مجلة جامعة دمشق للعلوم الأساسية
مجلد 40- عدد 3 (2024): 109-127

Damascus University Journal for the Basic Sciences
Vol 40 No.3 (2024): 109 -127

تطوير نظام ذكي لكشف الاختراقات الشاذة في الشبكات الحاسوبية استنادا إلى الشبكة العصونبية ذات الانتشار العكسي المرن

جورج أنور كراز [1]

[1]قسم الذكاء الصنعي ومعالجة اللغات الطبيعية، كلية الهندسة المعلوماتية، جامعة دمشق، سوريا.

george.karraz@damascusuniversity.edu.sy

تاريخ الإيداع: 2022/08/08
تاريخ الموافقة: 2022/11/06

الملخص

تعتبر الهجمات الشاذة المختلفة وتعطيل شبكات المعلومات من المشاكل الخطيرة التي تؤثر على حماية المعلومات المتبادلة بين هذه الشبكات وتؤثر على الحفاظ على موثوقية وسرية تبادل المعلومات. في العقد الماضي، واجه الباحثون في جميع أنحاء العالم العديد من التحديات ويحتاجون إلى اقتراح مجموعة من الأنظمة ذات بنيات مرنة للكشف الدقيق والتلقائي عن هجمات التسلل الشاذة لمعالجة تعقيداتها. اقترحت الأبحاث ذات الصلة العديد من الحلول واسعة النطاق القائمة على تقنيات التعلم الآلي (ML). ركزت الأبحاث الحديثة على بناء نظام كشف التسلل الشاذ للإيدز من وجهة نظر رياضية ومعمارية، باستخدام أساليب متطورة مثل أجهزة ناقل الدعم(SVMs) والشبكات العصبية التلافيفية (CNNs). تستخدم العديد من الدراسات الإيدز المعتدل والمنخفض التعقيد استنادا إلى الشبكة العصبية الكلاسيكية متعددة الطبقات MLNN. ولذلك، فإن دقة مصنفات MLNN في مرحلة الاختبار معتدلة أو منخفضة. استنادا إلى دراسات الإيدز ذات الصلة المقترحة في الأدبيات وتحقيقاتنا التفصيلية، نجد أن خوارزمية RBP المرنة للانتشار الخلفي لا تستخدم كوسيلة تعليمية للإيدز المستند إلى MLNN. على وجه الخصوص، تعتبر RBP أداة فعالة في العديد من المصنفات الثنائية غير الخطية. في هذا البحث، نقدم طريقة بناء الإيدز بناء على MLNN المدربة بواسطة خوارزميةRBP ، باستخدام البيانات ذات الصلة المعروفة NSL–KDD و CIC–DDoS2019.في هذه الدراسة، اخترنا بعناية بنية مناسبة للإيدز وقمنا بالعديد من المحاولات لتجنب الصعوبات المذكورة أعلاه. وقد تبين أن أدواتنا الخاصة بالإيدز قد تم تدريبها بشكل ثابت دون قيود خلال فترة زمنية معقولة، ثم تم اختبارها لاحقا على بيانات غير مسبوقة، بدقة تصل إلى حوالي 99%. قمنا أيضا بمقارنة أداء خوارزميتنا مع خوارزميات تعلم MLNN المعروفة الأخرىLevenberg Marquardt LM،

ISSN (online): 2789-6366
http://journal.damascusuniversity.edu.sy/index.php/index/index

وBayesian Regulated BR وQuasi-Newton QN باستخدام نفس بنية الإيدز ومجموعة البيانات. تظهر نتائج المقارنة أن خوارزمية RBP تتمتع بأفضل أداء بين العديد من الخوارزميات.

الكلمات المفتاحية: شبكات الحاسوب، هجمات الشبكة، نظام كشف التسلل الشاذ AIDS، التعلم الآلي ML،الشبكات العصبونية المتعددة، الانتشار العكسي المرن RBP، NSL-KDD،CIC-DDoS2019

Develop an Intelligent Anomaly Intrusion Detection System in Computer Networks based on Resilient Back-propagation Neural Network

George Anwar Karraz[1]

[1]Department of Artificial Intelligence and Natural Language Processing, Faculty of Information Technology Engineering,
Damascus University, Syria george.karraz@damascusuniversity.edu.sy

Abstract

Various anomaly attacks and disruptions to information networks are considered serious problems that affect the protection of information exchanged between these networks and affect the maintenance of reliability and confidentiality of information exchange. In the past decade, researchers around the world have faced many challenges and need to propose a set of systems with flexible architectures to accurately and automatically detect anomaly intrusion attacks to address their complexity. Related research has proposed many full-scale solutions based on machine learning ML techniques. Recent research has focused on building an anomaly intrusion detection system AIDS from a mathematical and architectural point of view, using sophisticated methods such as support vector machines (SVMs) and convolutional neural networks (CNNs). Many studies use moderate and low complexity AIDS based on the classical multilayer neural network MLNN. Therefore, the accuracy of MLNN classifiers in the testing phase is moderate or low. Based on relevant AIDS studies proposed in the literature and our detailed investigation, we find that the resilient backpropagation RBP algorithm is not used as a learning method for MLNN-based AIDS. In particular, RBP is an effective tool in many nonlinear binary classifiers. In this paper, we present an AIDS construction method based on MLNN trained by the RBP algorithm, using well-known related data NSL-KDD and CIC-DDoS2019. In this study, we carefully selected an appropriate AIDS architecture and made many attempts to avoid the above difficulties. Our AIDS was found to be stably trained without limitations in a reasonable amount of time, and subsequently tested on unprecedented data, with an accuracy of about 99%. We also compared the performance of our algorithm with other well-known MLNN learning algorithms (Levenberg Marquardt LM, Bayesian Regulated BR, and Quasi-Newton QN) using the same AIDS architecture and data set. The comparison results show that the RBP algorithm has the best performance among many algorithms.

Received : 08/08/2022
Accepted:06/11/2022

Keywords: computer networks, network attacks, anomaly intrusion detection system AIDS, machine learning ML, multilayer neural network MLNN, resilient backpropagation RBP, NSL-KDD, CIC-DDoS 2019.

1. Introduction

An intrusion is defined as any unauthorized act that causes damage to an information system. Or any attack that may threaten the confidentiality, integrity, or availability of information [11]. AIDS monitors network traffic for threats that attempt to penetrate a network or system. The main purpose of developing an AIDS is to detect threats that cannot be detected by traditional methods or firewalls. This is a key requirement to achieve protection against zero-day attacks (new attacks). When a threat is detected, the AIDS sends an alert to the network administrator, who can then take action. Therefore, organizations need to improve their products when installing AIDS. There is a main difference between AIDS and another similar type of cyber security system, the intrusion prevention system (IPS), since both of them monitor network packets entering the system, check for relevant malicious activity, and send alerts immediately, but IPS can also take preventive action against any malware activity detected. In recent years, artificial neural networks have been successfully applied to the development of AIDS because of their advantage of easily dealing with nonlinear binary problems between AIDS inputs and output [2]. Even if the data is incomplete or skewed, neural networks can analyze network data [12]. The idea of applying soft computing techniques such as MLNN is to build AIDS can reveal both the anomaly and normal hidden patterns among traffic records [3]. The working environment of AIDS can be categorized as follows [23]:

1. Network AIDS (NAIDS): Executes efficient analysis by matching all outgoing traffic on a subnet against a set of known attacks (abnormal intrusions), and in the case of a perfect match, it detects abnormal behavior or abnormal intrusions and generates then sends alerts to the network administrator. NAIDS should be installed on subnets where firewalls need to check to see if someone is trying to break into the firewall.
2. Host AIDS (HAIDS) Runs on individual hosts, HAIDS monitors incoming and outgoing packets that are first left behind and installed only by the host; HAIDS takes snapshots of the host's existing system files in sequence and compares each current snapshot with the previous one. If a snapshot of a particular file has been changed or modified, HAIDS sends an alarm to the network administrator; HAIDS can be seen in a working environment where there is no expectation of frequent change in configuration [9].

AIDS is employed to detect unknown malware attacks when new malware is present. ML-based approaches are applied to create reliable AIDS working models [5]. ML-based methods are more versatile because they can be trained according to application and hardware configurations. The process of building AIDS operates in two phases: a learning phase and a testing phase. Traffic profiles are used in the training phase to train the model on normal and abnormal behavior, and in the testing phase, a new data set is used to define the generalizability of AIDS against unknown intrusions.

In this study, we propose an AIDS based on MLNN architecture and RBP learning algorithm. Our AIDS is a type of ML-based method. The main reason to select RBP is that there are absolutely no previous studies employing the RBP algorithm as an efficient tool in the AIDS case study, and the RBP algorithm is simple from a mathematical point of view; the RBP algorithm requires less memory usage in the training phase, takes a reasonable amount of time to learn. In addition, RBP was proposed in the literature to achieve improvements over the classical backpropagation algorithm [18]. We will discuss these improvements in Section 5 of this paper. Our proposed AIDS achieved significant performance improvements compared to peer studies proposed in related work.

2. Problem Description

Over the past few years, we have examined much of the development of AIDS through several architectures and designs. On the other hand, we have conducted several experiments to conclude the best ML approach that can be applied to build an efficient AIDS. Therefore, through this study, we aimed to overcome the gaps included in previous studies, such as:

1. Large errors due to H-P approximation of nonlinear problems in the training phase of SVM-based and CNN-based AIDS.
2. The learning problems encountered in MLNN-based AIDS using other learning algorithms instead of RBP (these learning problems are discussed in the previous section of this paper

3. Related Studies

In this section, we review previous proposed related work to address the anomalous intrusion classification problem.

Venkatesan [24] employed the NSL-KDD dataset and developed three ML algorithms: decision forest classification, random forest, and SVM. This work also provides a feature selection mechanism for creating accurate AIDS models but has only been applied to limited data sets.

Yaqoob et al [25] introduced the dynamic convolutional automatic encoder anomaly detection CAaDet system based on deep learning for anomaly detection CAaDet uses custom automatic encoders and convolution layers to extract useful features and detect anomalies, NSL-KDD dataset was employed in this work. The proposed system was evaluated using the F1 score metric and achieved improved results in Fa-IoVs fog-assisted Internet of Vehicles.

Pawar et al [17] presented an optimized AdaBoost classifier fine-tuned using the Hybrid Firefly algorithm and Particle Swarm Optimization HFPSO. Data are preprocessed to fit a standard function by normalizing them. In addition, a cross-correlation technique was used to eliminate redundancy from the selected features. Finally, the constructed signals were used to train and test an AdaBoost classifier optimized for HFPSO. The results showed the ability to accurately predict attacks on a selected limited data set.

Fakiha [6] investigated four multiple ML techniques for detecting distributed denial-of-service DDoS attacks; the CIC-DDoS dataset was used to investigate the accuracy of DDoS attack detection. The Random Forest ML model achieved 99.997 % DDoS detection accuracy, which is higher than CNN, Cat Boost, and Light GB. However, this performance is only on a limited dataset.

Mehmood et al [14] proposed a hybrid AIDS-based method consisting of three steps. The first step preprocesses the dataset, the second step employs a random forest recursive feature elimination method, and the third step uses a different type of SVM, but only on limited data [4].

Meliboev et al [15] proposed an intelligent IDS system based on CNN. The NSL-KDD dataset was employed in this study. The system was built using an input layer with 41 neurons reduced by 32 convolutional kernels. Sequentially, two connected layers were used to classify the 32 features into two classes, normal and attack, with the SoftMax activation function used in the classification layer. Results for the testing phase were 85.5% accuracy, 77.1% recall, and 85.9% F-score. Performance was very moderate, indicating that it cannot approach the ideal AIDS. On the other hand, the authors attempted to demonstrate the effectiveness of this approach by referring to previous studies on the use of CNNs as computer vision tools, we are not convinced that this comparison is accurate. However, the approach we developed achieved better results than those reported in this study, and we trained and tested our AIDS on two different data sets.

Sokkalingam et al [20] proposed a hybrid AIDS that combines SVM and Harris Hawks optimization by (HHO). Its performance has been compared with other classical algorithms such as C4.5 and K-nearest neighbor using performance measures such as sensitivity, specificity, and accuracy. The proposed system detects DDoS with good performance. But there should be taken into account the complexity of this system.

Sumathi et al [21] used a Long Short-Term Memory (LSTM) recurrent neural network with an automatic encoder and decoder-based deep learning strategy using gradient descent learning rule. By employing the proposed hybrid Harris Hawks Optimization (HHO) and Particle Swarm Optimization (PSO) algorithms, network parameters such as weight vector and bias coefficient are optimally adjusted. The results confirm that the proposed LSTM and deep learning model achieves moderate performance compared to other models developed in related studies.

Gu et al [8] proposed an efficient AIDS based on SVM and Naive Bayes. In this study, four datasets, UNSW-NB15, NSL-KDD, Kyoto 2006, and CICIDS 2017, were employed. Experimental results showed that the CICIDS 2017 dataset achieved 98.92% accuracy, the NSL-KDD dataset achieved 99.35% accuracy, the UNSW - NB 15 dataset achieved 93.75% accuracy, and the Kyoto 2006 + dataset achieved 98.58% accuracy. However, the authors did not prove the stability of the system and the absence of problems that might occur during the training phase.

Ferrag et al [7] developed three deep learning approaches for anomaly-based IDS in agriculture using the CIC-DDoS 2019 and TON-IoT datasets with several types of DDoS attacks: a CNN, a deep neural network DNN, and MLNN. The study used several effective performance metrics during the testing phase. The authors created three different datasets from CIC-DDoS 2019 and named each dataset created according to the number of attack classes presented: dataset_13_class, dataset_7_class, and dataset_2_class. The results of this study are reported as a measurement of the performance indicators of each dataset created and compare the indicators reached between each approach developed, the obtained results don't give the reader clear information about the efficiency of the applied approaches. The efficiency appears weak in some results for certain types of attacks, while at the same time, is good for other categories of attacks. On the other hand, the chosen approach has a complexity compromising with the scope of this study, and compared to our work, using the same dataset (CIC-DDoS2019) gives better results, considering the simplicity and efficiency of our approach.

Mahdavisharif et al [13] proposed a big data-enabled AIDS in communication networks based on a Long Short-Term Memory (LSTM) deep learning approach. The NSL-KDD dataset was employed in this study, and this AIDS contains three layers: the first layer has 41 neurons depending on the number of NSL-KDD features, and all records of features go into the second layer (the hidden layer of the developed approach). Each LSTM cell has two outputs, one representing the current state of each cell (c) and the other representing the output of each cell (h). Furthermore, h is given as an input to the first cell of the same block, and all block outputs enter the neurons of the output layer to classify the current feature record. To evaluate the developed system, four indices were chosen: detection rate DR, false positive rate FAR, accuracy, and learning time of about TT. DR=0.988, FAR=0.01, Accuracy=0.969, and TT=33.65m respectively, the approach reached the best training performance at a learning rate of 0.01. There is an important gap in this study regarding the lack of representation of the relationship between DR and FAR at many thresholds of classification using the receiver operating curve ROC to ensure classification correctness. The reported results of this approach indicate that when DR = 0.988 and FAR = 0.01, logically the accuracy must be greater than 0.969 for correct classification. In our work, we have used ROC curves to prove the classification correctness of our approach.

Mishra [16] proposed an AIDS based on a feed-forward neural network using the BP algorithm. The KDD CUP dataset was employed in this study. The developed approach first performed the preprocessing phase of KDD CUP, selecting two main features that feed into two nodes of the input of the applied neural network, and two hidden layers with three nodes each (six hidden nodes in total) were used. The training model was run using mean squared error (MSE) analysis with a learning rate of 0.80. The best classification was obtained at epoch 54, with an MSE of 0.0038, and the network took 249.7030 seconds to reach the error goal. In the testing phase on unseen data, there were three categories of erroneous outputs: false positives, false negatives, and irrelevant neural network outputs that did not represent any of the output classes in the data set. This model is simpler in structure than the similar AIDS and reduces computational overhead and memory usage. However, there are two limitations to this approach: the first relates to the presence of irrelevant cases in the classification, and the second relates to an illogical mathematical method that reduces the 41 selected features to only 2. So, this AIDS is inadequate to detect all kinds of attacks presented in the KDD CUP dataset.

4. Motivation and Objectives

The main motivation behind this study is to present an initiative in the subject of designing a sophisticated AIDS adapted to the studied dataset, making several attempts to apply various neural network learning algorithms to achieve the best performance and stability of the suggested approach. Then, reach optimal performance during the testing phase concerning the success of applying the exact proposed AIDS to unseen test data.

5. Proposed Approach

Our approach is based on applying three main steps: The first step describes the structure of our selected dataset, the second step demonstrates our technique of data preprocessing, and the third step describes the architecture, training phase, and testing phase adopted in developing our AIDS.

5.1. Dataset Selection

In this paper, the NSL-KDD [22] and CIC-DDoS2019[19] datasets were used for the training and testing phases.

5.1.1. NSL-KDD Dataset

It is a modified and reduced version of the KDD-CUP-99 dataset [10] which contains a variety of simulated intrusions in a military network environment, created using network traffic captured from the 1998 DARPA AIDS evaluation program. The network traffic includes both normal traffic and anomaly traffic including Distributed Denial of Service DDoS, Probing, User-to-Root (U2R), and Root-to-Local (R2L) traffic. The training data set is compressed binary TCP dump data from seven weeks of network traffic. KDD-CUP contains about 5 million contact records; the two weeks of test data contain about 2 million contact records. The KDD CUP dataset has been widely used as a reference dataset for AIDS evaluation for several years. One of the main drawbacks of this dataset is that both the training and test data contain a huge amount of redundant records. It was found redundant ratios of 78.05% and 75.15% of the total number of records in the training data set and test data set respectively. This redundancy causes the training algorithm to be biased toward repeated attack records, resulting in poor classification results. NSL-KDD is a data set designed to address some of the problems inherent in the KDD'99 data set. Although this new version of the KDD dataset still has problems and may not fully represent existing real networks due to the lack of publicly available datasets for network-based AIDS, we believe it can serve as a strong reference dataset to assist researchers. Furthermore, the number of entries in the NSL-KDD training and testing materials is reasonable. Therefore, it would be easy to use all of the records in the NSL-KDD data set for the AIDS proposal, and it is not necessary to select a subset. Furthermore, the NSL-KDD is divided into two sets, a training dataset and a test dataset, consisting of 125973 and 22544 records, respectively. Each record is denoted by 41 attributes (features), 21 of which describe the connection itself, and the remaining 19 correlate to the host connection; the entire NSL-KDD traffic is classified into five classes, four of which are related to the types of attacks and a fifth class is related to the regular traffic. Table 1 below shows the attack and normal classes examined in the NSL-KDD training and test records. Figure 1 shows the distribution of traffic in the NSL-KDD training and test sets, respectively.

TABLE 1: Attacks and normal traffic in the NSL-KDD dataset

TRAFFIC TYPE		TRAINING RECORDS	TEST RECORDS
ATTACKS	DDoS	45915	7458
	U2R	51	67
	R2L	995	2887
	Probe	11664	2421
TOTAL ATTACKS		58625	12833
NORMAL		67348	9711

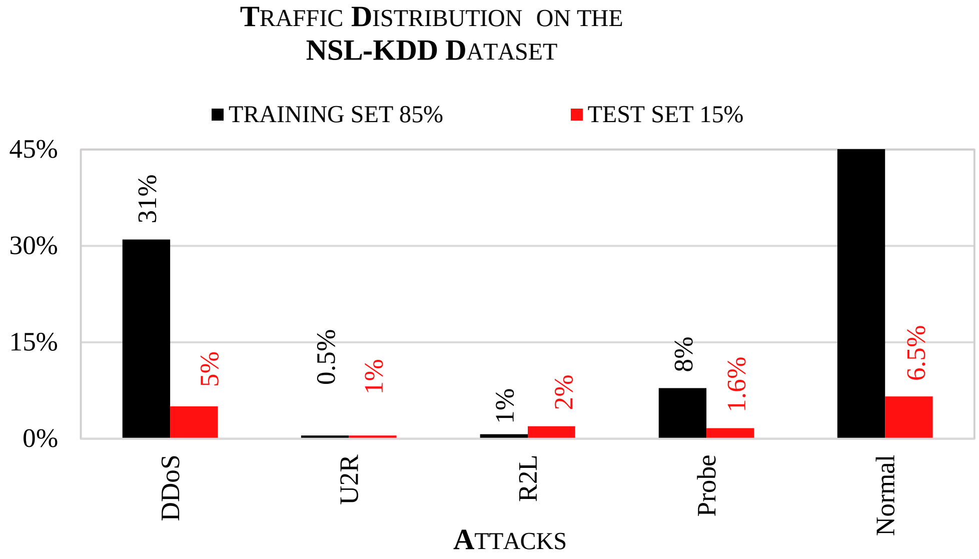


FIGURE 1: Traffic distribution on the NSL-KDD dataset.

5.1.2. The CIC-DDoS2019 Dataset

The CIC-DDoS2019 dataset was introduced by the Canadian Institute for Cybersecurity [19] and contains 50,063,112 records of DDoS attacks and 56,863 lines of benign traffic. Seven attacks are included in the test dataset: Port Scan, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, and SYN, in addition to normal records in the testing day on the CIC-DDoS2019 dataset. We noted the need to reduce the features of the original dataset due to the presence of correlated features that may lead to poor classification, so we utilized cross-correlation on 86 features and excluded similar features that affect the quality of classification, according to results, we specified only 66 dissimilar features from the total 86 presented in the CIC-DDoS2019. Then we created a sub-dataset named CIC-DDoS2019-1, which includes enough parts of the CIC-DDoS2019 traffic records according only to the attack types presented in the test dataset of CIC-DDoS2019, each record in the created dataset is described by 66 dissimilar features. Table 2 below shows the selected numbers of attacks and normal records in the training and test records of CIC-DDoS2019-1. Figure 2 shows the distribution of traffic in the training and test sets of CIC-DDoS2019-1, respectively.

TABLE 2: Attacks and normal traffic in the CIC-DDoS2019-1 dataset

TRAFFIC TYPE		TRAINING RECORDS	TEST RECORDS
ATTACKS	Port Scan	7000	3000
	NetBIOS	42000	18000
	MSSQL	43400	18600
	LDAP	39900	17100
	UDP	39130	16770
	UDP-Lag	10500	4500
	SYN	38500	16500
TOTAL ATTACKS		259000	110000
NORMAL		25900	11100

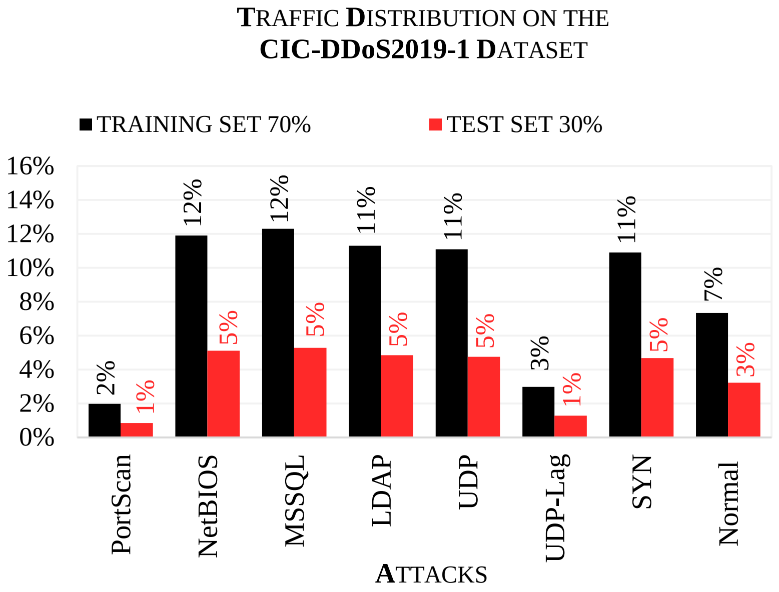


FIGURE 2: Traffic distribution on the CIC-DDoS2019-1 dataset.

5.2. Data Preprocessing

Data preprocessing encoded and normalized the values of the features in the two datasets as follows:

1- For columns with values of type "String", the string values were converted to integer values by giving each string value a distinct number from 1 to n.

2- For columns with values of type "double" that have two decimal places, the values were converted to integers by multiplying by 100 to dispose of the floating point. For columns with integer values, no modification is required.

3- Next, the normalization rule [(X-X_{MEAN})/X_{MAX}] is applied to the vector of all attributes, keeping the values in the range [-1 1]. In the training and test outputs, normal packets will receive 0, and abnormal packets, which indicate intrusion, will receive 1.

5.3. Classifiers' Architecture

We developed two MLN classifiers based on RBP Classifiers I and II were developed to work with NSL-KDD and CIC-DDoS2019-1, respectively. Each classifier contains four layers: one input layer, two hidden layers, and one output layer. The number of neurons in classifier I is [41, 20, 7,1]and in classifier II [66, 30, 8, 1]. The sigmoid function $F(x)=1/(1+e^{-X})$ was used as the activation function, which gives the RBP classifier the possibility to minimize errors better than other activation functions. Figure 3 shows the architecture of the developed classifiers 1 and 2.

5.4. Learning Algorithm

RBP is considered one of the best methods in this regard because of its necessity for nonlinear binary prediction; RBP has been proposed in the artificial intelligence literature to address the weaknesses of the classical backpropagation algorithm, BP, and its drawbacks with long training times and to speed up the learning process [18].

RBP works by directly adapting the weights based on local gradient information at each learning iteration, thus modifying the weights and bias network. The mean squared error (MSE) values are reduced and accuracy is improved. The algorithm elaborates each weight separately. If the sign of the partial derivation of the error changes compared to the sign of the previous iteration, the updated value of the correlated weights is multiplied by a factor of η-, where 0 <η−<1, and if the sign remains the same as in the last iteration, the updated value is multiplied by a factor of η +, where η+ > 1. That is, each weight is modified by the updated value of the associated partial derivation in the opposite direction. η+ and η- are empirically chosen to be 1.2 and 0.5, respectively. The algorithm starts by giving the initial value of each weight wii and its updated value Δj (t). The updated values are then changed according to a learning rule based on the sign of the partial derivation of the associated error, and the weight values on which they depend are updated. The mathematical conditions described above are illustrated in Equation 1 below.

$$\Delta_{ij}(t) =$$

$$\eta^{+}.\Delta_{ij}(t-1) \quad if \quad \frac{\partial E}{\partial w_{ij}}(t).\frac{\partial E}{\partial w_{ij}}(t-1) > 0$$

$$\eta^{-}.\Delta_{ij}(t-1) \quad if \quad \frac{\partial E}{\partial w_{ij}}(t).\frac{\partial E}{\partial w_{ij}}(t-1) < 0$$

$$\Delta_{ij}(t-1) else$$

Where $0<\eta-<1<\eta$ (1)

Whenever Equation 2 reveals a case where the partial derivation associated with wiŅ changes sign, it indicates that a large update value has occurred and the algorithm has exceeded the local minimum of the error, so the update value Δ(t) is decreased or increased by a factor η- based on positive or negative partial derivation, respectively.

$$\Delta w_{ij} =$$

$$-\Delta_{ij}(t) \quad \text{if} \quad \frac{\partial E}{\partial W_{ij}} > 0$$

$$\Delta_{\mathrm{ij}}(\mathrm{t}) \quad \text{if} \quad \frac{\partial \mathrm{E}}{\partial \mathrm{W_{ij}}} < 0$$

$$0 \quad \text{else} \qquad (2)$$

Equation 3 shows that when the correlated partial error derivation is negative, the new updated weights are computed from the previous ones by adding the updated values. The converse is to subtract the updated values when the derivation of the correlated partial error is positive.

$$\mathrm{W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij}(t)} \qquad (3)$$

Equation 4 shows an exceptional case where the derivation of the partial error changes, indicating that the local minima of the error were skipped because the previous update was too large. In this case, the algorithm undoes the previous weight update.

$$\mathrm{\Delta w_{ij}(t) = -w_{ij}(t-1)}$$

$$\text{If } \frac{\partial \mathrm{E}}{\partial \mathrm{w_{ij}}}(\mathrm{t}).\frac{\partial \mathrm{E}}{\partial \mathrm{w_{ij}}}(\mathrm{t}-1) < 0 \qquad (4)$$

The derivation of the partial error may change sign again in the next step, in which case the algorithm avoids a double update by setting Ewij (t-1) = 0 in the above update rule in Equation 4. Equation 5 describes the calculation of the partial derivation of the total error

.

$$\frac{\partial E}{\partial W_{ij}} = \frac{1}{2}\sum_{p=1}^{p}\frac{\partial E_p}{\partial W_{ij}} \qquad (5)$$

On the other hand, the RBP algorithm is much faster than the standard steepest descent algorithm. The presented RBP classifiers (1 and 2) were trained on our training NSL-KDD and CIC-DDoS2019-1 datasets and achieved mean squared errors of 0.01 and 0.0036 at epoch 600 and epoch 3628, respectively. The error of the network during the training phase was not indicative of an overfitting problem. In Figure 4 below, Section A shows the performance of the RBP classifier I applied to the NSL-KDD dataset during the training phase. Section B, on the other hand, shows the performance of the training phase when RBP classifier II is applied to CIC-DDoS 2019-1. Noteworthy in these two sections is the stability of the local minima of the error in the training phase of our two proposed AIDS, which indicates the stability of the classifier's learning algorithm.

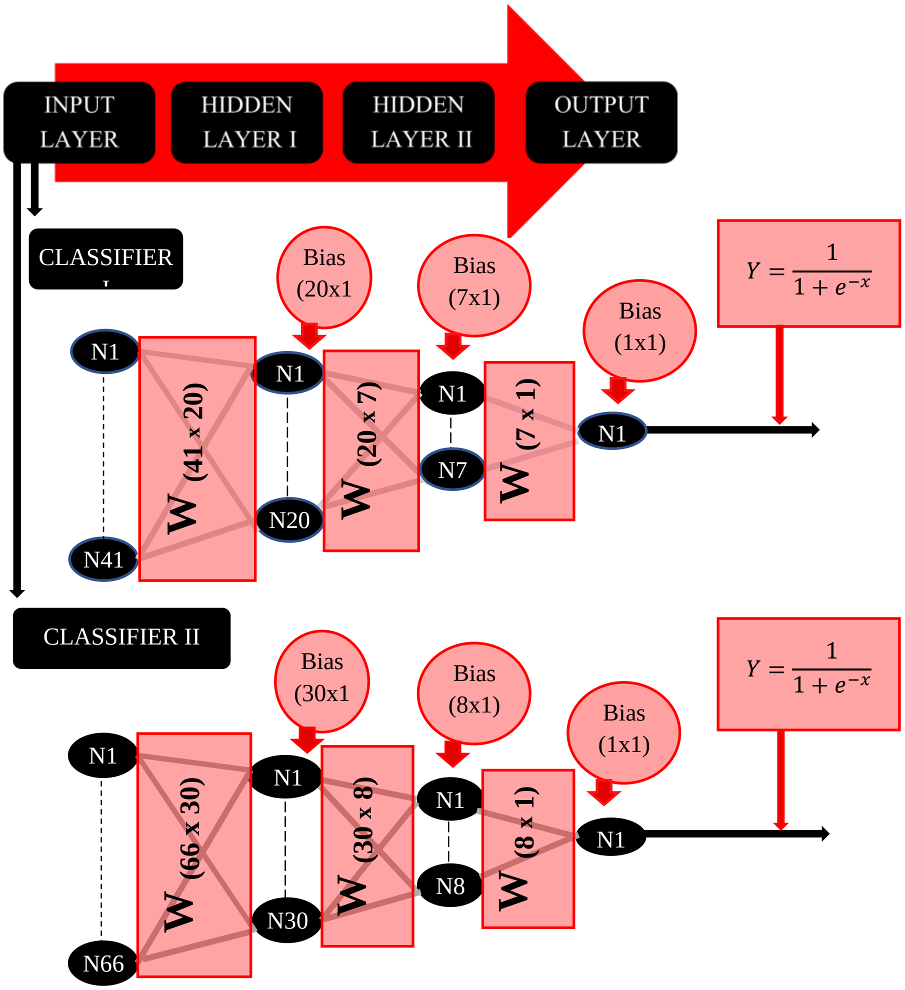


FIGURE 3: Architecture of AIDS Classifiers I and II

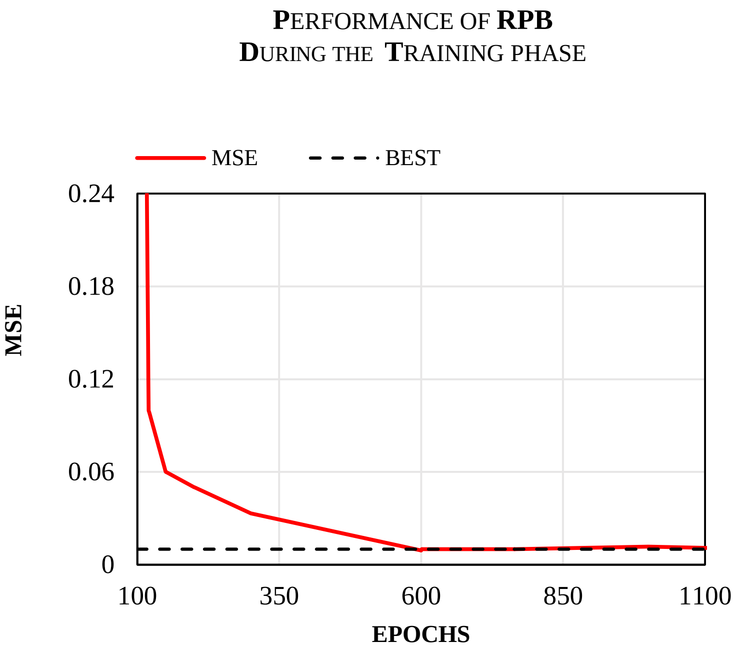
5.5. Testing and Validation

The trained classifiers were tested on a test dataset to classify positive cases representing anomalous intrusions and negative cases representing normal traffic in the network. One hundred different threshold values in the range of [0, 1] were applied to the output. The best results were obtained by applying a threshold of 0.03 to the output of classifier I and a threshold of 0.11 to the output of classifier II. We then organized the results of the testing phase using the associated confusion matrices for each classifier. Each confusion matrix contains a set of measurements, where Cp, Cn, Tp, Fp, Tn, and Fn are the number of true positives, true negatives, true positive detections, false positive detections, true negative detections, and false negative detections, respectively. Sensitivity Se and specificity Sp represent the system's ability to correctly detect positive (intrusion) and negative (normal) cases, respectively; accuracy Acc represents the system's ability to correctly detect both negative and positive cases; and test error Err represents the system's error in detecting both positive and negative cases. See Equations 6, 7, 8, and 9. Table 3 shows the confusion matrix of the classifiers obtained from the testing phase, and Figure 5 shows the results of the statistical evaluation metrics associated with the two classifiers developed.

$$\mathrm{S_e} = \frac{\mathrm{T_p}}{\mathrm{C_p}} \qquad (6\ ) \qquad\qquad \mathrm{S_p} = \frac{\mathrm{T_n}}{\mathrm{C_n}} \qquad (7)$$

$$\mathrm{A_{cc}} = \frac{\mathrm{T_P} + \mathrm{T_n}}{\mathrm{C_p} + \mathrm{C_n}} \qquad (8) \qquad\qquad E_{rr} = 1 - \mathrm{A_{cc}} \qquad (9)$$

(A)



(B)

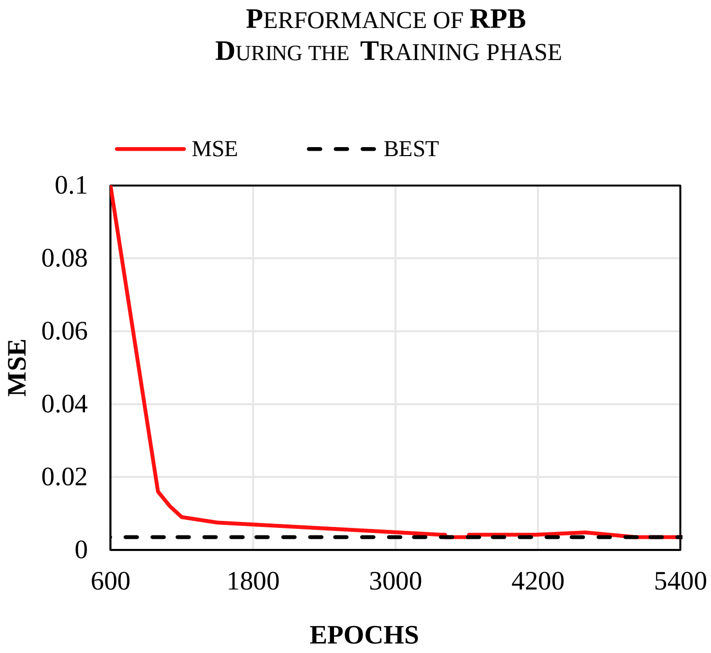


FIGURE 4: Performance of the AIDS classifiers during the training phase for (A): classifier I, (B): classifier II.

TABLE 3: Confusion matrix of the RBP classifiers in the testing phase.

CLASSIFIERS	CONFUSION MATRICES			
CLASSIFIER I	Tp	Tn	Fp	Fn
	12674	9672	39	159
CLASSIFIER II	Tp	Tn	Fp	Fn
	258498	11045	55	502

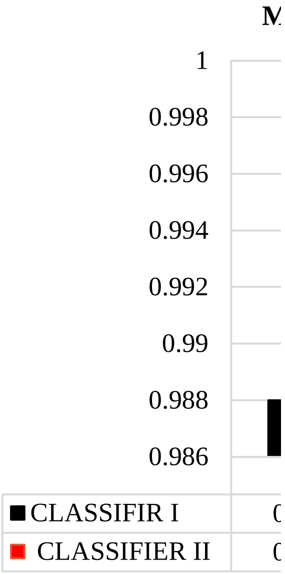


FIGURE 5: Performance of the AIDS classifiers during the testing phase.

As an important evaluation measure in the testing phase, we also employed the receiver operating characteristic curve (ROC), which represents the relationship between the probability of a true positive detection case (Se) and the probability of a false positive detection case, calculated from the specificity index: (1-Sp) The probabilities used in the ROC curve are 100 threshold values in the range [0, 1]. The probabilities are computed according to the area under ROC (AUC) was also calculated as an important indicator to evaluate ROC efficiency. Figure 6 shows (A) the ROC curve for classifier I and (B) the ROC curve for classifier II. Our AIDS achieved significant results in this regard, with AUC values of approximately 0.972 and 0.982 for classifiers 1 and 2, respectively. Figure 6

The proposed two classifiers were trained and tested with other learning algorithms of MLNN: LM, BR, and QN. These different algorithms except RBP have been previously applied in many related studies but didn't prove an efficient performance in any proposed work. Our obtained results in this regard proved the success of RBP which has not been employed in previous works to be an efficient AIDS learning algorithm. Table 4 shows a comparison of the performance of these different algorithms including RBP in the training phase, and Figure 7 shows the same comparison in the testing phase. Figure 8 illustrates two classification examples (intrusion and normal) on two records from the NSL-KDD dataset by classifier I, and Figure 9 illustrates two examples (intrusion and normal) on two records from the CIC-DDoS219-1 dataset by classifier II.

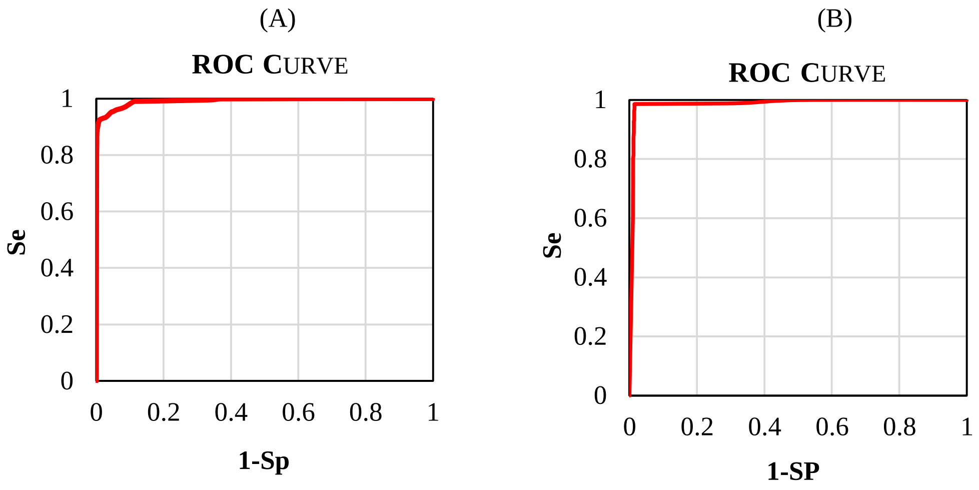


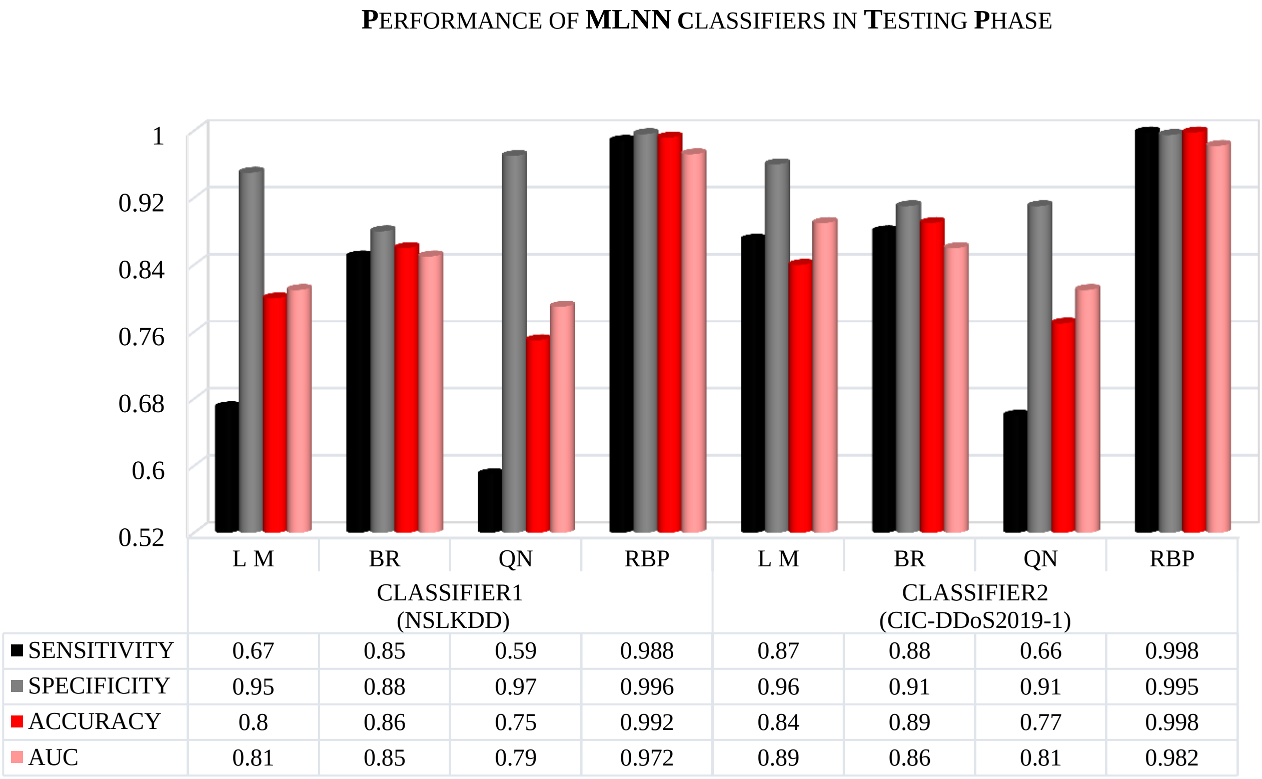
FIGURE 6: ROC curve for (A): classifier I and (B): classifier II

TABLE 4: Performance comparison of various AIDS learning algorithms during the training phase

CLASSIFIER	**LEARNING ALGORITH**	**PERFORMANCE**
Classifier I	L M	Local optimum of errors is not stable
	BR	Local optimum of errors is not stable
	QN	Learning stops early with a suboptimal error
	RBP	Optimal and stable performance
Classifier II	LM	Learning stops early with a suboptimal error
	BR	Local optimum of errors is not stable
	QN	Learning stops early with a suboptimal error
	RBP	Optimal and stable performance

6. Conclusion

The developed RBP classifiers successfully demonstrated the ability to distinguish normal from abnormal behavior among many other approaches used in ML and showed excellent performance in both training and testing phases using selected standard datasets used in many contributions in the related literature. Our approach achieved a prediction accuracy of 0.992% for classifier I and 0.996% for classifier II. These results are considered highly competitive with the results of other peer studies. Therefore, it can be said that the main objective of this study has been achieved. On the other hand, as to whether our proposed AIDS is effective in classifying other different data sets and cases, the limitation is that our developed AIDS must work with the same architecture or the same number of selected features as our selected dataset.



	L M	BR	QN	RBP	L M	BR	QN	RBP
	CLASSIFIER1 (NSLKDD)				CLASSIFIER2 (CIC-DDoS2019-1)			
■ SENSITIVITY	0.67	0.85	0.59	0.988	0.87	0.88	0.66	0.998
■ SPECIFICITY	0.95	0.88	0.97	0.996	0.96	0.91	0.91	0.995
■ ACCURACY	0.8	0.86	0.75	0.992	0.84	0.89	0.77	0.998
■ AUC	0.81	0.85	0.79	0.972	0.89	0.86	0.81	0.982

FIGURE 7: Performance comparison of suggested AIDS structures during the testing phase of different learning algorithms

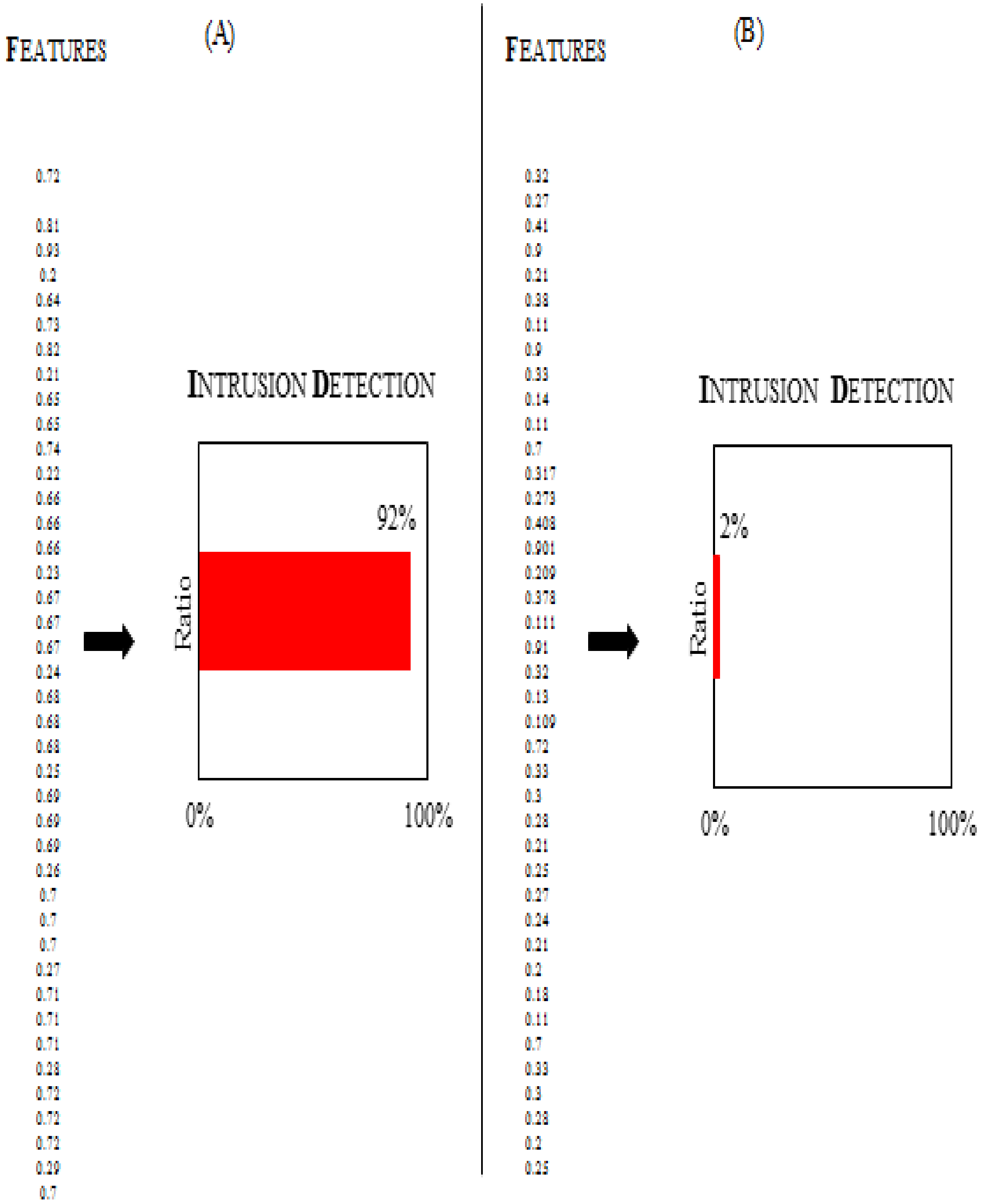


Figure 8. Two classification examples (A): intrusion and (B): normal) from the NSL-KDD dataset by classifier I

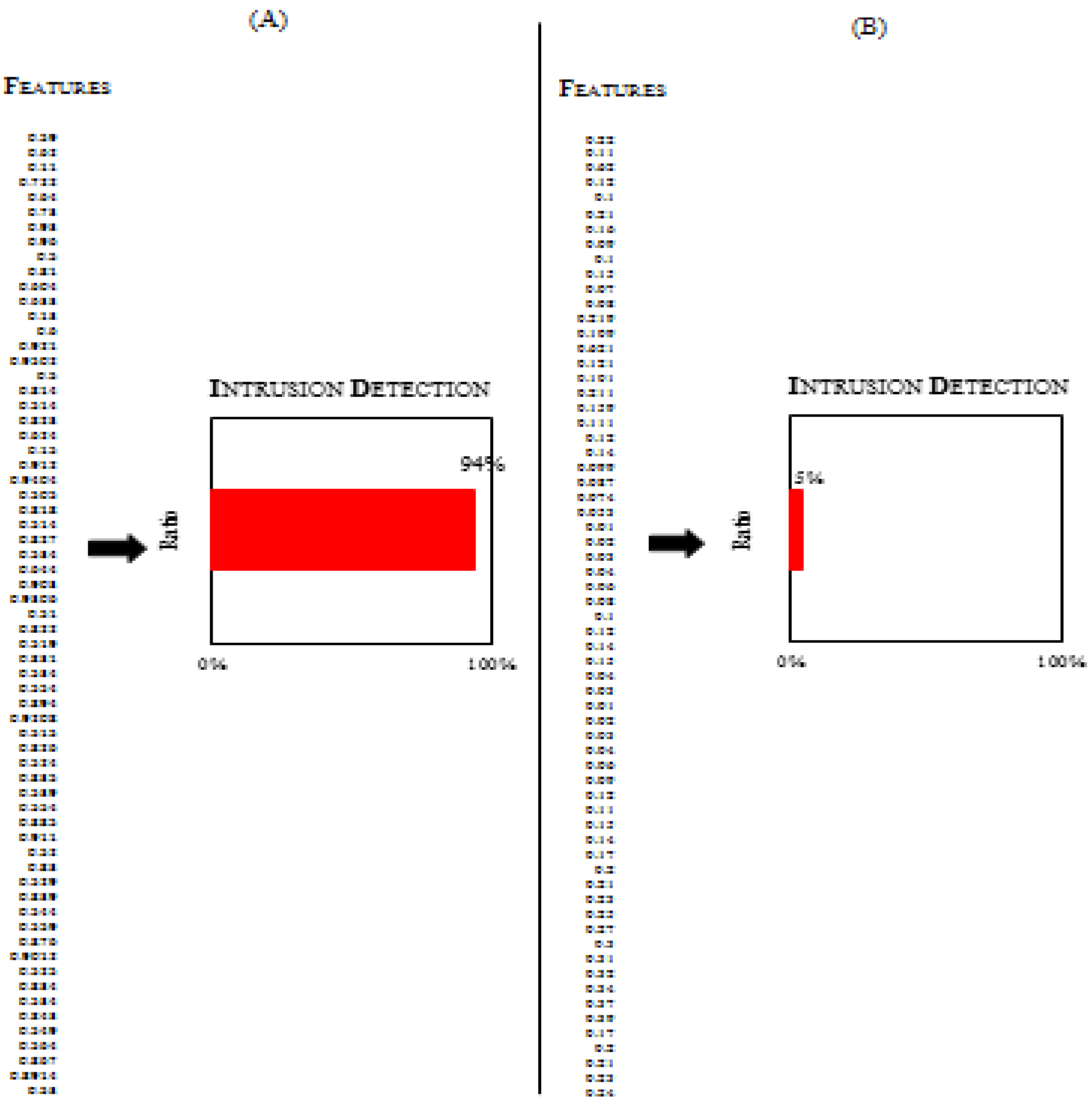


Figure 9. Two classification examples (A): intrusion and (B): normal from the CIC-DDoS2019-1 dataset by classifier II

7. Future Work

Our future work will concentrate on finding adaptive AIDS applicable to different computer network environments and dataset architectures, e.g., IoT, despite differences in the characteristics of measured cyber-attacks between different research environments. Currently, there are many attempts in related scientific research, many of which propose new methods based on well-known ML tools such as SVM. These topics include standardizing and minimizing the size of the features selected in AIDS and smoothing their H-P nonlinearity to be linearly approximated (there are new contributions in our previous work in this regard) [1]. Thus, in this case, the unique AIDS can be applied to different environments and data architectures.

8. Data Availability

As mentioned in sections 5.1.1 and 5.1.2 of this paper, the datasets used in this study are available online free of charge.

References

[1] Abo Zidan, R. and Karraz, G. (2022), Gaussian Pyramid for Nonlinear Support Vector Machine, *journal Applied Computational Intelligence and Soft Computing, Hindawi, 1-9*, https://doi.org/10.1155/2022/5255346

[2] Alazab, A., Hobbs, M., Abawajy, J. and Alazab, M. (2012), Using Feature Selection for Intrusion Detection System, *International Symposium on Communications, and Information Technologies (ISCIT), IEEE*, 296-301, http://www.doi.10.1109/ISCIT.2012.6380910

[3] Atiq, A. (2008), *Introduction of Soft Computing Systems for Software Security Management, Grin*, Project report, 1-10.

[4] Balyan, A. K., Ahuja, S., Lilhore, U. K., Sharma, S. K., Algarni, PA. D. & Raahemifar, K. (2022), A Hybrid Intrusion Detection Model Using EGA-PSO and Improved Random Forest Method, *Sensors*, *MDPI*, 22(16):5986 https://doi.org/10.3390/s22165986

[5] Butun, I., Morgera, S. D. and Sankar, R. (2013), A Survey of Intrusion Detection Systems in Wireless Sensor Networks, *Communications Surveys & Tutorials*, *IEEE*, 16(1), 266 – 282. http://www.doi.10.1109/SURV.2013.050113.00191

[6] Fakiha, B. (2022), Detecting Distributed Denial of Services Using Machine Language Learning Techniques, *Journal of Southwest Jiaotong University*, 57(5), 675-688.

[7] Ferrag, M. A., Shu, L., Djallel, H. and Choo, K-K. R. (2021), Deep Learning–Based Intrusion Detection for Distributed Denial of Service Attack in Agriculture, *Electronics*, *MDPI*, 10(1257), 1-26. https://doi.org/10.3390/electronics10111257

[8] Gu, J., & Lu, S. (2021), An Effective Intrusion Detection Approach using SVM with Naïve Bayes Feature Embedding, *Computers & Security, Science Direct*, 103, 1-16. https://doi.org/10.1016/j.cose.2020.102158

[9] Incorporated International Data Group Inc. IDG (2003), *Network World*, 20(45).

[10] KDD CUP 99
http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[11] Khraisat, A., Gondal, I., Vamplew, P. and Kamruzzaman, J. (2019), Survey of Intrusion Detection Systems, Techniques, Datasets and Challenges, *Cybersecurity*, 2(20), 1-22. https://doi.org/10.1186/s42400-019-0038-7

[12] Kumar, P. G. and Devaraj D. (2010), Intrusion Detection Using Artificial Neural Network with Reduced Input Features", *ICTAC Journal on Soft Computing*, 1(1), 30-36.
http://www.doi. 10.21917/ijsc.2010.0005

[13] Mahdavisharif, M., Jamali, S. and Fotohi, R. (2021), Big Data-Aware Intrusion Detection System in Communication Networks: A Deep Learning Approach, *Journal of Grid Computing, Springer*, 19 (46), 1-28. https://doi.org/10.1007/s10723-021-09581-z

[14] Mehmood, M., Javed ,T., Nebhen, J., . Abbas, S., Abid, R., Bojja, G. R. & Rizwan, M. (2022), A Hybrid Approach for Network Intrusion Detection, *CMC-Computer Mater, Contin*, 91-107. http://www.doi.10.1007/s10723-021-09581-z

[15] Meliboev, A., Alikhanov, J. and Kim, W. (2022), Performance Evaluation of Deep Learning Based Network Intrusion Detection System across Multiple Balanced and Imbalanced Datasets, *Electronics*, *MDPI*, 11(4):515, 1-13.
https://doi.org/10.3390/electronics11040515

[16] Mishra, Er. S., Dr. Pradhan, S. K., and Dr. Rath, S. K (2018), Performance Analysis of Network Intrusion Detection System Using Back-propagation for Feed Forward Neural Network in MATLAB/SIMULINK, *International Journal of Computational Engineering Research (IJCER)*, 8(5), 58-65.

[17] Pawar, A. and Tiwari, N. (2023), A Novel Approach of DDOS Attack Classification with Optimizing the Ensemble Classifier Using a Hybrid Firefly and Particle Swarm Optimization (HFPSO), *International Journal of Intelligent Engineering and Systems*, 16(4), 201-214. http://doi.10.22266/ijies2023.0831.17

[18] Saputra, W., Tulus, M., Zarlis1, R. W. and Hartama, D. (2017), Analysis Resilient Algorithm on Artificial Neural Network Backpropagation, *Journal of Physics, IOP science, Conference Series* 930 012035, 1-6.
http://www.doi.10.1088/1742-6596/930/1/012035

[19] Sharafaldin, I., Lashkari, A. H., Hakak, S., Ghorbani, A.A. (2019), Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy, *Proceedings of the International Carnahan Conference on Security Technology (ICCST), Chennai, India*, 1–8,
https://www.unb.ca/cic/datasets/ddos-2019.html

[20] Sokkalingam, S. & Ramakrishnan, R. (2022), An Intelligent Intrusion Detection System for Distributed Denial of Service Attacks: A Support Vector Machine with Hybrid Optimization Algorithm-based Approach, *Concurrency and Computation Practice and Experience, Wiley*, 1-18.
http://www.doi.10.1002/cpe.7334

[21] Sumathi, S., Rajesh, R. and Lim, S. (2022), Recurrent and Deep Learning Neural Network Models for DDoS Attack Detection, *Journal of Sensors, Hindawi*, 1-21. https://doi.org/10.1155/2022/8530312

[22] Tavallaee, M., . Bagheri, E., Lu, W. and Ghorbani, A. A. (2009), A Detailed Analysis of The KDD CUP 99 Dataset, Symposium on Computational Intelligence for Security and Defense Applications (CISDA), *IEEE*, 1-6. NSL-KDD official public site: https://www.unb.ca/cic/datasets/nsl.html

[23] Vacca, J. (2009), Computer and Information Security Handbook", 1st Edition, Elsevier, Chapter 18, "Intrusion Detection and Prevention Systems", eBook ISBN: 9780080921945, 356-376.

[24] Venkatesan, S. (2023), Design an Intrusion Detection System based on Feature Selection Using ML Algorithms, *Mathematical Statistician and Engineering Applications*, 72(1), 702-710. https://doi.org/10.17762/msea.v72i1.2000

[25] Yaqoob, Sh., Hussain, A., Subhan, F., Pappalardo, G. and Awais, M. (2023), Deep Learning Based Anomaly Detection for Fog-Assisted IoVs Network, *IEEE Access*, 11, 19024-19038. http://www.doi.10.1109/ACCESS.2023.3246660