network with high-speed capturing device and drivers, such as RF_RING and TNAPI [29] so that no packet can remain uncaptured. The captured traffic is sent to the next layer filtration and loads balancing server (FLBS). FLBS has two primary responsibilities. First, it filters only those flows' traffic for analysis, which are not yet decided as an intrusion or normal flows by efficient searching and comparisons in In-Memory intruder's database. Secondly, it sent the unidentified flows traffic and required packet header information to the 3rd layer (Hadoop layer) master servers. FLBS also balance the load by deciding which packets are sent to which master server depending on the IP addresses. Master takes the network traffic/packets and generates sequence file for each flow so that it can be processed by Hadoop data nodes. Master node extracts necessary information from each packet by using Pcap-Input Format, Hadoop-pcap-lib and Hadoop-pcap-serde APIs and stores that information into sequence file such that each packet corresponds to one line. The process continues for a particular duration for each flow. Afterward, the sequence file is sent to data nodes which are equipped with feature value calculation algorithm implemented in MapReduce. The MapReduce code of the algorithm calculates the network flow feature by processing sequence file line by line in parallel. Moreover, in order to perform real-time analysis, the Apache Spark is used as 3rd party tool with Hadoop ecosystem. Finally, the features values are sent to layer 4 decision server(s). Decision server(s) has the implementations of the various classifiers, such as J48, REPTree, SVM, etc., which classifies the flows as normal or attack based on their parameters values. Memory intrusion list

Finally, decisions about a particular flow are stored in In-

TABLE 1. SELECTED FEATURES OF THE PROPOSED IDS

S #	Features	Details
1	Duration	Whole duration of the flow/session
2	protocol	Protocol
3	Service	Particular service the host is using
4	Num_root	Number of roots involved
5	No. of packets	No. of packets
6	Pkt_rate	Packet rate in packet/second for a particular flow
7	Pkt_size_mean	Mean value of the packet sizes
8	Pkt_sd_size	Pkr sizes standard deviation
9	Range_pkt_size	Range of the packet sizes.

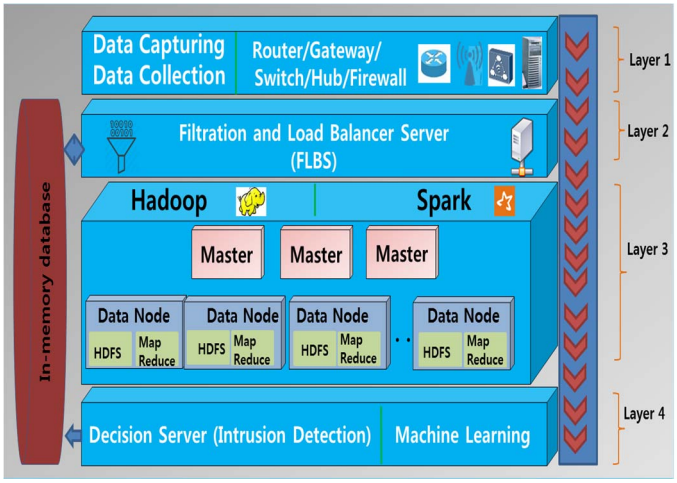

Fig. 2. The proposed intrusion detection system architecture.

can be used by filtration server for filtering intruder's traffic. In-Memory database increases the efficiency of the system by providing data with high speed for comparisons and searching. A complete picture of the architecture is shown in Figure 2.

D. Proposed Algorithm

A single algorithm is anticipated while working on all layers to identify intruder flows. All Flows are distinguished by four tuples i.e., source IP, destination IP, source port, and destination (Src_IP, Dst_IP, Src_Port, Dst_Port). Algorithm 1 describes the pseudo code of the proposed algorithm. Initially, for each captured packet, the filtration is performed at FLBS as describe in step 2 by passing the packets for processing, which are not identified as an intruder or normal flows. Step 3 is performed at a master node to checks whether the incoming packet belongs to already registered flow? If it is not belonging to the registered flow then it is registered as new flow distinct by (src_IP, dst_IP, src_port, dst_port) and new sequence file is created for this flow by inputting necessary packets information in the first line. On the other hand, if the packet belongs to the registered flow, then the packets information is just sent to the particular sequence file corresponding to that registered flow. When the duration threshold deviates, the sequence file is sent to one of the data nodes for flow parameters calculations as coded in step 5. Data node uses the Spark and Map and Reduce function as a backend, equipped with parameters calculations code to measure the final values for each of 9 features for intrusion detection. The Spark code have the capability to run in parallel by taking sequence file as input on Hadoop environment. Since each data node processes distinct flow information in parallel, the overall performance is enhanced. Finally, the calculated features values are sent to decision server, which is equipped with various ML classifiers to decide about the flow whether it is an intrusion or normal flow based on its features values. The decision made by decision servers are then informed to the in-memory database at FLBS for updating in intruders flows list. The complete picture of the flow of the system is depicted in Figure 3.

Algorithm 1. IDS Algorithm Pseudo Code

1. For each incoming packet do step 2-8
2. IF (Flow_already_classified?=Yes)
   Return; //return to next incoming packet
3. Else-IF(flow_already register?=No)
   Flow_list=Flow_list+new_flow(pkt_src_IP,pkt_dst_IP,Pkt_src_port,pkt_dst_port)
   Add_packet_Papameters(packet values); //Add parameters
   Return; //return to next incoming packet
4. Else-IF(flow_already register?=yes)
   add_packets_parametrs(registered flow, Sequence file)
   ENDIF
5. IF(Flow_duration< time_threshold)
   Return; //return to next incoming packet
   Else //Send to Data Node and processing
   a. Send sequence file to data node
   b. For each (sequence file) // at Data Node
      Calculate Flow parameters/features.
   c. Send feature values to Decision making server
   ENDIF
6. Result=Machine_Learning_classifier(params values)
7. Store result(); // in In-Memory DB
8. Return; //return to next incoming packet

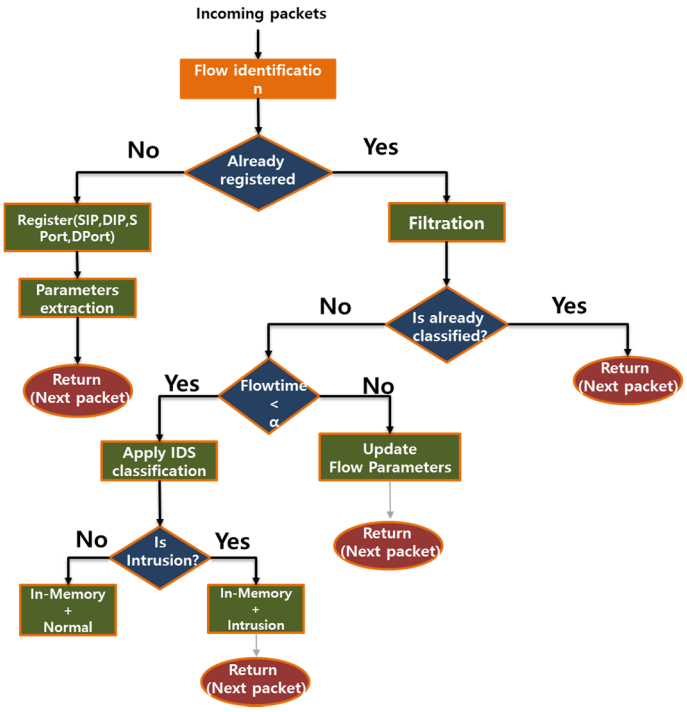

Fig. 3. The Flow of the IDS algorithm.

IV. IMPLEMENTATION AND EVALUATION

The proposed IDS is implemented in MapReduce programming using a single node Hadoop, which processes the sequence file and calculates parameters' values. Most widely and more efficient ML classifiers are selected and implemented in java to evaluate the proposed system and the selected features. The selected classifiers are Naïve Bayes, support vector machine (SVM), Conjunctive rule, Random forest tree, REPTree, and J48 (C4.5 Java implementation). The proposed system is evaluated by accuracy while considering true positive (TP) and False positive (FP) and by efficiency in terms of processing time. Accuracy evaluation is done using above mentioned ML classifiers by taking three KDD [24] dataset files i.e., corrected dataset file, KDDCup.data.corrected, and KDDcup.data.corrected.10%. Finally, the comparison is made with existing IDS i.e., RF-FSR and RF-BER [26], Kayacik [8], Araujo [9], and Kantor [10], with respect to accuracy and efficiency. The complete accuracy results in terms of TP and FP is shown in Table2. Intrusion detection by using proposed 9 features performs well at J48 and REPTree classifiers. The accuracy in terms of TP is more than 99.9% on KddCup.Data.Corrected and KddCup.Data.Corrected_10% dataset files, and Corrected Dataset file. Similarly, the FP rate is also very less i.e., less than .0001% for all intrusion dataset files using J48 and REPTree. Accuracy results also show that the choice of using conjunctive rule classifier for intrusion detection is not good as it has very low TP and higher FP rate as compared to other ML classifiers.

While considering efficiency in terms of time, since the proposed solution has less number of parameters and it is implemented on the parallel environment of Hadoop, therefore, it takes a short time to process larger datasets than the existing intrusion detection techniques. The IDS implementation using REPTree classifiers is most efficient for both accuracy and efficiency point of view as shown in Figure 4. Naïve based

TABLE 2. ACCURACY OF THE PROPOSED SYSTEM ON KDD99 DATASET FILES

		Corrected Dataset File		KddCup.Data.Corrected		KddCup.Data.Corrected_10%	
S #	Classifiers	TP(%)	FP(%)	TP(%)	FP (%)	TP (%)	FP (%)
1	Naive Bayes	94.1	.002	94.7	0.0001	95	0.0015
2	Conjunctive Rule	80.1	.05	94.3	0.0001	95.8	0.0001
3	SVM	97.7	.005	75	0.0001	78.95	0.061
4	Random Forest	98.9	.002	99.9	0.0001	99.9	0.00001
5	J48	99.9	0	99.9	0.0001	99.9	0
6	REPTree	99.9	.0005	99.9	0.0001	99.9	0

classifiers also performed well using proposed features in terms of processing time, but it is not more accurate. Moreover, it takes more time to detect intrusions than REPTree detection time. The time consumed by different classifier on the model building by using proposed feature on three datasets file is shown in Figure 4. The size of KddCup.Data.Corrected file is larger. Therefore, every classifier takes more time while building a model using KddCup.Data.Corrected file.

Figure 5 shows the time elapsed while making a decision by each classifier after model building using KDDCup dataset files. Random Forest and Naïve-Bayes implementation took more time while identifying intrusions in KDDcup.Data.Corrected file. REPTree, J48, and Conjunctive rule classifiers took almost same time while processing dataset for intrusion detection. However, as shown in table 2, the Conjunctive rule classifier's accuracy is lower as compared to other classifiers. Moreover, Naïve Bayes classifier is more efficient for the model building but less efficient for decision making. The SVM is neither efficient while model building nor by decision making. Finally, by analyzing the accuracy and efficiency results of various ML classifiers, we conclude that REPTree and J48 are two best choices for intrusion detection with higher accuracy and more efficiency using the proposed features on Hadoop.

Finally, the comparison is made with existing techniques, mentioned above, while considering efficiency in terms of processing time and accuracy in terms of TP and FP. It is obvious from results on various datasets that the proposed IDS system have better accuracy results using any of the ML classifiers as shown in Table 3. The system outperforms all the other techniques except RF-BER. In the case of RF-BER IDS technique, the TP and FP rate for both of the system are almost equal using J48 classifier. However, for using other classifiers, the proposed system outperforms the RF-BER as well. Moreover, the system also defeated the RF-BER with respect to efficiency as shown in Figure 4, 5. Similarly, in some other cases, the FP rate of the proposed system are equal to some of the other existing schemes. But, the TP of those schemes is quite lower.

The efficiency comparison is made based on the time consume for building a classification model for intrusion detection as well as for decision-making time i.e., classification itself for corrected dataset file. The efficiency comparison graph is shown in Figure 6 in terms of processing time of model building. Figure 7 shows a comparison in terms of processing time for institution classification or decision making. It is clear

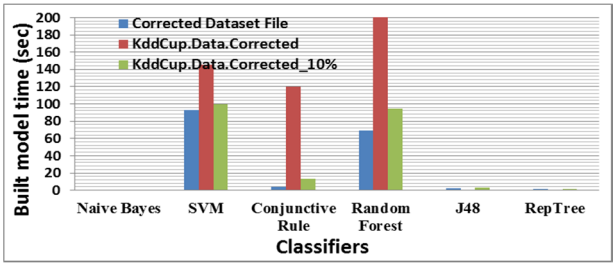

Fig. 4. Time consumed by each ML classifier to build a model on various of KDD99 Dataset files.

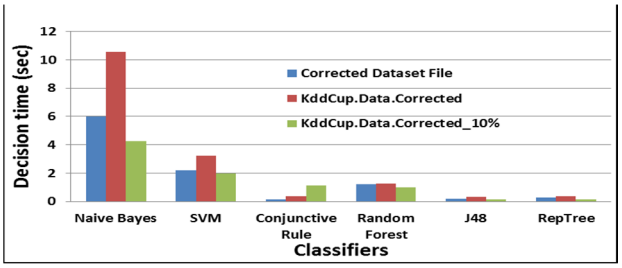

Fig. 5. Time consumed by each ML classifier to detect intrusion on various of KDD99 Dataset files.

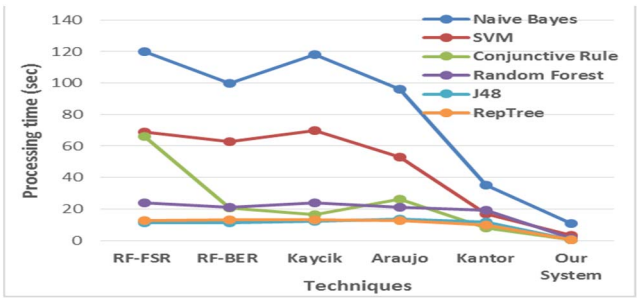

Fig. 6. Efficiency comparison of the proposed scheme against existing IDS based on classification (detection) time on KDDcup.Data.Corrected file.

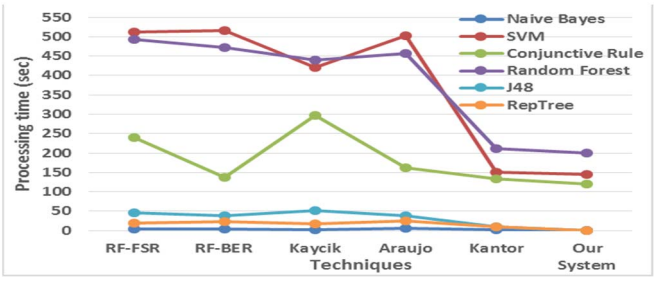

Fig. 7. Efficiency comparison of the proposed scheme against existing IDS based on model-building time on KDDcup.Data.Corrected file.

TABLE 3. ACCURACY COMPARISON AMONG DIFFERENT IDS ON CORRECTED DATA FILE OF KDD99 DATASET

	RF-FSR	RF-BER	Kaycik	Araujo	Kantor	Proposed System	RF-FSR	RF-BER	Kaycik	Araujo	Kantor	Proposed system
classifiers	TP (%)						FP\(%)					
Naive Bayes	91.4	88	91.8	90.1	91.4	94.1	0.003	0.003	0.005	0.002	0.003	0.002
SVM	95.8	95.8	95.8	95.4	94.1	97.7	0.009	0.009	0.009	0.01	0.11	0.005
Conjunctive Rule	72.2	72.2	72.2	72.2	72.2	80.1	0.067	0.067	0.067	0.067	0.067	0.05
Random Forest	98.1	97.9	97.9	97.6	97.2	98.9	0.002	0.003	0.002	0.001	0.004	0.002
J48	98	99.9	97.9	97.5	97.2	99.9	0.002	0	0.002	0.001	0.004	0
REPTree	97.9	97.7	97.9	97.4	97.2	99.9	0.003	0.003	0.003	0.001	0.004	0.0005

from the efficiency graph of all the IDS schemes that the proposed scheme always has minimum processing time while building IDS model and identifying intrusions using the built model. Since IDS model building is done without any parallel processing (not using multiple mapper and Reducer), therefore, the proposed system model building time is slightly higher than few existing schemes. For REPTree and J48 classifiers implementation, the proposed system is most efficient and with higher accuracy than any other system. The Evaluation of the system proved that the system is more accurate and more efficient and have the capability to perform better in high-speed networks.

V. CONCLUSION

This paper presents a novel real-time intrusion detection system that can work in a high-speed network environment. The system includes the proposed architecture with four-layers, the parameters selection mechanism, and the proposed intrusion detection technique. The processing layer, which is the main component of the system, is composed of various Hadoop master and data nodes, which is responsible for handling high-speed real-time traffic more efficiently in order to identify any type of intrusions in the network. The system is evaluated with respect to accuracy and efficiency using Apache Spark as a third party tool on top of the Hadoop single node setup and MapReduce as a backend programming. Various machine learning approaches are used with proposed features to provide a distinction between the intrusions and the normal flows. Finally, the REPTree and J48 ML classifiers are selected for intrusion classification using the proposed features based on their performance results. The evaluation results and comparative study show that the proposed system is more efficient, more accurate, and have the ability to work in the high-speed real network environment.

ACKNOWLEDGMENT

This research was supported by Kyungpook National University Bokhyeon Research Fund, 2015. This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP). [No. 10041145, Self-Organized Software platform (SoSp) for Welfare Devices].

REFERENCES

[1] Vegard Engen, “machine learning for network based intrusion,” Ph.D. dissertation, Bournemouth Univ., Poole, UK, 2010.

[2] ofcom. (2013, Aug 1). “Communications market report 2013,” [Online]. Available: www.ofcom.org.uk/cmruk/

[3] S. Sagiroglu and D. Sinanc, “Big Data: a review," 2013 International Conference on Collaboration Technologies and Systems (CTS), pp. 42-47, IEEE, 2013.

[4] Xindong Wu, Xingquan Zhu, Gong-Qing Wu, Wei Ding, "Data mining with Big Data," IEEE Transactions on Knowledge and Data Engineering, vol.26, no.1, pp.97-107, Jan. 2014.

[5] W. R. Pires, Jr., T. H. de Paula Figueiredo, H. C. Wong, and A. A. F. Loureiro, “Malicious node detection in wireless sensor networks,” in Proc. 18th Int. Parallel Distrib. Process. Symp., Apr. 2004.

[6] R. Puttini, M. Hanashiro, F. Miziara, R. de Sousa, L. Garcia-Villalba and C. Barenco, “On the Anomaly Intrusion-Detection in Mobile Ad Hoc Network Environments,” Proc. of 11th IFIP TC6 international conference on Personal Wireless Communications, Springer, pp. 182-193, 2006.

[7] R. Rao and G. Kesidis, “Detecting malicious packet dropping using statistically regular traffic patterns in multihop wireless networks that are not bandwidth limited,” IEEE GLOBECOM, 2003.

[8] H. G. Kayacik, A. N. Zincir-Heywood, and M. I. Heywood, “Selecting features for intrusion detection: a feature relevance analysis on kdd99 intrusion detection datasets,” 3rd annual conference on privacy, security and trust, Citeseer, 2005.

[9] N. Araujo, R. de Oliveira, E.-W. Ferreira, A. Shinoda, and B.Bhargava, “Identifying important characteristics in the kdd99 intrusion detection dataset by feature selection using a hybrid approach,” IEEE 17th International Conference on Telecommunications (ICT), pp. 552–558, 2010.

[10] P. Kantor, G. Muresan, et.al. “Analysis of three intrusion detection system benchmark datasets using machine learning algorithms,” Intelligence and Security Informatics, Germany, sec. 3, pp. 363 Springer - Verlag, 2005.

[11] T. Abbes, A. Bouhoula, and M. Rusinowitch, “Efficient decision tree for protocol analysis in intrusion detection,” International J. Security and Networks, vol. 5, no. 4, pp. 220–235, December 2010.

[12] C. Wagner, J. Franc¸ois, R. State, and T. Engel, “Machine Learning Approach for IP-Flow Record Anomaly Detection,” ln International Conference on Research in Networking, Berlin Heidelberg, pp. 28-39. , 2011.

[13] J. R. Quinlan, “Induction of Decision Trees,” Machine Learning,” vol. 1, no. 1, pp. 81–106, March 1986.

[14] B. Sch¨olkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, “Estimating the Support of a High-Dimensional Distribution,” Neural Computation, vol. 13, no. 7, pp. 1443–1471, July 2001.

[15] A.P. da Silva, M. Martins, B. Rocha, A. Loureiro, L. Ruiz and H.C. Wong, “Decentralized Intrusion Detection in Wireless Sensor Networks,” Proc. 1st ACM International Workshop on Quality of Service and Security in Wireless and Mobile Networks (Q2SWinet ’05), pp. 16-23, 2005.

[16] El-Khatib, "Impact of feature reduction on the efficiency of wireless intrusion detection systems," IEEE Transactions on Parallel and Distributed Systems, vol. 21, no. 8, 1143-1149, 2010.

[17] K. Nadkarni and A. Mishra, “Intrusion detection in MANETs-the second wall of defense,” Proc. 29th Annual Conference of the IEEE Industrial Electronics Society, 2003.

[18] L. Khan, M. Awad, and B. Thuraisingham, “A New Intrusion Detection System Using Support Vector Machines and Hierarchical Clustering,” The VLDB Journal, vol. 16, no. 4, pp. 507–521, October 2007.

[19] Yu, Z., Tsai, J. J., & Weigert, “An automatically tuning intrusion detection system,” IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 37(2), 373-384.

[20] Prathibha, P. G., and E. D. Dileesh. "Design of a hybrid intrusion detection system using snort and hadoop," International Journal of Computer Applications, 73(10), 2013.

[21] S. R. Bandre and J. N. Nandimath, "Design consideration of Network Intrusion detection system using Hadoop and GPGPU," 2015 International Conference on Pervasive Computing (ICPC), Pune, pp. 1-6, 2015. doi: 10.1109/PERVASIVE.2015.7087201

[22] H. D. J. Jeong, W. Hyun, J. Lim and I. You, "Anomaly Teletraffic Intrusion Detection Systems on Hadoop-Based Platforms: A Survey of Some Problems and Solutions," 2012 15th International Conference on Network-Based Information Systems (NBiS), Melbourne, pp. 766-770, 2012. doi: 10.1109/NBiS.2012.139

[23] I. S. T. G. MIT Lincoln Lab, “DARPA Intrusion Detection Data Sets,” [ONLINE].http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/2000data.html, March 2000.

[24] KDDcup99, “Knowledge discovery in databases DARPA archive,” [ONLINE]. http://www.kdd.ics.uci.edu/databases/kddcup99/ task.html, 1999.

[25] NSL-KDD, “NSL-KDD data set for network-based intrusion detection systems,”[ONLINE]http://iscx.cs.unb.ca/NSL-KDD/, March 2009.

[26] Al-Jarrah, O. Y., et al. "Machine-Learning-Based Feature Selection Techniques for Large-Scale Network Intrusion Detection." 2014 IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW), 2014.

[27] ENGEN, “Machine learning for network based intrusion detection,” Doctoral dissertation, Bournemouth University, 2010.

[28] S. Zaman and F. Karray, “Features selection for intrusion detection systems based on support vector machines,” Sixth IEEE Consumer Communications and Networking Conference, CCNC 2009. pp. 1–8, 2009.

[29] F. Fusco and L. Deri, “High Speed Network Traffic Analysis with Commodity Multi-core Systems,” ACM IMC 2010, Nov. 2010.