Online-Journals.org

iJOE International Journal of Online and Biomedical Engineering

iJOE | eISSN: 2626-8493 | Vol. 20 No. 3 (2024) | OPEN ACCESS

https://doi.org/10.3991/ijoe.v20i03.45249

PAPER

Web Attack Intrusion Detection System Using Machine Learning Techniques

Mahmoud Khalid Baklizi[1], Issa Atoum[2], Mohammad Alkhazaleh[1], Hasan Kanaker[3], Nibras Abdullah[4,5](✉), Ola A. Al-Wesabi[5], Ahmed Ali Otoom[6]

[1]Department of Computer Sciences, Faculty of Information Technology, Isra University, Amman, Jordan

[2]Software Engineering Department, Faculty of Information Technology, The World Islamic Sciences and Education, Amman, Jordan

[3]Department of Cybersecurity, Faculty of Information Technology, Isra University, Amman, Jordan

[4]School of Computer Sciences, Universiti Sains Malaysia (USM), Gelugor, Malaysia

[5]Faculty of Computer Science and Engineering, Hodeidah University, Hodeidah, Yemen

[6]Cybersecurity and Cloud Computing Department, Faculty of Information Technology, Applied Science Private University, Amman, Jordan

nibras@usm.my

ABSTRACT

Web attacks often target web applications because they can be accessed over a network and often have vulnerabilities. The success of an intrusion detection system (IDS) in detecting web attacks depends on an effective traffic classification system. Several previous studies have utilized machine learning classification methods to create an efficient IDS with various datasets for different types of attacks. This paper utilizes the Canadian Institute for Cyber Security's (CIC-IDS2017) IDS dataset to assess web attacks. Importantly, the dataset contains 80 attributes of recent assaults, as reported in the 2016 McAfee report. Three machine learning algorithms have been evaluated in this research, namely random forests (RF), k-nearest neighbor (KNN), and naive bayes (NB). The primary goal of this research is to propose an effective machine learning algorithm for the IDS web attacks model. The evaluation compares the performance of three algorithms (RF, KNN, and NB) based on their accuracy and precision in detecting anomalous traffic. The results indicate that the RF outperformed the NB and KNN in terms of average accuracy achieved during the training phase. During the testing phase, the KNN algorithm outperformed others, achieving an average accuracy of 99.4916%. However, RF and KNN achieved 100% average precision and recall rates compared to other algorithms. Finally, the RF and KNN algorithms have been identified as the most effective for detecting IDS web attacks.

KEYWORDS

intrusion detection systems, CIC-IDS2017, machine learning, false alarms, naive bayes (NB), k-nearest neighbors (KNN), random forest (RF)

1 INTRODUCTION

Despite the tremendous development of the Internet, it is still vulnerable to security issues in web-based applications. These vulnerabilities give hackers the ability to carry out various web attacks, such as SQL injection, XSS, and brute force [1]. In web attacks, hackers exploit vulnerabilities to gain unauthorized access to web

Baklizi, M.K., Atoum, I., Alkhazaleh, M., Kanaker, H., Abdullah, N., Al-Wesabi, O.A., Otoom, A.A. (2024). Web Attack Intrusion Detection System Using Machine Learning Techniques. *International Journal of Online and Biomedical Engineering (iJOE)*, 20(3), pp. 24–38. https://doi.org/10.3991/ijoe.v20i03.45249

Article submitted 2023-09-24. Revision uploaded 2023-11-15. Final acceptance 2023-11-23.

servers, network devices, or the network itself [2]. Hence the importance of intrusion detection systems, which play a crucial role in addressing these attacks and protecting against them [3]. Moreover, successful detection of new attacks requires a vast amount of data to create models of normal behavior and anomalies [4]. The need arises to utilize intrusion detection systems (IDSs) for training on a compelling dataset [3]. This highlighted the use of supervised machine learning algorithms to effectively analyze the data and create a predictive model that can accurately predict new attacks [4]. An intelligent approach to identifying new attack types involves using machine learning in combination with feature selection methods and classification algorithms [2].

Machine learning classification techniques have been utilized in numerous published papers to develop effective IDS using diverse datasets for different types of attacks. The dataset for wireless sensor network intrusion detection is referred to as WSN_DS. The dataset represents a variety of denial of service (DoS) attacks, including blackhole, flooding, gray hole, and scheduling attacks, with 19 features and 374,661 records [5, 6]. The KDD'99 dataset was created by simulating routine and traffic attacks in a military environment, specifically the US Air Force LAN [4]. It has 41 features related to traffic, content, and intrinsic attacks, corresponding to four different categories of attacks [4, 7, 8]: DoS, U2R, Prob, and R2L. The NSL KDD dataset aims to overcome the challenges of redundancy, duplication, and data imbalance found in the KDD 99 dataset [4]. The UNSW Canberra Cyber Range Lab's IXIA Perfect Storm program developed the UNSW Nb15 dataset to provide a blend of real-world normal behaviors and simulated modern attack behaviors [4, 9]. The dataset includes nine types of attacks, represented by 49 characteristics. Approximately 82,000 records are used for testing, while 175,000 records are used for training. The types of attacks are: fuzzers, analysis, DoS, backdoor, exploit, generic, reconnaissance, shellcode, and worm. [4, 10, 11]. Another IDS dataset used to evaluate web attacks in this research is the Canadian Institute for Cybersecurity (CIC-IDS2017). The dataset represents recent cyberattacks as documented in the 2016 McAfee report [12]. The dataset includes 80 features extracted from network traffic, representing prevalent attack types such as Heartbleed, DoS, Brute Force SSH, Infiltration Brute Force FTP, Web Attack, DDoS, and Botnet [9, 12]. The rest of this article's research is divided into the following sections: An overview of the literature review is presented in Section II. Section III presents the methodology of the proposed strategy. Results from the simulation are compared and discussed in Section IV. A conclusion is provided at the end.

2 LITERATURE REVIEW

2.1 Web attacks

A web attack exploits website vulnerabilities to gain unauthorized access and obtain confidential information. There are numerous web attacks that hackers have employed. This research will discuss the most common web attacks currently, including brute force, SQL injection (SQLIA), and cross-site cite scripting (XSS).

- **Brute force attacks**

In a brute-force attack, attackers or crackers attempt all possible password combinations in order to crack the file [13]. This is a simple yet effective method for gaining unauthorized access to personal accounts, business systems, and networks [14]. "Brute force" refers to attacks that use excessive force to access user accounts.

The size of the key space has a significant impact on the likelihood of a brute-force attack [14]. This implies that brute-force attacks can only succeed when using short keys, as longer keys result in exponentially larger key spaces.

- **Attacks using cross-site scripting (XSS)**

A web application's most common vulnerability is cross-site scripting (XSS) [15]. This is an injection problem that allows malicious scripts to be injected into trusted websites viewed by other users [16]. XSS attacks enable an attacker to compromise data and steal cookies, credit card numbers, passwords, and other sensitive information by allowing the attacker to execute malicious scripts on the victim's web browser [17].

- **SQL injection attack**

By exploiting application vulnerabilities, web attacks known as SQL injection attacks (or SQLIA) alter SQL queries and inject malicious code [18, 19]. These attacks pose a significant threat to any web application that utilizes user input to create SQL queries for an underlying database [20].

Successful SQLIA attacks enable attackers to modify database information, manipulate the database, and retrieve system files [21]. They also grant them access to sensitive data. Attackers may, in some cases, issue commands to the operating system of the database [22, 18].

2.2 Machine learning algorithms

Machine learning can be a potent tool for accurately detecting new web attacks [23–25]. It is a subfield of artificial intelligence that involves constructing models using algorithms trained on specific data and then applying those models to other data to make predictions [26]. There are several different machine learning classifiers, including random forest (RF), k-nearest neighbors (KNN), and Naive Bayes (NB).

- **Random forest**

Academics are interested in the RF algorithm due to its speed and accuracy in categorization. In predictive modeling and machine learning approaches, the RF involves a collection of supervised learning procedures for regression and classification [27]. It aggregates the results and predictions from multiple decision trees [26] to select the optimal output, as illustrated in Figure 1.

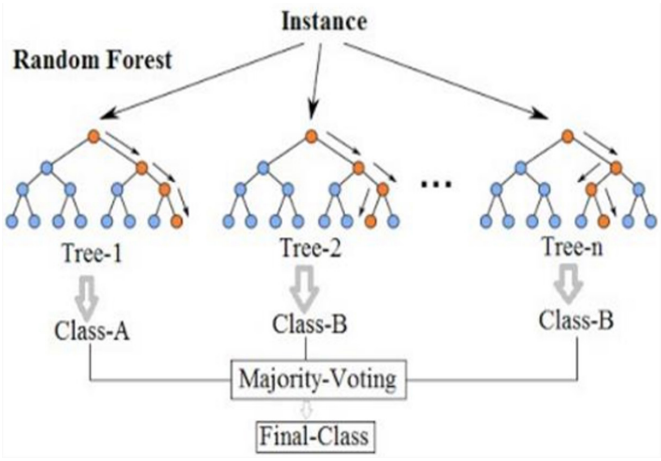

Fig. 1. Random forest

- **K-nearest neighbor**

The k-nearest neighbors (KNN) algorithm is considered one of the most fundamental classification techniques in machine learning, specifically in supervised learning [28, 29]. Using KNN can yield a respectable level of accuracy in making predictions through classification [28].

The KNN algorithm heavily relies on the value of k to determine the number of neighbors that should be selected based on the input data, as shown in Figure 2 [30]. The KNN algorithm is simple and easy to understand and execute [28, 29]. However, as the amount of data being used increases, finding the ideal value of k causes KNN to become slower [30].

- **Naive Bayes**

Naïve Bayes, a supervised learning method, utilizes the Bayes theorem to predict the probability of an event occurring based on previous observations of related events, as depicted in Figure 3 [30]. The NB classifier is one of the simplest machine learning classification algorithms. It can be used to create a fast classifier that makes rapid predictions from a dataset [29]. However, the NB classifier does not take into account the relationship between features for classification, which affects its accuracy [30].

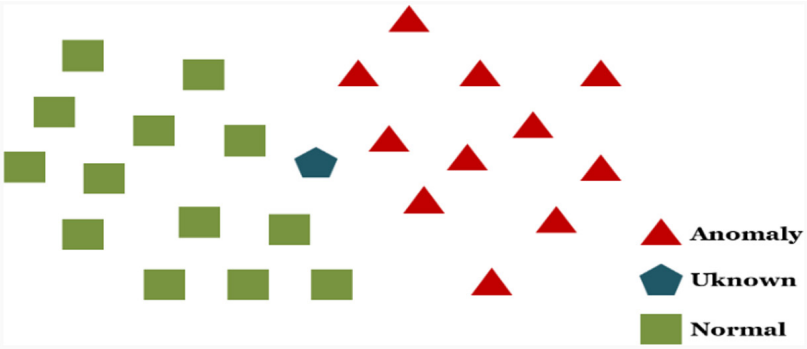

Fig. 2. K-nearest neighbor [30]

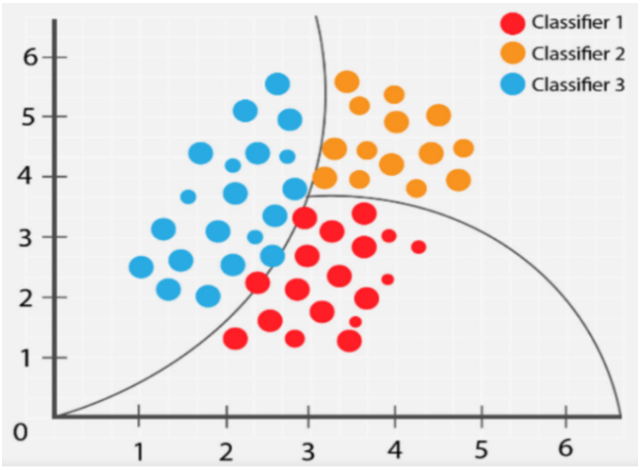

Fig. 3. Naive Bayes

2.3 Related works

Several previous studies have utilized machine learning classification methods to create an effective IDS using various datasets for different types of attacks.

The majority of these works were used for wireless sensor networks (WSNs), such as "efficient denial of service attack detection in WSNs" [7], "A WSN intrusion detection mechanism for smart environments (SLGBM)" [31], "assessment of machine learning techniques for WSNs denial-of-service detection" [32], "Using CNN, a deep learning method for efficient intrusion detection in wireless networks" [33], "A comparison of machine learning models for WSNs cyberattack detection" [34], "anomaly detection using machine learning techniques in wireless sensor networks" [35], "A lightweight multilayer machine learning detection system for WSN cyberattacks" [36], and "performance assessment of nave Bayesian procedures for WSN cyberattack detection" [37]. These are examples of machine learning techniques that have been evaluated for their usefulness in DoS detection in WSNs. The intrusion detection system proposed by [11] is called the hybrid deep neural network for network intrusion detection (CNN LSTM). It is commonly used for intrusion detection systems with various types of datasets, including WSN-DS, UNSW-NB15, and CIC-IDS2017. The project aims to develop a hybrid intrusion detection system model by combining the spatial feature extraction capabilities of convolutional neural networks with the temporal feature extraction capabilities of long- and short-term memory networks [11].

In reference to [38], it only uses machine learning for web attacks, in particular. Using R's statistical computing language, they proposed building many predictive models and evaluating the CIC-IDS2017 dataset [38]. The research aims to preprocess, evaluate, and develop a prediction model using the R language with the CIC-IDS2017 dataset to determine whether network connections are malicious. Their research includes the following machine learning classifiers: RF and artificial neural networks (ANN).

The accuracy results using ANN were as follows: brute force = 99.867%, XXS = 100%, and SQL injection = 90.476%. The accuracy results for RF were as follows: brute force = 98.009%, XXS = 99.540%, and SQL injection = 95.238%.

The summary of the evaluation of the methods presented is shown in Table 1.

3 RESEARCH METHODOLOGY

This section provides a comprehensive explanation of the methods employed for the study, encompassing dataset collection, data preprocessing, model selection, model training and testing, and model evaluation. The proposed model for web intrusion detection systems (Web IDS) is illustrated in Figure 4.

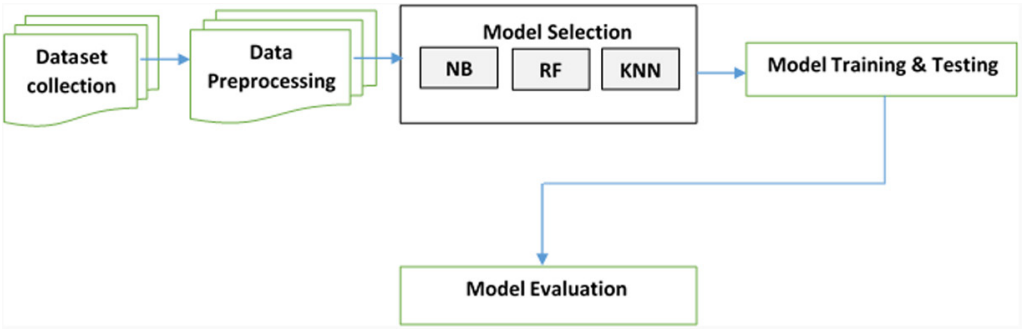

Fig. 4. Proposed model for IDS web attacks

Table 1. The summary of the evaluation

Reference	Objective	Dataset	ML Method	Results
Ref. [7]	This work examines 8 ML models using a feature selection algorithm to reduce the complexity.	WSN-DS	ANN, NB, BayesianNet, DecisionT, J48, RF, SMO, KNN	ANN 0.983 NB 0.954 BayesianNet 0.966 DecisionT 0.99 J48 0.997 Random Forest 0.997 SMO 0.971 KNN 0.994
Ref. [31]	To propose SLGBM as an intrusion detection method for WSN.	WSN-DS	ANN, DNN, J48, SMO, CNN, SLGBM	Normal – ANN, DNN, J48, SMO, CNN, and SLGBM: 0.998, 0.98, 0.999, 0.994, 0.972, and 0.998, respectively. Grayhole – ANN, DNN, J48, SMO, CNN, and SLGBM: 0.756, 0.919, 0.982, 0.501, 0.914, and 0.999, respectively. Blackhole – ANN, DNN, J48, SMO, CNN, and SLGBM: 0.928, 0.939, 0.993, 0.955, 0.938, and 1, respectively. TMDA – ANN, DNN, J48, SMO, CNN, and SLGBM: 0.922, 0.992, 0.927, 0.862, 0.987, and 0.999, respectively. Flooding – ANN, DNN, J48, SMO, CNN, and SLGBM: 0.994, 0.994, 0.975, 0.941, 0.956, and 0.999, respectively.
Ref. [32]	This research goals to assess the benefit of machine learning classification approaches to identify DoS attacks in WSNs that include (i) flooding, (ii) gray hole, and (iii) black hole.	WSN-DS	RF, REP Tree, J48, NB and Random Tree.	J48, NB, REP Tree, RF, and Random Tree have respective accuracy rates of 98.11%, 98.11%, 98.06%, 97.50%, and 98.07% for the gray hole. 97.88%, 97.89%, 97.71%, 97.47%, and 97.72% for NB, REP Tree, RF, Black Hole-J48 and Random Tree, respectively. Flooding – Random Forest and Random Tree: 99.13% and 99.11%, respectively. In J48, REP Tree, the accuracy was lower.
Ref. [38]	Using the CIC-IDS2017 dataset, preprocess, analyze, and develop a prediction model in R that can determine whether or not network connections are harmful	CIC -IDS2017	ANN, RF	Using ANN Brute force = 99.867%. XXS = 100%. SQL Injection = 90.476%. Using RF Brute force = 98.009%. XXS = 99.540%. SQL Injection = 95.238%.
Ref. [33]	Choose the most contributed features from the current convolutional neural network and classify them.	KDD 99 Cup	CNN with CRF-LCFS	Overall detection accuracy is 98.88%.
Ref. [34]	This paper compares machine learning classification techniques to detect cyberattacks in Wireless Sensor Networks and analyzes their performance.	WSN-DS	NB, KNN, GBM, RF, LightGBM, and Catboost	For the whole dataset, GBM, RF, LightGBM, and Catboost achieve 98.9%, 98.5%, 99.3%, and 99%, respectively; NB and KNN achieve the lowest result.
Ref. [35]	To propose an intrusion detection model (ID-GOPA) compatible with the characteristics of WSN.	WSN-DS	ID-GOPA, SVM, NB, DT, and RF.	ID-GOPA = 96% SVM = 89% NB = 94% DT = 94% RF = 94%

(Continued)

Table 1. The summary of the evaluation *(Continued)*

Reference	Objective	Dataset	ML Method	Results
Ref. [36]	To mitigate cyber-attacks that target Wireless Sensor Networks. They intend to use a mobile robot's assistance to combat internal WSN attacks.	WSN-DS	First-layer identification utilizing the NB calculation is utilized for parallel characterization. Second-layer identification utilizing the Light GBM calculation is utilized for multi class arrangement	Accuracy = 99.3%.
Ref. [37]	To compare three well-known base algorithms, SVM, KNN, and Multilayer Perceptron, with three variants of the Gaussian, Bernoulli, and multinomial Naive Bayes are examples of naive Bayes machine learning classification techniques.	WSN-DS	SVM, KNN, MLP, Multinomial NB, Bernoulli NB, and Gaussian NB	Gaussian NB = 72.43% Multinomial NB = 72.39% Bernoulli NB = 98.33% KNN = 97.04 SVM = 82.38% MLP = 70.6%
Ref. [11]	To consolidate the latitudinal element mining capacities of convolutional brain networks with the fleeting component extraction capacities of long momentary memory organizations to make a crossover interruption discovery framework model.	UNSW-NB15, WSN-DS, and CIC-IDS 2017	Deep learning algorithms CNN and LSTM	Accuracy for 5 epochs for 657 out of 658 binary classifications using the CIC IDS2017 was 99.64%, 94.53%, and 99.67%. The greatest binary and 62 multiclass identification rates for K = 8 with 5 epochs of UNSW-NB15 were, respectively, 94.53 and 82.41%. The best accuracy, detection rate, and F1-score were obtained with five binary WSN-DS epochs with K = 10, which were 99.67%, 98.14%, and 98%, respectively.
Ref. [6]	Looking at the viability of five AI strategies for identifying flooding, dark opening, blackhole, and planning DoS assaults in remote sensor networks involving the Waikato Climate for Information Examination (WEKA).	WSN-DS	NB, NN, SVM, J48, and RF	NB = 95.35% NN = 98.57% SVM = 97.11% J48 = 99.66% RF = 99.72%

- **Dataset collection**

Using the platform of the Canadian Institute for Cybersecurity (CIC-IDS2017), samples of malicious and benign web attacks were collected. The Canadian Institute for Cybersecurity has compiled the most recent harmless and widespread cyber-attacks to include in its CIC-IDS2017 dataset, which is used for cybersecurity research. It includes eighty features representing the most typical attack methods used today, obtained from simulated network traffic. These features include brute force FTP, web attack, infiltration, botnet, brute force SSH, DoS, Heartbleed, and DDoS [12], [9]. This study analyzed web attacks using the IDS dataset from the Canadian Institute for Cybersecurity (CIC-IDS2017). The 2016 McAfee study [12] shows that the dataset represents recent attacks. Machine learning methods will be used to further analyze web attacks using this dataset.

- **Data preprocessing**

Preparing the features for machine learning is referred to as data preprocessing [39, 40]. Data cleaning, feature selection, and data balancing are examples of data preparation procedures. This involves removing unnecessary or irrelevant

data, addressing missing values, and standardizing or otherwise modifying the data. To prepare the data for this study, the unprocessed CIC IDS2017 dataset will be obtained from the Canadian Institute for Cybersecurity website. Feature selection, data cleansing, and data transformation are essential tasks that need to be performed.

- **Selection model**

After preprocessing the data and selecting features, specific machine-learning algorithms are trained, evaluated, and implemented. At this stage, we are independently implementing machine learning techniques on the dataset using Python in the Jupyter environment. The CIC-IDS2017 datasets are classified using classifiers that incorporate KNN, RF, and NB, as well as 10-fold cross-validation.

- **Model for training and testing**

The training and testing procedures of machine learning are crucial for categorizing the CIC-IDS2017 datasets and are essential aspects that influence the effectiveness of machine learning. The quality of the proposed model is enhanced through an efficient training methodology. The quantity of instruction and testing is the most important aspect of the success rate. The datasets used in this study were divided into two halves for training and testing, using 10-fold validation. The 10-fold cross-validation method is commonly used to assess the error rate of a learning scheme on a specific dataset [40]. The dataset is divided into ten equal parts (folds) for 10-fold cross-validation, which assesses each part individually before averaging the results. Each data point in the dataset is utilized nine times for training and once for testing [40].

- **Evaluation model**

The accuracy, recall, precision, and F1-score of each machine learning classifier's performance were measured. We utilized Python and the Jupyter environment to execute these classifiers on the dataset. Section 4 provides a comprehensive explanation of the terms being assessed.

4 EXPERIMENTAL RESULTS AND DISCUSSION

The entire dataset has been used in experiments. This experiment utilizes Python software and the 10-fold cross-validation procedure. In this experiment, machine learning techniques are implemented using Python software. Python offers an extensive set of tools for these domains and is a popular programming language in data research, scientific computing, and machine learning [41]. Python was used to generate a 10-fold cross-validation dataset for the experiment. The industry standard for calculating the error rate of a knowledge scheme on a specific dataset is the 10-fold cross-validation approach. For reliable results, ten-fold cross-validation was used [40, 42]. Moreover, the dataset is divided into ten portions (folds) for 10-fold cross-validation. As a result, each portion of the dataset is used once for testing and nine times for training. Measures such as recall, accuracy, F1-score, and precision have been used to evaluate the performance of the classifiers in machine learning techniques. The trial results are presented in Table 2 for this analysis.

Table. 2. Results of three machine learning approaches for precision, recall, F1-score, and accuracy

Algorithm	Training Phase				Testing Phase			
	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score	Accuracy
RF	1	1	1	99.7883	1	1	1	99.4370
NB	0.99	0.772	0.861	77.0600	0.99	0.772	0.861	77.0794
KNN	1	1	0.99	99.6478	1	1	0.99	99.4916

4.1 Training phase

In this section, we will discuss the training phase results of F1-score, precision, recall, and accuracy using the machine learning techniques employed in our study.

- **Precision results in the training phase**

Precision measures the proportion of true positive classifications among all positive findings. This refers to the percentage of samples that have been accurately identified and are not false positives. According to Equation (1) accuracy is determined by the true positive (*TP*) and false positive (*FP*) rates. A higher *TP* value indicates greater accuracy.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Where "*TP*" refers to the number of samples that were correctly classified as positive. *FP* is the percentage of samples that are incorrectly identified as positive when they are actually negative.

The precision results for the various classifiers used in this experiment during the training phase are displayed in Table 1. In contrast to the accuracy value of 0.99, Figure 5 shows that the precision value of 1 for both RF and KNN is sufficient. This demonstrates how the NB classifier was outperformed by KNN and random forest.

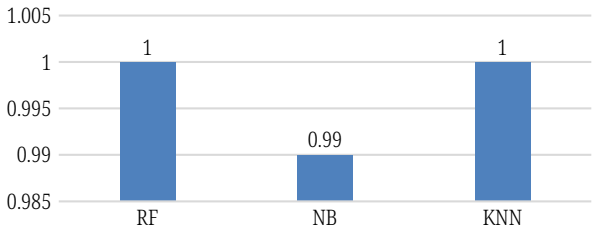

Fig. 5. Precision rate of various classification algorithms in the training phase

- **Recall results in the training phase**

Recall is the ratio of *TP* observations to all observations in the actual class that were truly anticipated to be positive. Other terms for it include sensitivity, hit rate, and true positive rate (TPR), denoted by Equation (2).

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

According to the results presented in Table 1, Figure 6 clearly shows that the RF and KNN have the highest recall values in the training phase, both of which are 1. While the recall values of the NB classifier are equal, their precision value is 0.772%.

Where is the false negative, or FN? This is a measure of the percentage of samples that were incorrectly identified as positive when they were actually negative.

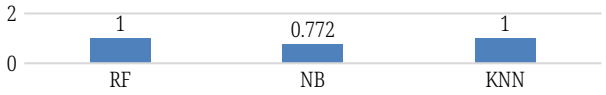

Fig. 6. Recall rate of various classification algorithms in training phase

- **F1-score results in the training phase**

To evaluate the overall effectiveness of the system, a single metric called the F1-score combines recall and precision. Equation (3) illustrates the calculation of the F1-score.

$$F1-score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

According to the experimental results in Table 1 and Figure 7, the RF model achieved the highest F1-score of 1 during the training phase. NB and KNN, with F1-score rates of 0.99 and 0.861, respectively, produced effective results. As can be seen, in terms of F1-score, the RF classifier outperforms the KNN and NB classifiers.

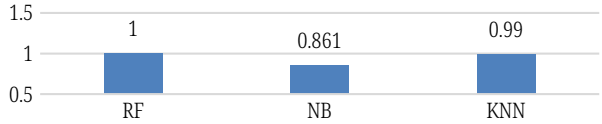

Fig. 7. F1-score rate of various classification algorithms in training phase

- **Accuracy results in the training phase**

Accuracy can also be defined as correctly categorizing. A performance measure known as "accuracy" can be used to describe the proportion of correct forecasts. The calculation of accuracy is shown in Equation (4)

$$Accuracy = \frac{TP + TN}{FP + FN + TP + TN} \tag{4}$$

where, *TN* occurs.

The accuracy rates for various classifiers during the training phase are presented in Table 1 and Figure 8. The RF model has achieved a remarkable accuracy rate of 99.7883% during the training phase. In terms of accurately classifying data, this algorithm performed better. With a classification accuracy of only 77.06%, NB exhibits the lowest accuracy. It achieved a 99.6478% accuracy rate for the KNN algorithm, demonstrating strong performance. The RF machine learning algorithm outperformed other machine learning algorithms in terms of accuracy in categorizing data during the training phase.

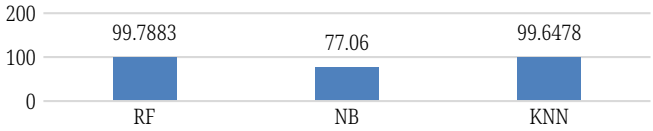

Fig. 8. Accuracy rate of various classification algorithms in the training phase

4.2 Testing phase

This section will examine the various common machine learning methods utilized in our study and analyze their relevance to the experimental outcomes for the different machine learning classifiers employed. During the training phase, the RF model achieved an outstanding accuracy rate of 99.7883%.

- **Precision results in the testing phase**

Figure 9 illustrates the precision rates of the various machine learning classifiers used in this study. In contrast to the accuracy value of 0.99 for NB, the precision values for KNN and RF are both 1. This indicates that the NB classifier performs less effectively than the KNN and RF classifiers.

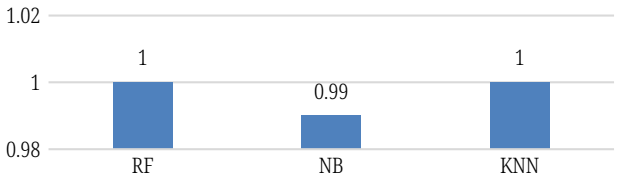

Fig. 9. Precision rate of various classification algorithms in the testing phase

- **K Recall results in the testing phase**

The highest recall values, which are 1 for RF and KNN, can be seen in Table 1 and Figure 10. While the recall values of the NB classifier are similar, the recall value for these individuals is 0.772%. It is evident that, in terms of recall rate, the NB classifier was outperformed by the RF and KNN classifiers.

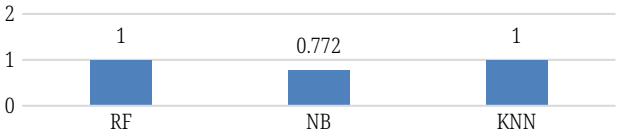

Fig. 10. Recall rate of various testing classification algorithms

- **F1-score results in the testing phase**

Experimental findings presented in Table 1 and Figure 11 demonstrate that RF achieved the highest F1-score of 1. NB and KNN also yielded efficient outcomes, with F1-score rates of 0.999 and 0.861, respectively. In terms of the F1-score, the RF classifier outperforms the NF and KNN classifiers.

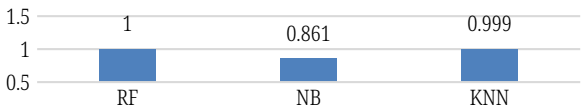

Fig. 11. F1-score rate of various classification algorithms in the testing phase

- **Accuracy results in the testing phase**

Correct categorization is another method for defining accuracy. The accuracy of forecasts can be measured using a performance metric. In the testing phase, Table 1 and Figure 12 display the accuracy rates of various classifiers. During the testing phase, the KNN algorithm achieved the highest accuracy rate (99.4916%), as indicated by the experimental results from the various machine learning classifiers

used in this study. This algorithm classified the data more accurately. NB has the lowest accuracy, with only 77.0794% of the proportion adequately identified. The RF algorithm achieved 99.437% classification accuracy, which is considered a good result. In terms of accuracy and classification rate during testing, the KNN method outperformed the other algorithms.

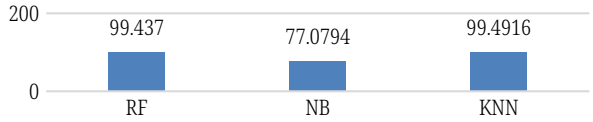

Fig. 12. Accuracy of different classification algorithms during testing phase

The results presented in Table 1 and Figures 5 to 12 indicate that the RF classifier outperformed the NB and KNN classifiers in terms of accuracy during the evaluation of the training phase. During the testing and assessment phases, the KNN classifier outperformed the RF and NB classifiers in terms of accuracy rate.

5 CONCLUSION

This study compared three machine learning classifiers for identifying web attacks in the instruction detection system (IDS). These classifiers include the NB, RF, and KNN classifiers. Python and the Jupyter environment were utilized to execute these classifiers on the dataset and assess the accuracy rate. The datasets for web attacks have been initially gathered. Data cleaning, primary feature selection, and extraction were performed during the data preprocessing phase after obtaining the raw CIC-IDS2017 dataset. After that, the CIC-IDS2017 datasets are classified using various machine learning classifiers, such as KNN, RF, and NB classifiers. These machine-learning classifiers undergo training, testing, and evaluation. Finally, the Python software compares the detection accuracy rate, F1-score rate, recall, and precision of several machine-learning classifiers using the CIC-IDS2017 datasets. The experiments utilized a 10-fold cross-validation procedure. RF classifiers were found to be the most accurate for detecting web attacks in IDS during the training phase, according to the studies conducted. During the testing phase, the KNN classifier outperformed the NB and RF classifiers in terms of accuracy when classifying IDS web attacks. The results of this study will be beneficial to other researchers as they strive to develop an effective method for a web-attack intrusion detection system.

6 REFERENCES

[1] Y. Pan *et al.*, "Detecting web attacks with end-to-end deep learning," *J. Internet Serv. Appl.*, vol. 10, no. 1, pp. 1–22, 2019. https://doi.org/10.1186/s13174-019-0115-x

[2] D. Kshirsagar and S. Kumar, "An ensemble feature reduction method for web-attack detection," *J. Discret. Math. Sci. Cryptogr.*, vol. 23, no. 1, pp. 283–291, 2020. https://doi.org/10.1080/09720529.2020.1721861

[3] R. Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems," *Int. J. Eng. Technol.*, vol. 7, no. 3.24, pp. 479–482, 2018.

[4] E. E. Abdallah, W. Eleisah, and A. F. Otoom, "Intrusion detection systems using supervised machine learning techniques: A survey," *Procedia Comput. Sci.*, vol. 201, pp. 205–212, 2022. https://doi.org/10.1016/j.procs.2022.03.029

[5] I. Almomani, B. Al-Kasasbeh, and M. Al-Akhras, "WSN-DS: A dataset for intrusion detection systems in wireless sensor networks," *J. Sensors*, vol. 2016, pp. 1–16, 2016. https://doi.org/10.1155/2016/4731953

[6] L. Alsulaiman and S. Al-Ahmadi, "Performance evaluation of machine learning techniques for DOS detection in wireless sensor network," *Int. J. Netw. Secur. Its Appl.*, vol. 13, no. 2, pp. 21–29, 2021. https://doi.org/10.5121/ijnsa.2021.13202

[7] G. A. N. Segura, S. Skaperas, A. Chorti, L. Mamatas, and C. B. Margi, "Efficient denial of service attacks detection in wireless sensor networks," *J. Inf. Sci. Eng.*, vol. 34, no. 4, pp. 977–1000, 2018. https://doi.org/10.1109/ICCWorkshops49005.2020.9145136

[8] P. Nancy, S. Muthurajkumar, S. Ganapathy, S. V. N. Santhosh Kumar, M. Selvi, and K. Arputharaj, "Intrusion detection using dynamic feature selection and fuzzy temporal decision tree classification for wireless sensor networks," *IET Commun.*, vol. 14, no. 5, pp. 888–895, 2020. https://doi.org/10.1049/iet-com.2019.0172

[9] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, no. c, pp. 41525–41550, 2019. https://doi.org/10.1109/ACCESS.2019.2895334

[10] L. Segurola-Gil, F. Zola, X. Echeberria-Barrio, and R. Orduna-Urrutia, "NBcoded: Network attack classifiers based on encoder and naive bayes model for resource limited devices," *Commun. Comput. Inf. Sci.*, pp. 55–70, 2021. https://doi.org/10.1007/978-3-030-93733-1_4

[11] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: Hybrid deep neural network for network intrusion detection system," *IEEE Access*, vol. 10, pp. 99837–99849, 2022. https://doi.org/10.1109/ACCESS.2022.3206425

[12] A. Yulianto, P. Sukarno, and N. A. Suwastika, "Improving adaboost-based Intrusion Detection System (IDS) performance on CIC IDS 2017 dataset," *J. Phys. Conf. Ser.*, vol. 1192, pp. 1–10, 2019. https://doi.org/10.1088/1742-6596/1192/1/012018

[13] S. Vaithyasubramanian and A. Christy, "An analysis of CFG password against brute force attack for web applications," *Contemp. Eng. Sci.*, vol. 8, no. 9, pp. 367–374, 2015. https://doi.org/10.12988/ces.2015.5252

[14] F. Ayankoya and B. Ohwo, "Brute-force attack prevention in cloud computing using one-time password and cryptographic hash function," *Int. J. Comput. Sci. Inf. Secur.*, vol. 17, no. 2, pp. 7–19, 2019. [Online]. Available: https://www.academia.edu/38523734/Brute-Force_Attack_Prevention_in_Cloud_Computing_Using_One-Time_Password_and_Cryptographic_Hash_Function.

[15] I. Tariq, M. A. Sindhu, R. A. Abbasi, A. S. Khattak, O. Maqbool, and G. F. Siddiqui, "Resolving cross-site scripting attacks through genetic algorithm and reinforcement learning," *Expert Syst. Appl.*, vol. 168, p. 114386, 2021. https://doi.org/10.1016/j.eswa.2020.114386

[16] I. Hydara, A. B. M. Sultan, H. Zulzalil, and N. Admodisastro, "Current state of research on cross-site scripting (XSS) – A systematic literature review," *Inf. Softw. Technol.*, vol. 58, pp. 170–186, 2015. https://doi.org/10.1016/j.infsof.2014.07.010

[17] S. Gupta and B. B. Gupta, "Cross-Site Scripting (XSS) attacks and defense mechanisms: Classification and state-of-the-art," *Int. J. Syst. Assur. Eng. Manag.*, vol. 8, pp. 512–530, 2017. https://doi.org/10.1007/s13198-015-0376-0

[18] M. Alghawazi, D. Alghazzawi, and S. Alarifi, "Detection of SQL injection attack using machine learning techniques: A systematic literature review," *J. Cybersecurity Priv.*, vol. 2, no. 4, pp. 764–777, 2022. https://doi.org/10.3390/jcp2040039

[19] S. W. Boyd and A. D. Keromytis, "SQLrand: Preventing SQL injection attacks," *Lect. Notes Comput. Sci.*, vol. 3089, pp. 292–302, 2004. https://doi.org/10.1007/978-3-540-24852-1_21

[20] W. G. J. Halfond, J. Viegas, and A. Orso, "A classification of SQL injection attacks and countermeasures," in *Proceedings of the IEEE International Symposium on Secure Software Engineering*, 2006, pp. 1–11.

[21] M. Baklizi, I. Atoum, N. Abdullah, O. A. Al-Wesabi, A. A. Otoom, and M. A. S. Hasan, "A technical review of SQL injection tools and methods: A case study of SQL map," *Int. J. Intell. Syst. Appl. Eng.*, vol. 10, no. 3, pp. 75–85, 2022.

[22] M. Baklizi, I. Atoum, M. A. S. Hasan, N. Abdullah, O. A. Al-Wesabi, and A. A. Otoom, "Prevention of website SQL injection using a new query comparison and encryption algorithm," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 1, pp. 228–238, 2023.

[23] H. X. Dau, N. T. T. Trang, and N. T. Hung, "A survey of tools and techniques for web attack detection," *J. Sci. Technol. Inf. Secur.*, vol. 1, no. 15, pp. 109–118, 2022. https://doi.org/10.54654/isj.v1i15.852

[24] H. M. Kanaker, M. M. Saudi, and M. F. Marhusin, "A systematic analysis on worm detection in cloud-based systems," *ARPN J. Eng. Appl. Sci.*, vol. 10, no. 3, pp. 1405–1412, 2015.

[25] H. M. Kanaker, M. M. Saudi, and M. F. Marhusin, "Detecting worm attacks in cloud computing environment: Proof of concept," in *Proceedings – 2014 5th IEEE Control and System Graduate Research Colloquium, ICSGRC*, 2014, pp. 253–256. https://doi.org/10.1109/ICSGRC.2014.6908732

[26] A. Chawla, "Phishing website analysis and detection using machine learning," *Int. J. Intell. Syst. Appl. Eng.*, vol. 10, no. 1, pp. 10–16, 2022. https://doi.org/10.18201/ijisae.2022.262

[27] S. Alnemari and M. Alshammari, "Detecting phishing domains using machine learning," *Appl. Sci.*, vol. 13, no. 8, p. 4649, 2023. https://doi.org/10.3390/app13084649

[28] A. Awasthi and N. Goel, "Phishing website prediction using base and ensemble classifier techniques with cross-validation," *Cybersecurity*, vol. 5, no. 1, 2022. https://doi.org/10.1186/s42400-022-00126-9

[29] A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, S. Ramana, and K. Joga, "Phishing detection system through hybrid machine learning based on URL," *IEEE Access*, vol. 11, pp. 36805–36822, 2023. https://doi.org/10.1109/ACCESS.2023.3252366.

[30] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, and A. Wahab, "A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions," *Electron.*, vol. 9, no. 7, pp. 1–45, 2020. https://doi.org/10.3390/electronics9071177

[31] S. Jiang, J. Zhao, and X. Xu, "SLGBM: An intrusion detection mechanism for wireless sensor networks in smart environments," *IEEE Access*, vol. 8, pp. 169548–169558, 2020. https://doi.org/10.1109/ACCESS.2020.3024219

[32] S. E. Quincozes and J. F. Kazienko, "Machine learning methods assessment for denial of service detection in wireless sensor networks," in *IEEE World Forum Internet Things, WF-IoT 2020 – Symp. Proc.*, 2020, pp. 1–6. https://doi.org/10.1109/WF-IoT48130.2020.9221146

[33] B. Riyaz and S. Ganapathy, "A deep learning approach for effective intrusion detection in wireless networks using CNN," *Soft Comput.*, vol. 24, no. 22, pp. 17265–17278, 2020. https://doi.org/10.1007/s00500-020-05017-0

[34] S. Ismail, T. T. Khoei, R. Marsh, and N. Kaabouch, "A comparative study of machine learning models for cyber-attacks detection in wireless sensor networks," in *IEEE 12th Annu. Ubiquitous Comput. Electron. Mob. Commun. Conf. UEMCON 2021*, 2021, pp. 313–318. https://doi.org/10.1109/UEMCON53757.2021.9666581

[35] S. Ifzarne, H. Tabbaa, I. Hafidi, and N. Lamghari, "Anomaly detection using machine learning techniques in wireless sensor networks," *J. Phys. Conf. Ser.*, vol. 1743, no. 1, 2021. https://doi.org/10.1088/1742-6596/1743/1/012021

[36] S. Ismail, D. Dawoud, and H. Reza, "A lightweight multilayer machine learning detection system for cyber-attacks in WSN," in *IEEE 12th Annu. Comput. Commun. Work. Conf. CCWC*, 2022, pp. 481–486. https://doi.org/10.1109/CCWC54503.2022.9720891

[37] S. Ismail and H. Reza, "Evaluation of naive bayesian algorithms for cyber-attacks detection in wireless sensor networks," in *IEEE World AI IoT Congr. AIIoT 2022*, 2022, pp. 283–289. https://doi.org/10.1109/AIIoT54504.2022.9817298

[38] Z. Pelletier and M. Abualkibash, "Evaluating the CIC IDS-2017 dataset using machine learning methods and creating multiple predictive models in the statistical computing language R," *Int. Res. J. Adv. Eng. Sci.*, vol. 5, no. 2, pp. 187–191, 2020.

[39] S. B. Kotsiantis and D. Kanellopoulos, "Data preprocessing for supervised leaning," vol. 1, no. 2, pp. 111–117, 2006. https://users.ece.utexas.edu/~ethomaz/courses/dm/papers/data-preprocessing.pdf

[40] H. Kanaker, N. A. Karim, S. A. B. Awwad, N. H. A. Ismail, J. Zraqou, and A. M. F. Alali, "Trojan horse infection detection in cloud based environment using machine learning," *Int. J. Interact. Mob. Technol.*, vol. 16, no. 24, pp. 81–106, 2022. https://doi.org/10.3991/ijim.v16i24.35763

[41] S. Raschka and V. Mirjalili, *Python Machine Learning: Machine Learning and Deep Learning with Python. Scikit-Learn, and TensorFlow*. Second edition. 2017.

[42] R. Bouckaert *et al.*, "WEKA manual for version 3-9-1," 2016. [Online]. Available: http://usermanual.wiki/Document/WekaManual391.1255144600.

7 AUTHORS

Mahmoud Khalid Baklizi is an Associate Professor in Department of Computer Sciences, Faculty of Information Technology, Isra University, Jordan (E-mail: mbaklizi@iu.edu.jo).

Issa Atoum is an Associate Professor with the Software Engineering Department, Faculty of Information Technology, The World Islamic Sciences and Education University, Jordan (E-mail: issa.atoum@wise.edu.jo).

Mohammad Alkhazaleh is an Assistant Professor in Department of Computer Sciences, Faculty of Information Technology, Isra University, Jordan (E-mail: m.alkhazaleh@iu.edu.jo).

Hasan Kanaker is an Assistant Professor in Cybersecurity department and the Head of Cybersecurity and Computer Information System departments, Faculty of Information Technology at Isra University, Jordan (E-mail: hasan.kanaker@iu.edu.jo).

Nibras Abdullah is an Assistant Professor in School of Computer Sciences, Universiti Sains Malaysia, Malaysia (E-mail: nibras@usm.my).

Ola A. Al-Wesabi is an Assistant Professor in Faculty of Computer Science and Engineering, Hodeidah University, Hodeidah, Yemen (E-mail: ola.wosabi@gmail.com).

Ahmed Ali Otoom is an Assistant Professor in Cybersecurity and Cloud Computing Department, Faculty of Information Technology, Applied Science Private University, Jordan (E-mail: a_otoom@asu.edu.jo).