Optimality of Feature set for Intrusion Detection System

Zeeshan Naseer
Applied Artificial Intelligence Group
Islamabad, Pakistan
zeshan412@gmail.com

Muhammad Rizwan
Department of Computer Science
Capital University of Science and Technology, Islamabad, Pakistan
rizwanabuahmad@gmail.com

Sohail Sarwar
Department of Computer Science
NUST, SEECS
Islamabad, Pakistan
sohail.sarwar@seecs.edu.pk

Muhamad Bilal Khan
Department of Computer Science
Capital University of Science and Technology, Islamabad, Pakistan
K.bilal_g@yahoo.com

2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST) | 978-1-6654-0516-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/IBCAST51254.2021.9393257

***Abstract*—Intrusion detection systems (IDSs) became indispensable with the emerging requirement of security in computer network systems. Conventional detection techniques, like signature and rule-based intrusion detection, require regular human intervention or let the intrusion undetected. Fortunately, detection through Machine learning (ML) is free of such shortcomings. However, the selection of the most significant and predictive features is a challenge. The research community is quite active in the selection of the best subset of features in IDS. However, there is lacking a structured selection procedure and ordered list of features. We attempt to provide a more concrete list of features regarding their significance in predicting intrusion. We perform a survey and follow a structured methodology in features' selection out of the publicized dataset NSL-KDD. The features' selection procedure comprises five steps. The first three steps are dedicated to drop trivial features, while the last two steps are performed to identify the useful features. Model building is done using Support Vector Machine (SVM) for its wide acceptability in the IDS research community. The result comprises the ordered list of features. The features are sorted as per their predictive ability in classifying the malicious and benign network traffic.**

***Index Terms*—Features' subset selection, Host based systems, Intrusion detection system, Machine learning**

I. Introduction

Intrusion Detection System (IDS) is a necessary component in ensuring network infrastructures. Over the past decade, security issues gain a significant focus due to excessiveness in network intrusions as reported by Computer Emergency Response Team (CERT). These intrusions give birth to disasters and extensively violates security, i.e., Confidentiality, Integrity, and Availability (CIA). The research community is quite active in addressing the associated issues in IDS. With the emerging requirements of security in computer network systems, the issues related to the coping of threats to network and information security are even more candidate to be addressed. Though there is a number of existing literature to survey IDS and its taxonomy [1]–[6].

As defined by National Institute of Standards and Technology (NIST) [7] Intrusion is an attempt to compromise any or all of the CIA factors. Whereas, detection aims to monitor the presence of intrusion. The same is true for both wired and wireless communication. [8, 9]. An IDS is a software or hardware or both to automatically perform detection of intrusion in the network traffic [7, 10]. Apart from IDS, intrusion prevision system (IPS) attempts to prevent the occurrence of intrusion events. IPS works with IDS and the community used them interchangeably. However, IPS is beyond the scope of this study.

IDSs can broadly be classified into two classes; Signature-based Detection (SD), Machine learning-based detection (MLD). SD is the relatively simple and equally effective method to detect known attacks or threats. The process compares the stored patterns against the event under test. However, the technique is not capable to detect unknown attacks, with the added requirement of signature keeping. Whereas, the MLD is quite effective in identifying the unknown intrusions, however, the technique is not susceptible to False positives [11]–[14].

Another orthogonal classification of IDS is host-based intrusion detection (HIDS) and network-based intrusion detection (NIDS). A HIDS visualizes the behavior of hosts that contain sensitive information. Whereas, a NIDS analyzes the activities of applications/protocols and evaluates specific network segments to identify distrustful occurrences.

Prevention of False positive (FP) and false negative (fn) while keeping the optimum degree of accuracy are major challenges of any IDS system. The FP occurs when IDS incorrectly classifies the legitimate activity as malicious, whereas the FN occurs when IDS could not recognize nasty occurrences. [10, 15, 16]. Recently, Ho et al. [17] collect False Positive (FP) and False Negative (FN) cases from real-world traffic.

Unfortunately, SD covers attacks that are far less than the number of false alarms raised. This lets real intrusion undetected. Moreover, the added noise can greatly degrade the performance of SD. Maintenance and updating is always required for SD to keep the signatures up to date with the

978-0-7381-0535-2/21/$31.00 ©2021 IEEE

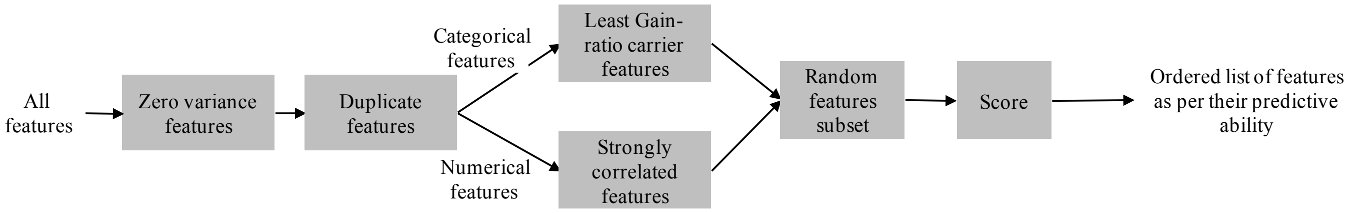
All features
Zero variance features
Duplicate features
Categorical features
Numerical features
Least Gain-ratio carrier features
Strongly correlated features
Random features subset
Score
Ordered list of features as per their predictive ability

Fig. 1: Feature selection methodology

new threats. Apart from that Protocol-based attacks can cause the SD-based IDS to fail.

Fortunately, this is exactly where ML-based IDS comes into play. ML-based IDS lets computers to train and act without any human assistance. It "learns" the specific patterns of the network and can determine the presence of intrusion. It is capable to identify the types of intrusion which are not known before, thus prone to false alarms also. We teach ML models to differentiate between a benign and a malicious packet in the traffic. Moreover, it can handle the attack which it learned before quite effectively and can also recognize mutants of these known attacks. Keeping in view the above pros, numerous studies have been performing intrusion detection using ML, hence advocates the viability of ML-based IDS. However, the selection of the most significant and predictive features is a challenge. Which is the main problem being addressed in this paper.

The rest of the paper is structured as follows: *Section II* presents the literature review of this field followed by the *Section III* that elaborates the process of selection and filtration of features. Finally, the conclusion and possible issues to be addressed in future are elaborated in *Section IV*.

II. Literature review

The research community is quite active in the identification/selection of the best features to improve IDS. However, there is lacking structured selection procedures and an ordered list of features. Here we briefly discuss a few studies that focus on the selection of the best subset of features in IDS.

Jain et al. [18] report that in large datasets forward sequential approach performs better. Whereas, another study conducted by Kudo and Sklansky [19] found the effectiveness of random methods. When it comes to large-scale feature selection these methods are quite computationally expensive. For evaluation criterion, support vector machines (SVM) is appreciated for their overcoming local minima. Weston [20] apply SVM for feature selection based upon finding the features which minimize bounds on the leave-one-out error. After that, Guyon et al. [21] use SVM based upon Recursive Feature Elimination. Specifically, in IDS, Vapnik [22] report the outperformance of SVM for its acceleration in model building and reporting.

[23] has introduced a wrapper based approach for feature selection to make a lightweight IDS. They have modified random mutation hill climbing to search for subset and modified SVM as a wrapper approach to extract an optimized feature subset. They have performed experiments on KDD Cup '99 dataset. [24] use the Genetic Algorithm and SVM for feature selection. Their aim is to reduce the dimension of data, increase the true positive rate, and simultaneously decrease false plosive rates. The genetic algorithm assigns the features a fitness score based on which the optimal feature subset is constructed. [25] has also used SVM to identify a discriminant function for feature reduction. [26] proposes normalized gain based IDS that consists of two modules. The first module constitutes the selection of optimum features using Particle Swarm Optimization (PSO) and the second module uses support vector machine for detection and classification of threats. The first module performs feature ranking using normalized gain and then extracts the most significant features using semi-supervised clustering.

[27] has introduced a mutual information-based algorithm to choose optimal features. The key feature of this paper is that it deals with both linearly and non-linearly correlated data features. They have performed evaluations on KDDCup'99, NSL-KDD, and CIC-IDS2017 datasets [28] has proposed a weighted-feature selection method to improve IDS performance. The proposed approach uses SVM, ANN, and DT for feature selection and ANN for classification. The ML algorithms assign heuristic weights to features based on their contribution and the optimal feature subset is achieved. [29] has also used the same approach as [**?**] using a genetic algorithm and SVM. Senthilnayaki et al. [30] provide a detailed insight of previous work done on feature selection and their approaches. The paper examines filter-based feature selection techniques and wrapper based and Hybrid feature selection techniques. The authors also identify the problem of the KDD dataset which is being used in 26 out of 28 papers they examined and urge to use new datasets for modern-day attacks. Alazab et al. [31] rank the features on a filter based technique with information gain. Using a decision tree for classification they have reduced the number of features to 12 from 41 and hence achieved a remarkable performance enhancement.

Khor et al. [32] perform two different types of feature selection techniques using the KDD 99 dataset. They use Consistency Subset Evaluator and Correlation for Feature Selection Subset Evaluator to reduce the features from 41 to 8.

Zhou et al. [33] propose Correlation-based-Feature-Selection-Bat-Algorithm to address the challenges of irrelevant and redundant data. It's a heuristic algorithm that evaluates the correlation between features and extracts an optimal subset of features. The authors use KDDCup'99, NSL-KDD, and CIC-

IDS2017 datasets for evaluation.

Keeping in view the above studies, it is concluded that there is a lack of structured selection procedure and ordered list of features. In the next section we perform a precisely structured procedure of features' selection and eventually provide the best subset of features for IDS.

III. Experimentation

A. NSL KDD

We select NSL KDD dataset for our experiment [34]. NSL KDD dataset is designed to address the problems found in the KDD'99 dataset. Like in the large study of academic research, we used NSL-KDD dataset which is now a de facto standard benchmark data. The dataset is generally being used to upgrade the efficacy of intrusion detection rate. Through the objective of the dataset was to build an efficient and effective model for intrusion detection, yet the dataset is being used by frequent studies for DL, and data mining community for validation purpose also.

It can safely be used as a benchmark for the training and evaluation of IDS models. Also, there are enough records available for training and testing so no need to use techniques to generate synthesized records that somehow affect the results. The dataset is frequently used by the research community [33, 35]–[43].

The dataset comprises 185559 records with 92904 anomaly instances. Therefore, the dataset is almost balanced. The dataset has 42 features altogether wherein, 11 are categorical and the rest are all numerical. Amongst the categorical, five features are binary-nominal. The semantics of the dataset may be found through its documentation.

B. Selection methodology

Feature selection methodology comprises six phases which is shown in Figure 1.

1) First phase:: In the first phase we drop the features having no change across the 185559 instances. We find one such feature which is *No of outbound commands*.

2) Second phase:: In the second phase we search for the duplicate features. Though this phase does not drop any further feature, yet the phase is worth to perform before applying any ML technique.

3) Third phase:: In this phase we perform two experiments, one is specific to categorical features of the dataset while other the is specific to the numeric features of the dataset. Therefore, we split the dataset into two. The first comprises the categorical features while the second comprises the numerical features. both datasets have their corresponding labels also.

In categorical feature carrier dataset we find the least important categorical feature set by computing Gain-ratio against the *Label*. Six such features are found which are as follows;

- land
- root shell
- is host login
- is guest login
- su attempted
- urgent

Features of numeric features carrier dataset is analyzed by correlation with each other. We perform Pearson rank correlation and identify the features having correlation of 0.9 and above. Table I shows the list of strongly correlated features. Keeping in view this, we drop following seven features;

TABLE I: Strongly correlated features

First feature	Second feature	Correlation
No of compromised	No of root	1
server rate	server server rate	0.99
server rate	destination host server rate	0.97
server rate	destination host server server rate	0.97
server server rate	destination host server rate	0.97
server server rate	destination host server server rate	0.98
error rate	server error rate	0.98
error rate	destination host error rate	0.91
error rate	destination host server error rate	0.95
server error rate	destination host server error rate	0.96
destination host server rate	destination host server server_rate	0.98
destination_host_error_rate	destination_host_server_error_rate	0.91

- No of compromised
- server rate
- server server rate
- error rate
- server error rate
- destination host server rate
- destination host error rate

Upon completion of above processs, we recombine both datasets which were split at the start of third phase.

Upon completion of the above three phases, we are left with 28 features, wherein five are categorical features, whereas the rest of 23 features are numerical in type. Brief statistical description of the numerical and categorical features are shown in Table III and II respectively.

TABLE II: Statistical description of categorical feature set in NSL-KDD dataset

Features	Unique	Top	Frequency
logged in	2	0	112794
protocol type	3	tcp	150727
wrong fragment	3	0	184045
flag	11	SF	112071
service	70	http	57594

4) Random features subset selection: To find the best subset out of the features, wrappers can be a choice. However, it is sometime impractical due to its high computational cost [44]. In our case when we have only 23 features, we need to evaluate 8388608 subsets, which is infeasible. Therefore, we randomly generate 5000 subsets and move to the next phase.

5) Model building and result evaluation: SVM is used for model building and model is evaluated using AuC. The 5000 subsets generated in the last phase, are used as an independent variable. After that we filter the subset which successfully achieve AuC of 0.75. Figure 2 shows the list of best subset of features.

TABLE III: Statistical description of numerical feature set in NSL-KDD dataset

Features	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**	**Std**
duration	289.5	2462.2	0	0	0	0	57715	2462.2
source bytes	4.E+4	5.E+6	0	0	44	276	1.E+9	5.E+6
destination bytes	14230	3.E+6	0	0	0	480	1.E+8	3.E+6
hot	0.2	2	0	0	0	0	101	2
No of failed logins	0	0.1	0	0	0	0	5	0.1
No of root	0.3	20.9	0	0	0	0	7468	20.9
No of file creations	0	0.6	0	0	0	0	100	0.6
No of shells	0	0	0	0	0	0	5	0
No of access files	0	0.1	0	0	0	0	9	0.1
No of outbound commands	0	0	0	0	0	0	0	0
count	84.2	119.2	0	2	13	141	511	119.2
server count	29.5	78.7	0	2	7	17	511	78.7
same server rate	0.7	0.4	0	0.1	1	1	1	0.4
diff server rate	0.1	0.2	0	0	0	0.06	1	0.2
server diff host rate	0.1	0.3	0	0	0	0	1	0.3
destination host count	185.7	97.9	0	92	255	255	255	97.9
destination host server count	118.7	110.9	0	11	70	255	255	110.9
destination host same server rate	0.5	0.4	0	0.05	0.58	1	1	0.4
destination host diff server rate	0.1	0.2	0	0	0.02	0.07	1	0.2
destination host same source port rate	0.2	0.3	0	0	0	0.06	1	0.3
destination host server diff host rate	0	0.1	0	0	0	0.01	1	0.1
destination host server server rate	0.2	0.4	0	0	0	0.31	1	0.4
destination host server error rate	0.1	0.3	0	0	0	0	1	0.3

C. Score computing

Final section comprises the scoring of each features. Scoring is accomplished using the equation 1.

$$Score(F) = \sum_{i=1}^{n} \frac{AccompaningF_i}{AUC_i} \times PresenceOfF_i \quad (1)$$

Figure 3 shows the ordered list of features by their decending order of significane in the prediction of anomaly and normal network traffic.

IV. Conclusion and future work

In this paper, we include a numerous studies in current IDSs. More specifically, ML approaches and features selection for improving the ML approaches in IDS. We also elaborated different approaches has its benefits and challenges, so that we should be careful in the selection of ML approaches. However, there are still lack of optimum framework for improving the feature selection of IDS system. Therefore, we propose a novel framework of feature selection. The proposed framework is simple to it is simple to implement. We empirically evaluate the proposed framework on NSL-KDD dataset. The results show that *No of failed logins* is the best feature followed by *logged in* and *flag*. Moreover, if computation and time persist model may be built by adding more features as per their ordered importance. Now the selected features can be utilized by the IDS development community to upgrade the intrusion detection through ML.

In this article, we provide an overall trend and inclination of current IDS community using ML. Yet, there are still numerous open issues and challenges to be addressed in the future. Considering the other dataset with the variation of classifier may be a prudent work to do.

References

[1] M. Soni, M. Ahirwa, and S. Agrawal, "A survey on intrusion detection techniques in manet," in *2015 International Conference on Computational Intelligence and Communication Networks (CICN)*, Dec 2015, pp. 1027–1032.

[2] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222–232, Feb 1987.

[3] T. F. Lunt, "A survey of intrusion detection techniques," *Comput. Secur.*, vol. 12, no. 4, pp. 405–418, Jun. 1993. [Online]. Available: http://dx.doi.org/10.1016/0167-4048(93)90029-5

[4] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network intrusion detection," *IEEE Network*, vol. 8, no. 3, pp. 26–41, May 1994.

[5] M. Xie, S. Han, B. Tian, and S. Parvin, "Anomaly detection in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, vol. 34, p. 1302–1325, 07 2011.

[6] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers and Security*, vol. 28, pp. 18–28, 02 2009.

[7] R. Bace and P. Mell, "Intrusion detection systems, national institute of standards and technology (nist)," Technical Report 800-31, Tech. Rep., 2001.

[8] K. Pelechrinis, M. Iliofotou, and S. V. Krishnamurthy, "Denial of service attacks in wireless networks: The case of jammers," *IEEE Communications Surveys Tutorials*, vol. 13, no. 2, pp. 245–257, Second 2011.

[9] Y. Tan, S. Sengupta, and K. P. Subbalakshmi, "Analysis of coordinated denial-of-service attacks in ieee 802.22 networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 890–902, April 2011.

[10] P. Stavroulakis and M. Stamp, *Handbook of Information and Communication Security*, 01 2010.

[11] A. G. Fragkiadakis, E. Z. Tragos, T. Tryfonas, and I. G. Askoxylakis, "Design and performance evaluation of a lightweight wireless early warning intrusion detection prototype," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 73, Mar 2012. [Online]. Available: https://doi.org/10.1186/1687-1499-2012-73

[12] J. Mar, I.-F. Hsiao, Y.-C. Yeh, C.-C. Kuo, and S.-R. Wu, "Intelligent intrusion detection and robust null defense for wireless networks," *International Journal of Innovative Computing, Information and Control*, vol. 8, pp. 3341–3359, 05 2012.

[13] K. Ali, A. Saidi, F. Bezzazi, M. El marraki, and A. Radi, "A new approach to intrusion detection system," *Journal of Theoretical and Applied Information Technology*, vol. 36, pp. 284–289, 02 2012.

Features	Presence of corresponding features in the list of independent variables												Freq.
logged in		■			■					■		■	4
protocol type								■					1
wrong fragment							■						1
flag		■		■								■	3
service			■				■						2
duration						■							1
Source bytes	■								■				2
destination bytes		■		■								■	3
hot							■						1
No failed logins			■	■			■			■			4
No root			■				■						2
No file creations		■	■					■					3
No shells					■		■				■		3
No access files					■					■			2
No outbound cmds					■					■		■	3
count				■									1
server count						■			■		■		3
same server rate	■									■			2
diff server rate			■									■	2
server diff host rate						■		■					2
destination host count									■			■	2
destination host server count				■								■	2
destination host same server rate												■	1
destination host diff server rate		■				■					■		3
destination host same Source port rate	■										■		2
destination host server diff host rate									■		■	■	3
destination host server error rate				■									1
destination host server rerror rate								■					1
Number of features	3	5	5	6	4	4	6	4	4	5	5	9	
AuC	**0.9**	**0.95**	**0.92**	**0.95**	**0.9**	**0.89**	**0.93**	**0.75**	**0.923**	**0.91**	**0.93**	**0.9**	
	0.30	0.32	0.31	0.32	0.30	0.30	0.31	0.25	0.31	0.30	0.31	0.30	

Fig. 2: AUC computed across the subset of features along with the frequency of corresponding feature

Features	Score
No failed logins	1.237
logged in	1.220
flag	0.933
destination bytes	0.933
destination host diff server rat	0.923
No shells	0.920
destination host server diff ho	0.918
server count	0.914
No outbound cmds	0.903
No file creations	0.873
service	0.617
No root	0.617
destination host server count	0.617
destination host same Source	0.610
Source bytes	0.608
destination host count	0.608
diff server rate	0.607
No access files	0.603
same server rate	0.603
server diff host rate	0.547
count	0.317
destination host server error r	0.317
wrong fragment	0.310
hot	0.310
destination host same server r	0.300
duration	0.297
protocol type	0.250
destination host server rerror	0.250

Fig. 3: List of the feature with decreasing order of their importance

[14] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in cloud," *Journal of network and computer applications*, vol. 36, no. 1, pp. 42–57, 2013.

[15] H. T. Elshoush and I. M. Osman, "Alert correlation in collaborative intelligent intrusion detection systems-a survey," *Appl. Soft Comput.*, vol. 11, no. 7, pp. 4349–4365, Oct. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2010.12.004

[16] S. Shanbhag and T. Wolf, "Accurate anomaly detection through parallelism," *IEEE Network*, vol. 23, no. 1, pp. 22–28, January 2009.

[17] C. Ho, Y. Lai, I. Chen, F. Wang, and W. Tai, "Statistical analysis of false positives and false negatives from real traffic with intrusion detection/prevention systems," *IEEE Communications Magazine*, vol. 50, no. 3, pp. 146–154, March 2012.

[18] A. Jain and D. Zongker, "Feature selection: evaluation, application, and small sample performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 153–158, Feb 1997.

[19] M. Kudo and J. Sklansky, "Comparison of algorithms that select features for pattern classifiers," *Pattern Recognition*, vol. 33, pp. 25–41, 01 2000.

[20] Y. Grandvalet and S. Canu, "Adaptive scaling for feature selection in svms," in *Proceedings of the 15th International Conference on Neural Information Processing Systems*, ser. NIPS'02. Cambridge, MA, USA: MIT Press, 2002, pp. 569–576. [Online]. Available: http://dl.acm.org/citation.cfm?id=2968618.2968689

[21] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1, pp. 389–422, Jan 2002. [Online]. Available: https://doi.org/10.1023/A:1012487302797

[22] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Berlin, Heidelberg: Springer-Verlag, 1995.

[23] Y. Li, J.-L. Wang, Z.-H. Tian, T.-B. Lu, and C. Young, "Building lightweight intrusion detection system using wrapper-based feature selection mechanisms," *Comput. Secur.*, vol. 28, no. 6, pp. 466–475, Sep. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.cose.2009.01.001

[24] H. Gharaee and H. Hosseinvand, "A new feature selection ids based on genetic algorithm and svm," in *2016 8th International Symposium on Telecommunications (IST)*. IEEE, 2016, pp. 139–144.

[25] R. R. Reddy, Y. Ramadevi, and K. N. Sunitha, "Effective discriminant function for intrusion detection using svm," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2016, pp. 1148–1153.

[26] M. Usha and P. Kavitha, "Anomaly based intrusion detection for 802.11 networks with optimal features using svm classifier," *Wireless Networks*, vol. 23, no. 8, pp. 2431–2446, 2017.

[27] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, "Building an intrusion detection system using a filter-based feature selection algorithm," *IEEE transactions on computers*, vol. 65, no. 10, pp. 2986–2998, 2016.

[28] M. E. Aminanto, H. Tanuwidjaja, P. D. Yoo, and K. Kim, "Weighted feature selection techniques for detecting impersonation attack in wi-fi networks," in *Proc. Symp. Cryptogr. Inf. Secur.(SCIS)*, 2017, pp. 1–8.

[29] B. Aslahi-Shahri, R. Rahmani, M. Chizari, A. Maralani, M. Eslami, M. J. Golkar, and A. Ebrahimi, "A hybrid method consisting of ga and svm for intrusion detection system," *Neural computing and applications*, vol. 27, no. 6, pp. 1669–1676, 2016.

[30] B. Senthilnayaki, K. Venkatalakshmi, and A. Kannan, "Intrusion detection using optimal genetic feature selection and svm based classifier," in *2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN)*. IEEE, 2015, pp. 1–4.

[31] A. Alazab, M. Hobbs, J. Abawajy, and M. Alazab, "Using feature selection for intrusion detection system," in *2012 International Symposium on Communications and Information Technologies (ISCIT)*. IEEE, 2012, pp. 296–301.

[32] K.-C. Khor, C.-Y. Ting, and S.-P. Amnuaisuk, "From feature selection to building of bayesian classifiers: A network intrusion detection perspective," *American Journal of applied sciences*, vol. 6, no. 11, p. 1948, 2009.

[33] Y.-Y. Zhou and G. Cheng, "An efficient network intrusion detection system based on feature selection and ensemble classifier," *arXiv preprint arXiv:1904.01352*, 2019.

[34] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," *Submitted to Second IEEE Symposium*

on Computational Intelligence for Security and Defense Applications (CISDA), vol. -, no. -, pp. –, 2009.

[35] L. M. Ibrahim, D. T. Basheer, and M. S. Mahmod, “A comparison study for intrusion database (kdd99, nsl-kdd) based on self organization map (som) artificial neural network,” *Journal of Engineering Science and Technology*, vol. 8, no. 1, pp. 107–119, 2013.

[36] P. Aggarwal and S. K. Sharma, “Analysis of kdd dataset attributes-class wise for intrusion detection,” *Procedia Computer Science*, vol. 57, pp. 842–851, 2015.

[37] B. Ingre and A. Yadav, “Performance analysis of nsl-kdd dataset using ann,” in *2015 International Conference on Signal Processing and Communication Engineering Systems*. IEEE, 2015, pp. 92–96.

[38] S. Lakhina, S. Joseph, and B. Verma, “Feature reduction using principal component analysis for effective anomaly–based intrusion detection on nsl-kdd,” 2010.

[39] S. Revathi and A. Malathi, “A detailed analysis on nsl-kdd dataset using various machine learning techniques for intrusion detection,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, no. 12, pp. 1848–1853, 2013.

[40] R. Zuech and T. M. Khoshgoftaar, “A survey on feature selection for intrusion detection,” in *Proceedings of the 21st ISSAT International Conference on Reliability and Quality in Design*, 2015, pp. 150–155.

[41] I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani, “Towards a reliable intrusion detection benchmark dataset,” *Software Networking*, vol. 2018, no. 1, pp. 177–200, 2018.

[42] D. D. Protić, “Review of kdd cup’99, nsl-kdd and kyoto 2006+ datasets,” *Vojnotehnički glasnik*, vol. 66, no. 3, pp. 580–596, 2018.

[43] L. Dhanabal and S. Shantharajah, “A study on nsl-kdd dataset for intrusion detection system based on classification algorithms,” *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 6, pp. 446–452, 2015.

[44] R. Zuech and T. Khoshgoftaar, “A survey on feature selection for intrusion detection,” pp. 150–155, 01 2015.