Machine-Learning-Based Feature Selection Techniques for Large-Scale Network Intrusion Detection

O. Y. Al-Jarrah[1], A. Siddiqui[1], M. Elsalamouny[1], P. D. Yoo[1, 2], S. Muhaidat[1, 3], K. Kim[2]

[1]*Dept. of Electrical and Computer Engineering, Khalifa University, Abu Dhabi, U.A.E*
[2]*Dept. of Computer Science, KAIST, Daejeon, South Korea*
[3]*Dept. Electrical Engineering, University of Surrey, Guildford, U.K*

{paul.d.yoo,sami.muhaidat}@ieee.org, kkj@kaist.ac.kr

***Abstract*—Nowadays, we see more and more cyber-attacks on major Internet sites and enterprise networks. Intrusion Detection System (IDS) is a critical component of such infrastructure defense mechanism. IDS monitors and analyzes networks' activities for potential intrusions and security attacks. Machine-learning (ML) models have been well accepted for signature-based IDSs due to their learnability and flexibility. However, the performance of existing IDSs does not seem to be satisfactory due to the rapid evolution of sophisticated cyber threats in recent decades. Moreover, the volumes of data to be analyzed are beyond the ability of commonly used computer software and hardware tools. They are not only large in scale but fast in/out in terms of velocity. In big data IDS, the one must find an efficient way to reduce the size of data dimensions and volumes. In this paper, we propose novel feature selection methods, namely, RF-FSR (RandomForest-Forward Selection Ranking) and RF-BER (RandomForest-Backward Elimination Ranking). The features selected by the proposed methods were tested and compared with three of the most well-known feature sets in the IDS literature. The experimental results showed that the selected features by the proposed methods effectively improved their detection rate and false-positive rate, achieving 99.8% and 0.001% on well-known KDD-99 dataset, respectively.**

***Keywords—intrusion detection system, feature selection, machine learning, random forest*.**

I. INTRODUCTION

Due to recent technological advances, network-based services have become increasingly vital in modern society. Intruders look for the vulnerabilities of computer systems in order to compromise their communications or to gain illegal access to the core of the systems. However, existing security mechanisms are still inflexible, unscalable, and not powerful enough to deal with such attacks.

In early days, the rule-based methods were dominant. These methods find the intrusions by comparing its characteristics to known attack signatures. Security experts manage the computer-encoded rules which are extracted from real intrusions. As the network traffic grows rapidly, keeping these rules updated becomes more and more difficult, tedious, and time-consuming. Since then, Machine Learning (ML) based methods were introduced to the problem of network intrusion detection. ML refers to computer algorithms that have ability to learn from past examples. In context of intrusion detection, a detection model learns from previously recorded attack patterns (i.e., signatures), and detects similar ones in incoming traffic. The popularity of ML-based models came from the fact that it could be "tailored" to the network data of the system where the model is being used. ML-based IDSs have performed well in the literature as well as the reality. However, the "model-free" property of such methods causes relatively high-computational cost. Moreover, as the volume and velocity of network data grows rapidly, such computing cost issues must be resolved. Hence, this paper gives an insight into the features selection techniques in IDS and proposes two different novel feature selection methods that could help improve performance of any ML-based IDS. The proposed methods use an ensemble of Random Forest (RF) algorithm, with forward and backward ranking features selection techniques [1–2]. To prove the usefulness of the proposed methods, we compare our results with those of other three well-known feature sets [3–5] on KDD-99 dataset.

II. BIG DATA IDS

Due to the recent emergence of new technologies such as ubiquitous networks, wireless sensors networks and web technology, our society is heavily connected, and the usage of computers has grown exponentially. For an example, only 65% of U.K. houses were connected to the Internet in 2008 [6], which was approximately 16 million households, whereas in the first quarter 2013, 80% of U.K houses were connected to the internet [7]. In addition, the world's digital content was estimated 2.72 zettabytes in 2012, and it is expected to reach 8 zettabytes by 2015 [8]. 90% of the world's current digital content has been generated during the last two years only [9]. Such data in large or extreme scale have a few important characteristics, which pose a challenge to the existing computer systems, namely, variety, velocity, and volume.

The rapid growth of such large data volume not only overwhelms existing computer systems but also introduces new challenges to existing technologists [10]. First of all, large volumes of data pose the need for efficient methods to store, link, and process data. However, existing database management systems (i.e., DBMS) are not capable enough to handle such tasks. Second, big data flows at high velocity, which exceeds the organizations' ability to store and process it [11]. To build efficient and rapid IDS, such data velocity must be carefully considered. Third, big data has various types. It could be structured, semi-structured, and unstructured, as all comes from different sources. Thus, inflexible models may not be able to deal with such data variety.

With big data, the attention must be given to the relationships and connections between objects and entities. Its

value comes from the knowledge that might be extracted from data structures among pieces of data. In addition, big data in nature has various connections spreading in different directions. We might be able to find patterns or data structures that do not exist or appear clearly in small scale. Such scenario could cause undesirable consequences that affect system's performance and security. It is thus crucial to investigate such big data characteristics and build an IDS that addresses such issues. More importantly, in the light of finding relevant features, an IDS not only decreases its computational cost but also detects its hidden patterns that do not exist in small scale more efficiently.

III. METHODS

Our approach to ML-based feature selection for the detection of network intrusions consists of four consecutive steps. i. dataset selection and preprocessing, ii. feature selection, iii. model selection and iv. evaluation.

A. Datasets Selection and Preprocessing

Choosing a right dataset for model validation and evaluation is not a trivial task. The properties we look at are completeness, noise-free, consistency, and redundancy. Among several publicly available datasets, KDD-99 is the most widely accepted benchmark. However, KDD-99 dataset has some drawbacks [12]. First, the full dataset is large which increases the computational cost of the IDS. Therefore, only 10% of the set is usually used [13]. Second, KDD-99 has many redundant data in the training set and duplicated records in the testing set. That might affect the learning process. It causes learning biasing to the frequent records and prevents learning the infrequent records, which might be more harmful to the system.

NSL-KDD99 dataset is a filtered version of KDD-99. The redundancies between testing and training sets have been minimized. Due to its reduced size, a learning algorithm could learn from NSL-KDD99 almost instantly [12–13]. Therefore, IDSs can use the whole dataset while detecting different types of attacks more precisely. Similar to KDD-99, NSL-KDD99 has 41 features that contain both normal and attack patterns. Attacks in KDD-99 dataset fall into four categories [12]:

- Denial of Service (DoS): attacker creates computational load on the system that drains system resources and prevents the legitimate use of its resources.
- User to Root Attack (U2R): it is when the attacker has normal user access and tries to gain root user access to the system.
- Remote to Local Attack (R2L): it is when the attacker gets access of a normal user account.
- Probing Attack (PROB): it is when the attacker tries to gather sensitive information by scanning data in the system with no access permission.

Due to the limitation of datasets, data needs processing in order to make it compatible and suitable for learning model. This process includes the following:

- Removal of Redundant Records: Within a dataset, there might be thousands of repeated records. Deleting the repetitive records improves the accuracy of the model and reduces computation cost [2]. For instance, 78% of KDD-99 records are repeated records [13]. However, if such records are not removed, the under training will not learn new records [14].
- Enumeration of Data: The variables of data are of different forms. Some are alphanumeric, while others are simply numbers such as 'Protocol Type' and 'Connection Duration' [15]. In such a case, data needs to be 'enumerated' before it is used to train or test a model. Enumeration of data includes converting alphanumerical variables to numerical.
- Normalization of Data: Normalization is done so that none of the input variables have a dominant impact on the training results [15]. The process of normalizing data involves condensing the data to the scope of the model without losing the effect of the data on the model.
- Discretization of Data: In the KDD-99 dataset, there are many types of variables. Some are continuous variables, whereas others are discrete. To make computation easier, it is important to discretize the continuous variables [16].
- Balancing of Data: In the KDD-99 dataset for instance, each type of attack have many connections. Denial of Service (DoS) attack type has 391458 connections; Whereas User to Root (U2R) attack has only 52 connections. The unbalanced dataset causes biasing in learning process of the models. For example, in KDD-99 dataset, model might learn DoS more precisely than U2R since the DoS set has more connections. The other advantage of balancing data is that it significantly reduces the time to build pattern [17]. Therefore, sometimes it is important to down sampling connections. In KDD-99 dataset for instant, the DoS connections might need some down sampling [17].

Each preprocessing step has its own importance. However, not all steps need to be applied for every dataset in order to prepare it for training. The machine-learning model and feature selection process play important roles in deciding the preprocessing measures.

B. Features Selection

Using all the features of a dataset does not necessarily guarantee the best performances from the IDS. It might increase the computational cost as well as the error rate of the system [1, 2]. As the number of selected features increases, the required computation power, which is needed to process the data, increases, and vice versa. Therefore, due to correlations between features, some selected features can achieve similar or better results in comparison to using all the features of a dataset.

Feature selection is used in intrusion detection to eliminate the redundant and irrelevant data. It refers to the process of selecting a subset of relevant features that fully describes the given problem with a minimum degradation of performance [19]. The irrelevant or redundant data might contain false correlations which obstruct the learning process of the classifiers. Therefore, features selection has a significant impact on intrusion detection systems performance as it reduces the computation cost, removes information

redundancy, increases the accuracy of detection algorithm, facilitates data understanding and improves generalization [20].

Typical features selection process includes three phases [14, 21], subset generation, subset evaluation and validation. Subset generation produces a candidate features subsets which are selected based on search strategy. Complete, heuristic and random approaches are the most common subset generation approaches. Then, each candidate features subset is compared with the best previous candidate features subset based on evaluation criteria. If the new candidate features subset found to be better, it replaces the previous best candidate features subset. Subset evaluation might be based on score, entropy, correlation, consistency and detection accuracy. Finally, the generated subset is validated in real world implementation or simulation environment.

Features selection models are categorized into three categories based on the evaluation criteria [21]: 1) filter model; 2) wrapper model; 3) and hybrid model. Filter model relies on the general characteristics of the training dataset independently from classifiers feedback to select the best features. Wrapper model optimize classifiers as a part of the selection process. Though it has a higher computational power, wrapper model tends to obtain better performances than filter because it uses classifiers as a part of the selection process [19]. On the other hand, filter model is adequate when dealing with large datasets [19]. Hybrid model exploits the advantages of the previously mentioned models by applying them in different stages of the selection process.

Several methods and techniques were used in literature for feature selection process and find the correlations between different features. These techniques depend on several measurements such as the dataset and detection models, etc. machine-learning algorithms were used in feature selection process. In [1], Random Forest (RF) was used to sort all the 41 features according to their weight. While, Enhanced Support Vector Decision Function (ESVDF) method was proposed in [2]. The proposed method uses Support Vector Decision Function (SVDF) to find different features weight. Then, features correlations, which are the dependence of features on each other, are determined by either Forward Selection Ranking (FSR) or Backward Elimination Ranking (BER) algorithms.

In case of FSR, the first three features with the highest weight form a new feature subset (S2) from the original feature set (S1). Then, the algorithm adds feature by feature from S1 into S2 and calculates system performances values (accuracy and training time). The feature which increases performances values is kept in S2, otherwise, the algorithm will test each feature in the features set (S2) with respect to its effect on features set performances values and select the appropriate feature to be removed. The same procedure is followed for all of the 41 features [2].

In case of BER, the first set (S1) contains all the 41 features. At each step, one feature is removed from S1. Then, the system compares system performances with the previous performances values. If feature removal increases system performances values, then the feature removed from S1, otherwise it retained back to S1. This step is repeated until cover all the 41 features in the dataset [2].

Kaycik in [3] proposed a feature selection method. The proposed method analyzes KDD'99 dataset and its 41 features, based on calculating the information gain and the entropy of each feature to measure its relevance [3]. The last features set, which is used for comparison in this paper, are the 6 important features. These 6 features are chosen by experts as representatives for the 41 features [5].

C. Model Selection

In the literature, many ML models were used in intrusion detection systems. For each classifier, significant research has been done to prove the effectiveness of the model. Single classifiers, hybrid classifier, and ensemble classifier are examples of these models. Single classifier uses one machine-learning algorithm to classify data. Ensemble classifier combines multiple models to generate a single model that has better prediction accuracy. K-Nearest Neighbor, Self-Organizing Maps, Decision Trees, Random Forest (RF), Naïve Bayes, Artificial Neural Networks (ANN) and Support Vector Machines (SVM) are examples of popular classifiers [22]. Due to its advantages, RF is used in this study. RF consists of crowd of decision trees, each one of the decision trees gives a classification of the input data. After that, voting takes place and the forest comes up with the final classification decision based on the voting result.

RF has many advantages [18]. First, computationally, it is not demanding especially when used with large-scale data (i.e. big data). Second, RF surpasses decision trees and Naïve Bayes models in terms of accuracy and detection rate. Third, the training time it takes is also comparable to that of decision trees. Fourth, it can find correlations among data which provides prediction ability. Fifth, it has method to balance error when data set is not balanced [23].

D. Evaluation

The model performance on each feature set should be compared and evaluated fairly. Several experimental settings were considered to evaluate the performance of each feature set. These include detection rate (also known as sensitivity-Sn), accuracy (Acc), training time (Tr), Mathew's correlation coefficient (Mcc), and, False Alarm Rate (Far). We aim to have a high Acc, Sn, and Mcc while low in Tr and Far. Some of these parameters are defined as follows [24]:

$$Far = \frac{FP}{TN+FP},$$

$$Acc = \frac{TP+TN}{TP+TN+FP+FN},$$

$$Sn = \frac{TP}{TP+FN},$$

$$Mcc = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}.$$

True-Positive (TP) is the number of attacks classified as attacks, True-Negative (TN) is the number of normal intrusions classified as normal intrusions, False-Positive (FP) is the number of attacks classified as normal intrusion, and False-Negative (FN) is the number of normal intrusions classified as intrusions.

IV. EXPERIMENTS AND RESULTS

The two proposed feature selection methods are devised by combining the results of researches [1–2] in a way that the weight of the features is determined from research [1] using the RF approach, then both FSR and BER are applied separately. The third feature selection method that is used to compare the proposed methods is done by Kaycik [3]. Analysis of KDD-99 dataset with its 41 features was presented based on calculating the information gain and the entropy of each feature to measure its relevance [3]. The fourth feature selection method is proposed by [4]. In this study, a hybrid approach is used to obtain the optimal set of 14 features. The last set of features includes the 6 important features. These 6 features are chosen by experts as representatives for the 41 features [5]. Table 1 shows the features sets used in this study.

TABLE 1: FEATURE SETS

Method	Features
RF-FSR	1, 3, 4, 5, 6, 8, 13, 16, 10, 23, 24, 32, 33, 35, 36
RF-BER	1, 2, 3, 5, 6, 10, 14, 16, 32, 33, 36, 37, 38, 41
Kaycik [3]	1, 2, 3, 4, 5, 6, 8, 11, 12, 16, 23, 24, 26, 32, 33
Araújo [4]	2, 3, 5, 6, 9, 11, 12, 14, 22, 30,31, 32, 35, 37
Kantor [5]	1, 2, 3, 4, 5, 6
KDD-99	1–41

Both KDD-99 and NSL-KDD99 datasets have the identical 41 features. The details of these features are provided in [3].

The machine, which has been used in this experiment, has a memory of 32 GB and a 2 x 2.4 GHz 6-Core Intel Xeon processor. NSL-KDD99 dataset is used for training and testing each selected features. The complete training and testing dataset used is available publicly. 10 fold cross- validation technique was used to perform the experiment. In this method, the training set is divided into 10 subsets and each subset is tested when the model is trained on the other 9 subsets. This process repeats 10 times in which each subset is used as a test data only once. The results obtained from testing each 'fold' are combined to produce a single result. Given a particular classifier in Weka, the parameters need to be adjusted and optimized. Optimizing parameters manually can be time consuming. Hence, an option provided by Weka for optimizing parameters automatically is used. The method is called CVParameterSelection. This parameter selection method and the RF classifier is used along with 10 folds cross validation to evaluate the combination of features.

TABLE 2: EXPERIMENTS RESULTS

Method	Tr	Sn (DR)	Acc	Mcc	Far
RF/FSR	12.75	**99.857**	**99.901**	**0.99801**	0.000609
RF/BER	11.52	99.833	99.881	0.99761	0.000772
Kaycik [3]	9.76	99.732	99.809	0.99616	0.001247
Araújo [4]	12.23	99.840	99.891	0.99781	0.000639
Kantor [5]	**4.77**	99.499	99.354	0.98702	0.007722
KDD-99	22.09	99.830	99.895	0.99790	**0.000505**

As seen in Table 2, the proposed features selection method RF-FSR achieved best performance measures in terms of Sn (Detection Rate), Acc and Mcc. Due to low number of features, Kantor's feature set obtained the least Tr. For the same reason, full features set KDD-99 achieved the highest Tr and the best Far results.

V. CONCLUSION

In this paper, we presented two features selection methods, namely, RF-FSR and RF-BER, the novel ensembles of decision-tree-based (J48/C4.8) voting algorithm with forward selection / backward elimination feature raking techniques. Such feature selection method is of great importance, especially for an IDS designed for large-scale networks where the volume and velocity are high. In this paper, the features selected by the proposed methods were compared with other three popular feature sets on widely known KDD-99 and NSL-KDD99 datasets. The selected features were fairly evaluated and compared with a RF classifier tuned with CVParameterSelection methods. The experimental results showed that the feature set selected by our proposed RF-FSR technique outperformed all other well-known feature sets in the literature, which seems to be promising and suitable for large-scale network IDSs.

ACKNOWLEDGMENT

This research was supported by the KUSTAR-KAIST Institute, under the R&D program supervised by the Korea Advanced Institute of Science and Technology (KAIST), South Korea.

REFERENCES

[1] ENGEN, "Machine learning for network based intrusion detection," Doctoral dissertation, Bournemouth University, 2010.

[2] S. Zaman and F. Karray, "Features selection for intrusion detection systems based on support vector machines," in Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE, pp. 1–8, 2009.

[3] 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," in Proceedings of the third annual conference on privacy, security and trust, Citeseer, 2005.

[4] 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," in IEEE 17th International Conference on Telecommunications (ICT), pp. 552–558, IEEE, 2010.

[5] P. Kantor, G. Muresan, F. Roberts, et.al. "Analysis of three intrusion detection system benchmark datasets using machine learning algorithms," in Intelligence and Security Informatics, Germany, Berlin Heidelberg, sec. 3, pp. 363 Springer - Verlag, 2005.

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

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

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

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

[10] C. Barnatt. (2013, Sep). "Big data,"[Online]. Available: http://explainingcomputers.com/big_data.html.

[11] S. Lohr. (2012, Feb 1)."The age of big data,"[Online]. Available: http://www.nytimes.com/2012/02/12/sunday-review/big-datas-impact-in-the-world.html?pagewanted=all&_r=0.

[12] M. Tavallaee, E. Bagheri, W. Lu, and A.-A. Ghorbani, "A Detailed analysis of the kdd cup 99 data set," in Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009, 2009.

[13] Y.-X. Meng, "The practice on using machine learning for network anomaly intrusion detection," in International Conference on Machine Learning and Cybernetics (ICMLC), vol. 2, pp. 576–581, IEEE, 2011.

[14] H. Harb, A. Zaghrot, M. Gomaa, A. Desuky ,"Selecting optimal subset of features for intrusion detection systems," Advances in Computational Sciences and Technology. Vol. 4, pages 179-192, 2011.

[15] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, "An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks," Expert systems with Applications, vol. 29, no. 4, pp. 713–722, 2005.

[16] M. Fadaeieslam, B. Minaei-Bidgoli, M. Fathy, and M. Soryani, "Comparison of two feature selection methods in intrusion detection systems," in IEEE International Conference on Computer and Information Technology, pp. 83–86, IEEE, 2007.

[17] J. Zhang, M. Zulkernine, and A. Haque, "Random-Forests-Based network intrusion detection systems," IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 38, no. 5, pp. 649–659, 2008.

[18] P. Amudha and H. Abdul Rauf, "Performance analysis of data mining approaches in intrusion detection," in International Conference on Process Automation, Control and Computing (PACC), pp. 1–6, IEEE, 2011.

[19] V. Bol´oN-Canedo, N. S´aNchez-Maro˜no, and A. Alonso-Betanzos, "Feature selection and classification in multiple class datasets: an application to kdd cup 99 dataset," Expert Syst. Appl., vol. 38, pp. 5947–5957, 2011.

[20] T. Lappas and K. Pelechrinis, "Data mining techniques for (network) intrusion detection systems," Department of Computer Science and Engineering UC Riverside, Riverside CA, vol. 92521, 2007.

[21] M. Bhuyan, D. Bhattacharyya, and J. Kalita, "Network anomaly detection: methods, systems and tools," Communications Surveys & Tutorials, IEEE, vol.PP, no.99, pp.1, 34, 0 2013.

[22] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "intrusion detection by machine learning: a review," Expert Systems with Applications, vol. 36, no. 10, pp. 11994–12000, 2009.

[23] L. Breiman, "Random Forests," Machine learning, vol. 45, no. 1, pp. 5-32, 2001.

[24] P. D. Yoo, A. Y. Zomaya," A Computational consensus approach for proline cis-trans isomerization prediction," IEEE Transactions on Computational Biology and Bioinformatics, Jan 2013 (accepted).