

Anomaly intrusion detection using machine learning- IG-R based on NSL-KDD dataset

Ashraf H. Aljammal¹, Ibrahim Al-Oqily², Mamoon Obiedat², Ahmad Qawasmeh¹, Salah Taamneh¹, Fadi I. Wedyan³

¹Department of Computer Science and Applications, Prince Al Hussein bin Abdullah II Faculty of Information Technology, The Hashemite University, Zarqa, Jordan

²Department of Information Technology, Prince Al Hussein bin Abdullah II Faculty of Information Technology, The Hashemite University, Zarqa, Jordan

³Department of Engineering, Computing, and Mathematical Sciences, Lewis University, Romeoville, USA

Article Info

Article history:

Received Aug 1, 2023

Revised May 4, 2024

Accepted Jun 1, 2024

Keywords:

Anomaly detection

Cyber security

Intrusion detection

Machine learning

Network security

Network security lab-

knowledge discovery dataset

ABSTRACT

Cybersecurity is challenging for security guards because of the rising quantity, variety, and frequency of attacks and malicious activities in cyberspace. Intrusion attacks are among the most common types of cyberspace attacks. Therefore, an intrusion detection system (IDS) is in high demand to accurately detect and mitigate their impact. In this paper, an anomaly IDS using machine learning and information gain-rank (IG-R) is proposed to improve the detection accuracy of intrusions. The network security lab-knowledge discovery dataset (NSL-KDD) is used to train and test the proposed IDS. Initially, the information gain (IG) algorithm and Ranker are used to evaluate, rank and reduce the number of selected instances from 41 instances to only 6 instances. Furthermore, many classifiers have been tested and evaluated; such as adaptive boosting (AdaBoostM1), random forest, J48, and naïve Bayes to choose the best performance classifier to be used in the detection process. After applying the IG-R and testing the suggested classifiers, the results showed that the random forest classifier has the best performance over the tested classifiers with TPR, FPR, and accuracy of 99.7%, 0.3%, and 99.7%, respectively, and is recommended to be used in the detection process.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Ashraf H. Aljammal

Department of Computer Science and Applications

Prince Al Hussein bin Abdullah II Faculty of Information Technology, The Hashemite University

Zarqa, Jordan

Email: ashrafj@hu.edu.jo

1. INTRODUCTION

The architecture of computer networks continues to suffer from inadvertently left vulnerabilities, either because of bad design or because of the nature of the used protocols and softwares over network or cyberspace, and occasionally due to human illegal conduct [1]-[3]. Regardless of the causes of these vulnerabilities, it draws attention to cybersecurity breaches, where attackers can exploit these flaws and infiltrate the computer network and its devices (i.e., zero-day-attacks) [4], [5]. Attacks might vary from just investigating the contents and files of network devices to causing substantial harm to the network devices and contents [6]-[8]. Therefore, a powerful and accurate intrusion detection system (IDS) is required to identify attackers' activity and handle them using additional security measures [9]. The term intrusion refers to any unauthorized activity on a computer network that poses a risk on the targeted devices and networks.

Researchers utilized many approaches to develop accurate IDS systems; meanwhile neural networks and data mining are two of the most recently used approaches to detect intrusion activities. The IDS is a software able to distinguish normal traffic of the network from malicious traffic [10]. For this purpose, two types of detection systems can be used: anomaly-based detection systems and signature-based detection systems [11]-[13].

Anomaly-based detection systems are behavioral detection systems that rely on determining the divergence of collected traffic from the network's typical behavior. As a result, the IDS must first understand (learn) the regular behavior of the environment it monitors. These systems are capable of recognizing and detecting novel sorts of attacks (zero-day-attacks).

Signature-based detection systems compare collected traffic to previously known signatures of attacks stored in a database. These systems are quite effective at identifying known attacks. The following is the structure of the paper. Section 2 goes through some of the existing IDSs that employ machine-learning techniques. Sections 3 and 4 examines the network security lab-knowledge discovery dataset (NSL-KDD) dataset that was used to evaluate the tested techniques and outlines the assessment matrix that was used to evaluate the suggested model. The suggested model is presented in section 4. Section 5 displays the experimental findings of the suggested model's evaluation. Section 6 concludes our work.

2. RELATED WORK

Many researchers have adopted machine-learning techniques to build IDSs due to their ability to handle datasets with a huge number of instances and attributes. In addition, it is able to improve the detection accuracy of the attacks compared to the other techniques [14]. However, Kaja *et al.* [15] have proposed two-stage intelligent IDS using machine-learning algorithms. K-means was used in the first stage to detect the attacks. However, in the second stage supervised learning algorithms were used to classify the types of attacks. C5 classifier-based IDS was proposed to detect normal as well as abnormal activities using the NSL-KDD dataset [16]. The aim was to increase the detection accuracy and reduce the false alarms rates. A hybrid clustering and autocorrelation function-based IDS has been proposed to handle series and nonseries data [17]. Unsupervised techniques have been used to categorize the collected data from host intrusion detection systems (HIDSs) and NIDSs based on domain similarity. Gad *et al.* [18] have proposed a machine learning based IDS using the extreme gradient boosting (XGBoost) method to detect attacks over vehicular ad hoc networks (VANETs). The ToN-IoT dataset was used to train and test the proposed IDS. In addition, Chi-square was used to select the significant instances to be used in the detection process. Furthermore, the SMOTE technique was used for class balancing to reduce the bias of the dominant class. An IDS model has been proposed to detect abnormal activities over wireless sensor networks (WSN) [19]. The WSN-DS dataset was used to train and test the proposed model (ID-GOPA). Information gain (IG) algorithm and online passive aggressive algorithm were used for instances selection and detection of the DoS attack types respectively. However, a deep neural network (DNN) based IDS has been proposed in [20] to detect intrusions over WSN. The cross-correlation method was used to select the appropriate instances from the used dataset. Eventually, these instances were employed as building blocks for the DNN algorithm, which was used in the detection process of the intrusions. In this paper, a subset of NSL-KDD dataset instances will be used. The selection of the instance's subset is based on the IG algorithm. In addition, machine-learning classifiers will be studied such as AdaBoostM1, random forest, J48, and naïve Bayes. Furthermore, to evaluate the performance of the classifiers we used true positive rate (TPR), false positive rates (FPR), and accuracy measurements.

3. DATASET, ADVERSARY MODEL, AND CONFUSION MATRIX

The suggested model is trained and tested using the NSL-KDD. Researchers use it extensively to train, test, and assess the performance of IDSs. The NSL-KDD dataset is an improved version of the KDD'99 dataset. The key improvement was to eliminate duplicate records in both the training and testing datasets to reduce classifier and learner bias [21]. The dataset contains 125973 instances divided into two categories: normal (67,343 instances) and attack (58,630 instances). Furthermore, the attacks are divided into four sub-categories: DoS, R2L, U2R, and probing. Each record, as shown in Table 1, has 41 basic characteristics omitting the class property. In addition, these characteristics are classified into four groups, as shown in Table 2. We analyzed these attributes (instances) in this study and chose the most effective ones to increase detection accuracy.

Table 1. Features included in NSL-KDD dataset

Feature-no	Feature-name	Feature-no	Feature-name
1	duration	22	is_guest_login
2	protocol_type	23	count
3	service	24	srv_count
4	flag	25	serror_rate
5	src_bytes	26	srv_serror_rate
6	dst_bytes	27	rerror_rate
7	land	28	srv_rerror_rate
8	wrong_fragment	29	same_srv_rate
9	urgent	30	diff_srv_rate
10	hot	31	srv_diff_host_rate
11	num_failed_logins	32	dst_host_count
12	logged_in	33	dst_host_srv_count
13	num_compromised	34	dst_host_same_srv_rate
14	root_shell	35	dst_host_diff_srv_rate
15	su_attempted	36	dst_host_same_src_port_rate
16	num_root	37	dst_host_srv_diff_host_rate
17	num_file_creations	38	dst_host_serror_rate
18	num_shells	39	dst_host_srv_serror_rate
19	num_access_files	40	dst_host_rerror_rate
20	num_outbound_cmds	41	dst_host_srv_rerror_rate
21	is_host_login		

Table 2. NSL-KDD dataset features categories

Features	Category
F1, F2, F3, F4, F5, F6, F7, F8, F9, F10	Basic features
F11, F12, F13, F14, F15, F16, F17, F18, F19, F20, F21, F22	Content features
F23, F24, F25, F26, F27, F28, F29, F30, F31	Time-based features
F32, F33, F34, F35, F36, F37, F38, F39, F40, F41	Host-based features

The widely used confusion matrix is applied in our testing environment to evaluate and compare the prediction performance of the tested classifiers. Following are the formulas used for the evaluation.

$$Accuracy = \frac{TP+TN}{(TP+TN+FN+FP)} \quad (1)$$

$$TP\ rate = \frac{TP}{TP+FN} \quad (2)$$

$$FP\ rate\ (Recall) = \frac{FP}{FP+TN} \quad (3)$$

Where TP is the number of records correctly classified as attacks; TN is the number of records correctly classified as normal; FP is the number of normal records incorrectly classified as attack; and FN is the number of attacks incorrectly classified as normal.

4. TESTING MODEL

The suggested model is discussed in this section. Subsections 4.1 and 4.2 go over the testing model in detail. The testing model is depicted in Figure 1 and the pseudocode of the testing model is depicted in Algorithm 1. Using the NSL-KDD dataset, the WEKA platform [22] is utilized to evaluate and assess the tested classifiers.

4.1. Data preprocessing and instances selection

During this step, we decreased the amount of attributes that might be employed in the proposed model's training and testing. The (IG) method is used in combination with Ranker to analyze the merit of the qualities and rank the 41 attributes based on their individual ratings. In (4) illustrates the (IG) algorithm.

$$InfoGain(Class, Attribute) = H(Class) - H(Class | Attribute) \quad (4)$$

Where H represents the entropy; class represents whether normal or attack; and attribute: denotes the 41 attributes (features) shown in Table 1.

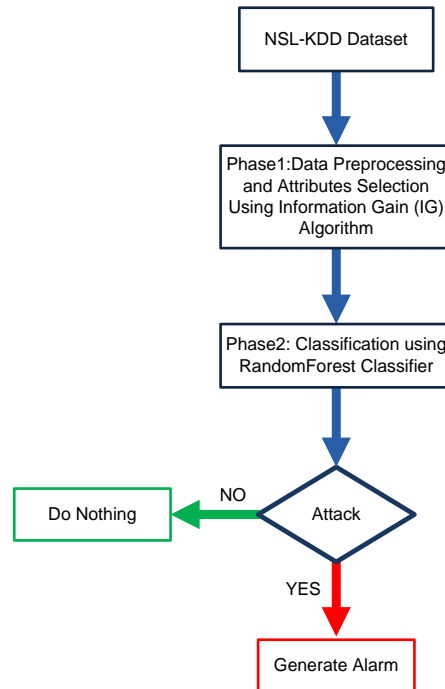


Figure 1. Testing model

Algorithm 1. Proposed model

```

1: Procedure model ()
2: Input=NSL-KDD dataset containing 41 attributes x1, x2...x41
3: Reduce 41 features to 6 features using IG algorithm and Rnker
4: Use classifier
5: Propose the model M
6: For every feature xn
7: Provide xn for AdaBoostM1, RandomForest, J48, NaiveBayes using NSL-KDD
   dataset
8: Calculate TPR1-4, FPR1-4, Accuracy1-4 for
9: 1-AdaBoostM1
10: 2-RandomForest
11: 3-J48
12: 4-NaiveBayes
13: Compare TPR1-4, FPR1-4, Accuracy1-4
14: Select the best performance model M= RandomForest
  
```

The selection of attributes with ranks of 0.5 and above were chosen to be used in the proposed model's training and testing stages. According to the IG algorithm and ranking results, six attributes matched the condition: service, flag, src_byte, dst_byte, same_srv_rate, and diff_srv_rate. The decision tree classifiers are only able to deal with numeric values. Therefore, the selected instances with nominal values have been converted into numerical values. Where F1 and F2 are the selected instances having nominal values

4.2. Data classification

In this step, the dataset was divided into 80% training and 20% testing, with 100778 and 25195 instances, respectively. However, numerous classifiers have been tested on the NSL-KDD dataset, including adaptive boosting (AdaBoostM1), random forest, J48, and naïve Bayes. The goal of testing them is to find the best classifier to employ in the proposed model. Testing was conducted using the six selected attributes based on the IG algorithm. In this step, the classifier with the greatest TPR and lowest FPR as well as the best accuracy was chosen. The testing results for the aforementioned algorithms are shown below.

4.2.1. Adaptive boosting

Freund and Schapire introduced the AdaBoost algorithm in 1995 [23] to handle multiclass dataset problems. It was tested in this section using the NSL-KDD dataset with the following settings.

```

Classifier: weka.classifiers.meta.Bagging - P 100 - S 1 - num
-slots 1 - I 10 - W weka.classifiers.trees.REPTree - - - M 2
  
```

–V 0.001 – N 3 – S 1 – L – 1 – I 0.0

The findings revealed that the TP rate of detecting attacks was 91.4%, with an FP rate of 7.0% and an accuracy of 92.2%. Figure 2 shows the AdaBoostM1 classifier performance results.

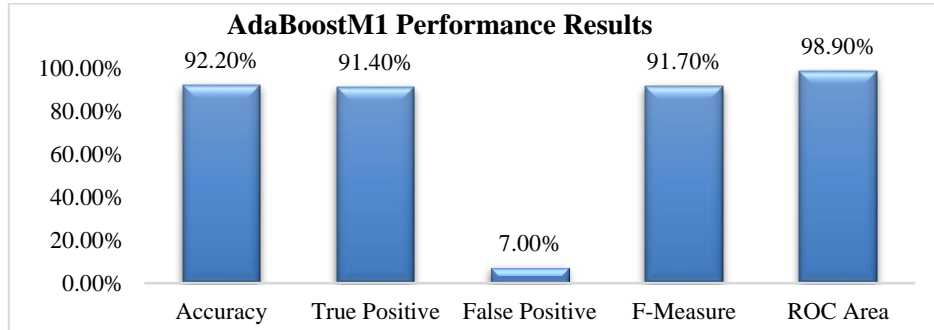


Figure 2. AdaBoostM1 performance

4.2.2. Random forest

Random forest is a tree-based classifier developed by Leo Breimans in 1996 that is considered as one of the most important algorithms in the field of neural networks [24].

Classifier: weka.classifiers.trees.Randomforest – P 100 – I 100
– num – slots 1 – K 0 – M 1.0 – V 0.001 – S 1

The attacks detection rates were 99.7%, 0.3%, and 99.7% TP rate, FP rate and accuracy respectively. Figure 3 shows the random forest classifier performance results.

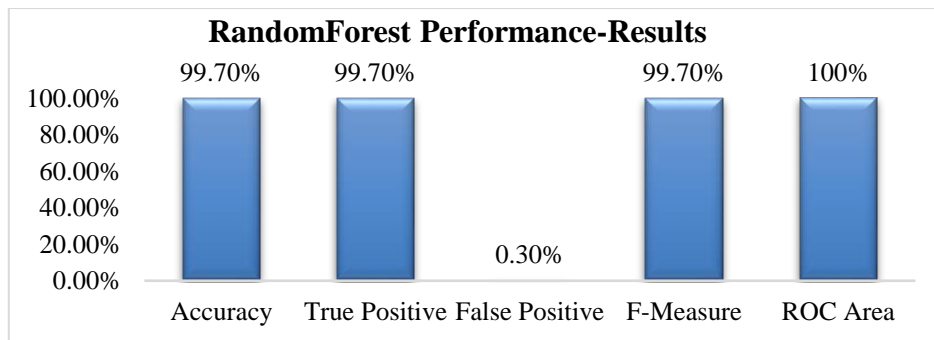


Figure 3. Random forest performance

4.2.3. J48

J48 is decision tree algorithm, which is developed by Quinlan and considered a statistical classifier since it depends on labeled input data in making decisions [25]. It has been tested over the dataset using the following configurations.

Classifier: weka.classifiers.trees.RandomForest – P 100 – I 100 – num – slots 1 – K 0 – M 1.0
– V 0.001 – S 1

The results indicated that it detected attacks with a 99.6%, 0.3%, and 99.6% TP rate, FP rate, and accuracy, respectively. The J48 classifier performance results are shown in Figure 4.

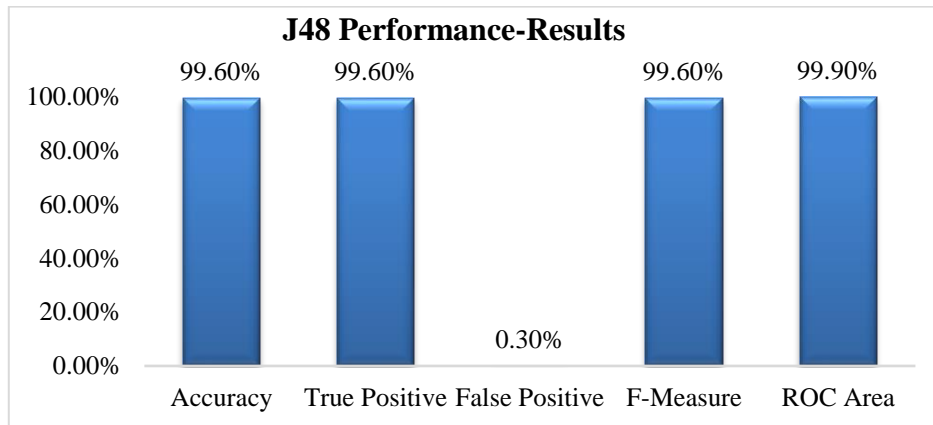


Figure 4. J48 performance

4.2.4. Naïve Bayes

In the field of machine learning and data mining, naïve Bayes is considered one of the most efficient and effective inductive learning algorithms. In classification, naïve Bayes employs a probabilistic approach based on the Bayes theorem, which results in good performance [26]. Following are the configurations used to test naïve Bayes over the dataset.

Classifier: weka.classifiers.bayes.NaiveBayes

It detected the attacks with a TP rate of 75.8%, FP rate of 3.7%, and accuracy of 86.6%, respectively. Figure 5 shows the naïve Bayes classifier performance results.

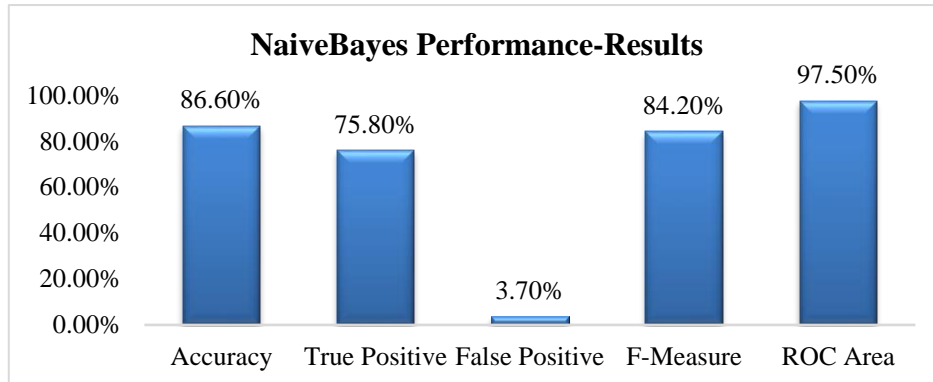


Figure 5. Naïve Bayes results

5. RESULTS EVALUATION AND ANALYSIS

The proposed model will be evaluated using the following metrics: TPR, FPR, and accuracy. Among the tested classifiers, the one with the highest TPR and accuracy and the lowest FPR is considered better. According to the results of the experiments, the random forest classifier outperformed the other classifiers with 99.7%, 0.3%, and 99.7% TP rate, FP rate, and accuracy, respectively. While the naïve Bayes classifier performed the poorest, with 75.8%, 3.7%, and 86.6% TP rate, FP rate, and accuracy, respectively. Although the J48 classifier produced results that were extremely close to those of random forest, random forest was better. As a result, the random forest classifier is recommended to be employed in the classification process of the IDS model. Figure 6 depicts a comparison of the tested classifiers. Furthermore, reducing the number of instances from 41 to 6 has not only boosted detection accuracy but has the potential to decrease detection time by about 96%, as shown in Table 3.

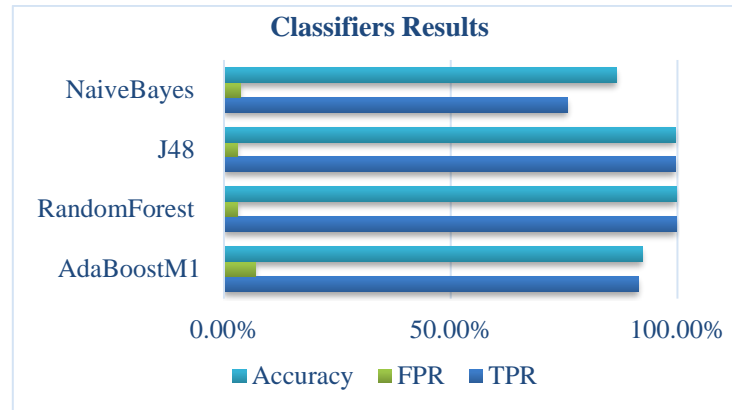


Figure 6. Classifiers results comparison

Table 3. Testing time of random forest classifier

No. of features	Testing-time (sec)
All dataset features (41)	0.8
The selected features (6)	0.45

6. CONCLUSION

In this paper, an intrusion detection technique is proposed with the use of a random forest classifier in addition to IG and Ranker to reduce the number of 41 instances to only 6 instances. NSL-KDD dataset is used to train and test the proposed IDS. We compared the performance results of four classifiers; AdaBoostM1, random forest, J48, and naïve Bayes. The IDS using random forest classifier has shown the highest accuracy 99.7% in performance results. Moreover, it showed a 99.7% and 0.3% TPR and FPR respectively. In addition, the detection speed using random forest based on 6 instances has been increased by 96% compared to the detection speed using 41 instances




REFERENCES

- [1] X. Ge and M. Yue, "Research on the application of computer network security and practical technology in the era of big data," in *International Conference on Machine Learning and Big Data Analytics for IoT Security and Privacy*, 2021: Springer, pp. 505-510, doi: 10.1007/978-3-030-89508-2_64.
- [2] I. Obeidat, A. Mughaid, and S. Alzoubi, "A secure encrypted protocol for clients' handshaking in the same network," *International Journal of Interactive Mobile Technologies*, vol. 13, no. 5, pp. 47-57, 2019, doi: 10.3991/ijim.v13i05.9845.
- [3] A. H. Aljammal, S. Taamneh, A. Qawasmeh, and H. B. Salameh, "machine learning based phishing attacks detection using multiple datasets," *International Journal of Interactive Mobile Technologies*, vol. 17, no. 5, pp. 71-83, 2023, doi: 10.3991/ijim.v17i05.37575.
- [4] N. Gupta, V. Jindal, and P. Bedi, "CSE-IDS: using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems," *Computers & Security*, vol. 112, p. 102499, 2022, doi: 10.1016/j.cose.2021.102499.
- [5] S. A. M. Al-Juboori, F. Hazzaa, Z. S. Jabbar, S. Salih, and H. M. Gheni, "Man-in-the-middle and denial of service attacks detection using machine learning algorithms," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 1, pp. 418-426, 2023, doi: 10.11591/eei.v12i1.4555.
- [6] A. H. Aljammal, H. Bani-Salameh, A. Qawasmeh, A. Alsarhan, and A. F. Otoom, "A new technique for data encryption based on third party encryption server to maintain the privacy preserving in the cloud environment," *International Journal of Business Information Systems*, vol. 28, no. 4, pp. 393-403, 2018, doi: 10.1504/IJBIS.2018.093654.
- [7] A. H. Aljammal, H. Bani-Salameh, A. Alsarhan, M. Kharabsheh, and M. Obiedat, "Node verification to join the cloud environment using third party verification server," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 11, no. 4, pp. 55-65, 2017, doi: 10.3991/ijim.v11i4.6501.
- [8] W. Zhang, "Design of computer network security monitoring system based on programming language," in *International Conference on Machine Learning and Big Data Analytics for IoT Security and Privacy*, 2021: Springer, pp. 401-408, doi: 10.1007/978-3-030-89511-2_51.
- [9] W. Wang, S. Jian, Y. Tan, Q. Wu, and C. Huang, "Representation learning-based network intrusion detection system by capturing explicit and implicit feature interactions," *Computers & Security*, vol. 112, p. 102537, 2022, doi: 10.1016/j.cose.2021.102537.
- [10] M. O. Okay, Ö. Aslan, R. Eryigit, and R. Samet, "SABADT: hybrid intrusion detection approach for cyber attacks identification in WLAN," *IEEE Access*, vol. 9, pp. 157639-157653, 2021, doi: 10.1109/ACCESS.2021.3129600.
- [11] P. Ioulianiou, V. Vasilakis, I. Moscholios, and M. Logothetis, "A signature-based intrusion detection system for the internet of things," *Information and Communication Technology Form*, 2018, *In press*.
- [12] I. G. A. Poornima and B. Paramasivan, "Anomaly detection in wireless sensor network using machine learning algorithm," *Computer Communications*, vol. 151, pp. 331-337, 2020, doi: 10.1016/j.comcom.2020.01.005.




- [13] H. A. Al Essa and W. S. Bhaya, "Ensemble learning classifiers hybrid feature selection for enhancing performance of intrusion detection system," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 1, pp. 665-676, 2024, doi: 10.11591/eei.v13i1.5844.
- [14] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 493-501, 2019, doi: 10.1007/s12083-017-0630-0.
- [15] N. Kaja, A. Shaout, and D. Ma, "An intelligent intrusion detection system," *Applied Intelligence*, vol. 49, no. 9, pp. 3235-3247, 2019, doi: 10.1007/s10489-019-01436-1.
- [16] A. Khraisat, I. Gondal, and P. Vamplew, "An anomaly intrusion detection system using C5 decision tree classifier," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2018: Springer, pp. 149-155, doi: 10.1007/978-3-030-04503-6_14.
- [17] K. Kumar, A. Kumar, V. Kumar, and S. Kumar, "A hybrid classification technique for enhancing the effectiveness of intrusion detection systems using machine learning," *International Journal of Organizational and Collective Intelligence (IJOICI)*, vol. 12, no. 1, pp. 1-18, 2022, doi: 10.4018/IJOICI.2022010102.
- [18] A. R. Gad, A. A. Nashat, and T. M. Barkat, "intrusion detection system using machine learning for vehicular ad hoc networks based on ToN-IoT Dataset," *IEEE Access*, vol. 9, no. 1, pp. 142206-142217, 2021, doi: 10.1109/ACCESS.2021.3120626.
- [19] S. Ifzame, H. Tabbaa, I. Hafidi, and N. Lamghari, "Anomaly detection using machine learning techniques in wireless sensor networks," in *Journal of Physics: Conference Series*, 2021, vol. 1743, no. 1: IOP Publishing, p. 012021, doi: 10.1088/1742-6596/1743/1/012021.
- [20] V. Gowdhaman and R. Dhanapal, "An intrusion detection system for wireless sensor networks using deep neural network," *Soft Computing*, no. 1, pp. 1-9, 2021, doi: 10.1007/s00500-021-06473-y.
- [21] M. Tavallae, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*, 2009: IEEE, pp. 1-6, doi: 10.1109/CISDA.2009.5356528.
- [22] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, "The WEKA data mining software: An update. *ACM SIGKDD Explorations*. 2009; vol. 11, no. 1, pp. 10-18, doi: 10.1145/1656274.1656278
- [23] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119-139, 1997, doi: 10.1006/jcss.1997.1504.
- [24] Z. Zhang and X. Xie, "Research on adaboost. m1 with random forest," in *2010 2nd International Conference on Computer Engineering and Technology*, 2010, vol. 1: IEEE, pp. 647-652, doi: 10.1109/ICCET.2010.5485910.
- [25] J. Han, M. Kamber, and J. Pei, *Data mining concepts and techniques third edition*, 3rd ed. (University of Illinois at Urbana-Champaign Micheline Kamber Jian Pei Simon Fraser University). Elsevier Science Ltd Publication, 2012, pp. 703, doi: 10.1016/C2009-0-61819-5
- [26] H. Zhang, C. X. Ling, and Z. Zhao, "The learnability of naive Bayes," in *Advances in Artificial Intelligence: 13th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, AI 2000 Montréal, Québec, Canada, May 14-17, 2000 Proceedings 13*, 2000: Springer, pp. 432-441, doi: 10.1007/3-540-45486-1_37.

BIOGRAPHIES OF AUTHORS






Ashraf H. Aljammal    is currently an Associate Professor at the Department of Computer Science and Applications, The Hashemite University, Zarqa, Jordan. He received the B.S. degree in computer science from Albalqa' Applied University, Al-Salt, Jordan, in 2006, the master's degree from Universiti Sains Malaysia, USM, Malaysia, in 2007, and the Ph.D. degree from Universiti Sains Malaysia, USM, Malaysia, in 2011. His research interests include but not limited to network security, cyber security, IoT security, network monitoring, cloud computing, machine learning, and data mining. He can be contacted at email: ashrafj@hu.edu.jo.






Ibrahim Al-Oqily    is an Associate Professor in the Faculty of Prince Al-Hussein Bin Abdallah II for Information Technology. He obtained his B.S. in computer science, Jordan University, Alkarak, Jordan, 1989-1993. He got his M.Sc. in computer science, Jordan University, Amman, Jordan, 2001-2003 and his Ph.D. in computer science, Ottawa University, Ottawa, ON, Canada, 2004-July 2008. He served three years as a vice dean of the IT faculty, chairman of computer information systems department, chairman of software engineering department, and chairman of computer science department. He has 10 years of professional experience in teaching university level courses, five years of professional experience in teaching computer software, and four years professional experience in application development testing and system analyst. He is an IEEE senior member and an active IEEE member since 2005. He is the co-founder of the Jordanian programming association and the YU-IEEE student branch. His current research interests include digital forensics, cyber security, autonomous and adaptive systems, parallel and distributed systems, highly parallel systems, networks management and policy-based networks, mobile computing, overlay networks and service-oriented systems, autonomic computing, and cloud computing. He can be contacted at email: izalqily@hu.edu.jo.






Mamoon Obiedat    received B.Sc. in computer science and M.Sc. in computer information systems from Yarmouk University, Jordan in 1992, 2005, respectively. He was a Lecturer at Al-Balqa Applied University in Jordan from 1998 until he received Ph.D. degree in computer science from Lincoln University, New Zealand in 2014. He has been a member in Centre for Advanced Computational Solutions (CFACS) at Lincoln University since 2011. His research interests lie in soft computing, fuzzy cognitive maps, data mining, and decision support systems. He is also interested in 3D image processing with MATLAB and Simulink. He also works on modeling of complex real-world problems. He is currently an Associate Professor at the Department of Information Technology in Hashemite University. He can be contacted at email: mamoon@hu.edu.jo.






Ahmad Qawasmeh    is a native of Jordan where he studied computer engineering. He obtained his M.S. degree in computer science in 2010 and completed his Ph.D. on performance analysis support for HPC applications in computer science from the University of Houston in 2015. His research interests include parallel programming languages, performance analysis, and machine learning. He joined The Hashemite University, Jordan in 2016 as an Assistant Professor in the Department of Computer Science. He can be contacted at email: ahmad@hu.edu.jo.



Salah Taamneh    is currently an Associate Professor at the Department of Computer Science and its Applications, Hashemite University, Zarqa, Jordan. He received the B.S. degree in computer science from Jordan University of Science and Technology, Irbid, Jordan, in 2005, the M.S. degree in computer science from Prairie View A&M University, Prairie View, Texas, in 2011 and the Ph.D. degree in computer science from University of Houston, Houston, Texas, USA, in 2016. His current research interests include parallel and distributed computing, machine learning, and human-computer interaction. He can be contacted at email: taamneh@hu.edu.jo.



Fadi I. Wedyan    joined the Department of Computer and Mathematical Sciences at Lewis University, Illinois in 2021. He was a visiting Associate Professor at the Department of Math. and Computer Science, Duquesne University, Pittsburgh, Pennsylvania. He also was an Associate Professor at the Department of Software Engineering, Hashemite University. His research interests include: evolutionary software testing, search-based software engineering, software quality metrics, and software design. His interests also include AI applications mainly planning and scheduling, and classification. He is also interested in mobile computing and the design and development of smartphone applications for health care, educational, and social uses. He can be contacted at email: fadi.wedyan@hu.edu.jo.