**Engineering and Technology Journal**
**Journal homepage:** https://etj.uotechnology.edu.iq

# Building an Efficient System to Detect Computer Worms in Websites Based on Ensemble Ada Boosting and SVM Classifiers Algorithms

**Ali K. Hilool** *, **Soukaena H. Hashem, Shatha H. jafer**

Computer Science Dept., University of Technology-Iraq, Alsina’a street, 10066 Baghdad, Iraq.
*Corresponding author Email: Ali1995a1995a@gmail.com

## HIGHLIGHTS

- Union of two feature selection methods strength the NIDS
- Performance of the NIDS will increase by using ensemble learning
- Bagging and boosting by SVM have much more power than DT
- Worm detection is much more strongest by using NIDS with two levels

## ARTICLE INFO

**Handling editor:** Rana F. Ghani

**Keywords:**
computer worms
IDS
DM
Ada booting
SVM
ensemble learning.

## ABSTRACT

Computer worms perform harmful tasks in network systems due to their rapid spread, which leads to harmful consequences on system security. However, existing worm detection algorithms are still suffered a lot to achieve good performance. The reasons for that are: First, a large number of irrelevant data impacts classification accuracy (irrelevant feature gives estimator new ways to go wrong without any expected benefit also can cause overfitting, which will generally lead to decreased accuracy). Second, the individual classifiers used extensively in the systems do not effectively detect all types of worms. Third, many systems are built based on old datasets, making them less suitable for new types of worms. The research aims to detect computer worms in the network based on data mining algorithms for their high ability to automatically and accurately detect new types of computer worms. The proposal uses misuse and anomaly detection techniques based on the UNSW_NB15 dataset to train and test the ensemble Ada Boosting algorithm using SVM and DT classifiers. To select the most important features, we propose to conduct the similar features selected by Correlation and Chi-Square feature selection (since correlation finds the relations between features and classes whereas Chi finds whether features and classes are independent or not). The contribution suggests using SVM in the boosting ensemble algorithm as base estimators instead of DT to efficiently detect various types of worms. The system achieved accuracy, reaching 100% with CFS+Chi2fs and 99.38, 99.89 with correlation and chi-square separately.

## 1. Introduction

The Internet has become an essential part of the modern life of people because it is used in their education, communication, work, entertainment, and storage of their data. The expansion of the use of the Internet has led to several problems that exploit the weakness in a certain aspect to carry out harmful actions, such as using computer worms to shut down the network or steal data, etc. Computer worms are small, self-contained programs that do not require the assistance of others [1]. They are designed to carry out destructive activities, steal data from users while they are surfing the Internet, or damage them or their callers. Due to their superior ability to colorize, replicate, and elude detection, they spread quickly and are difficult to eradicate. The worm spreads more widely and faster than viruses because it automatically infects machines linked to the network and without human intervention [2]. The danger of worms is that they are independent and not dependent on other software that joins them, rapidly spreading them. Several attempts were used to detect computer worms and size their damages, such as using a firewall, encryption, machine learning techniques, and many other attempts [3].

IDS is software that detects any activity that is normal or malicious. It is one of the most reliable systems for detecting penetrations and attacks [4]. IDS is generating several false alarms. This problem has encouraged many researchers to find a solution to distinguish alerts to the less important incident and reduce false alarms, which are false positive (FP) and false negative (FN). Based on data mining technique, IDS can enhance IDS in real time, remove the normal activity from alarm data for focusing on real attacks, and find an abnormal activity that uncovers a real attack. It's a computational framework for finding patterns in data sets that use approaches from artificial intelligence, machine learning, and database systems. Different parameters may be used by data mining applications to analyze various data sets. [5, 6]. Network Intrusion Detection Systems (NIDS) have

http://doi.org/10.30684/etj.v40i4.2148
Received 28 April 2021; Accepted 23 July 2021; Available online 25 April 2022
2412-0758/University of Technology-Iraq, Baghdad, Iraq

become the most important component of recent network infrastructure due to increased security threats nowadays. The intrusion Detection System (IDS) generates a good number of alarms. However, algorithmic procedures are deployed to reduce false positives [7, 8, and 9]. Ensemble learning is a machine learning technique that involves training a group of poor learners (models) to solve a problem and then combining their results to produce better results. The basic idea is that combining weak models in the right way can get more accurate and/or robust models. Ensemble approaches are divided into three categories. Bagging that combines homogeneous weak learners, trains and tests them in parallel, and then combines them using voting, average, and other methods. Boosting brings together homogeneous poor learners and trains and tests them sequentially (each iteration depends on the previous ones). Stacking is an ensemble method in which a new model learns how to combine the predictions of numerous existing models in the most effective way possible[10]. This article aims to detect computer worms in the network based on data mining algorithms for their high ability to automatically and accurately detect new types of computer worms. The remainder of the paper is laid out as follows: In Section 2, we discuss the related work of worm detection. In Section 3, we introduce the theoretical background of the model. In Section 4, we introduce the worm detection system architecture based on ensemble Ada boosting. Preprocessing is covered in section 4. Section 4.II introduces the train and test model and builds a classifier (DT and SVM). Our comprehensive experiments in assessing the proposed worm detection system are discussed in Section 5. Section 6 concludes by explaining the conclusion.

## 2. Related work

Worm detection, as an important tool in computer systems for ensuring the security of cyber systems, regularly draws the research community's attention. While several solutions to improve worm detection efficiency have been suggested, we only consider work that falls under the ML-based IDS umbrella, uses dimensionality reduction or ensemble classification and other data mining techniques, and focuses on hybrid approaches in this section.

Gautam and Doegar [11] suggested an Intrusion Detection Approach based on Ensemble methods. They used three algorithms which are the native Bayes adaptive boost part. Also, they used information gain to remove redundant features. They also combined the results of the three classifiers by using the average or majority of voters.

Yuyang Zhou et al. [12] suggested an IDS based on ensemble classification that uses forest by penalizing attributes algorithms, c4.5 decision tree, and random forest to train and test three datasets which are NSL-KDD, AWID, and CIC-IDS2017. Furthermore, the proposed model uses CFS-BA, which combines correlation and Bat algorithms to remove irrelevant features. Finally, To combine the probability distributions of the base learners, the voting technique was used.

Jing and Chen [13] suggested an intrusion detection system(IDS) by using a support vector machine(SVM) classifier with the UNSW-NB-15 dataset. The authors did not use any feature selection method. Instead, they used nonlinear scaling in preprocessing stage instead of the min-max normalization algorithm. They say it gives better results with the UNSW-NB-15 dataset than the min-max normalization process.

Thanh and Lang [14] suggested a fuzzers detection system using the UNSW-NB15 dataset. The system use ensemble methods such as Bagging, Ada Boost, Stacking, Decorate, and Random Forest for fuzzers detection. The Ada Boost decision tree method has the highest classification quality, with an F Measure of 96.76 percent.

Pelin Yildirim Taser in [15] presents the bagging and boosting method based on six decision tree-based (DTB) classifiers used to predict diabetes based on experimental data. In terms of accuracy rates, a comparison is made between individual implementation, boosting, and bagging of DTB classifiers; experimental results show that AdaBoost with Naive Bayes Tree (NBTree) has the best accuracy score of 98.65 percent.

Shigeyuki et al.[16] examined default loans from a Taiwanese database and compared three learning models (boosting, bagging, and random forest) with different activation functions to eight neural-networks techniques and computed prediction accuracy for each one. The results show that boosting has the best classification power among the other two learning models. The number of middle layers in machine learning neural networks and the activation function used to affect their performance. For the training and testing sets, the maximum accuracy ratio of the original data is 71.01 percent and 69.59 percent, respectively. The maximum accuracy ratio of normalized data for the training and testing sets is 71.14 percent and 68.75 percent. We will discuss this literature in the Experimental Work and Results section.

## 3. Theoretical background

In this section, we will explain the theoretical side of the algorithms used in the research as follows

### 3.1 Feature selection

Feature selection techniques are one of the most important preprocessing steps in data mining techniques. They are used to eliminate unnecessary and redundant features from the dataset, improve the model's performance by using the correct features, and minimize the time it takes to process the data. We used correlation features selection and chi2 features selection in this study.

#### 3.1.1 Correlation Feature Selection

CFS(Correlation-based feature selection) uses a heuristic evaluation function based on correlations to rank attributes. The function evaluated attribute vector subsets correlated with the class label but not with each other. The CFS algorithm assumes that irrelevant features have a low correlation with the class and should thus be careless. On the other hand, excessive features should be investigated because they are frequently robustly correlated with more or one of the other attributes. The following is the criterion for evaluating a subset of n features:

$$M_s = \frac{N\overline{t_{cf}}}{\sqrt{N+N(N-1)\overline{t_{ff}}}} \qquad 1$$

where $M_S$ denotes the assessment of a subset of S containing N features. The average correlation value between characteristics and class labels is denoted by $\overline{t_{cf}}$. The average correlation between two features is denoted by $\overline{t_{ff}}$.[17].

### 3.1.2 Chi-Square Feature Selection

The Chi2 test is a statistical test. The Chi2 test determines the dependency between a class and a feature, allowing it to identify more pertinent features for a specific dataset successfully. As a result, we can remove features from the feature space that aren't useful for classification [18]. For example, we will get observed count A and predicted count E from the data of two features. The Chi-Square test determined how far predicted count E and observed count A deviates from each other.

$$X_c^2 = \frac{(A_i-E_i)^2}{E_i} \qquad 2$$

Where C is the degree of freedom, A is the observed value(s), and E is the expected value(s). After calculation $X_c^2$ we compare it to the chi2 table value where alpha =0.05 and drop the feature if it is less than the chi2 table value (independent); otherwise, the feature will be accepted.

## 3.2 ADA-boost algorithm

The ADA-boost algorithm is a machine learning algorithm that starts by giving all instances in the training dataset equal weight. The learning algorithm is then used to create a classifier for this data by creating the number of the stumps (nodes with two leaves) as the same number of features. Then only one stump is selected after calculating Gini and Entropy for all trees with the lowest value (Gini or Entropy).then calculated, the total error (TE) and the performance of the stump is calculated. So we must increase the weight for the misclassified records and decrease the weight for the correctly classified instances and update weights for all instances of the dataset based on the performance of the stump. Then new dataset based on normalized weights is created. The algorithm will again create a new stump depending on this new dataset. It will repeat the same process until it sequentially passes through all trees and finds less error than the normalized weight that we had in the initial stage [19]. Instead of using stumps, we suggested using the SVM classifier to update the Ada-Boost algorithm, and all steps after creating stumps are the same.

## 3.3 Support vector machine

SVM is a supervised learning model in which The input data is represented as an n-dimensional feature space. Space is then divided into two parts by an (n-1) dimensional hyperplane. The Yi matrix labels n-dimensional input data xi (i = 1, 2,..., l) as Yi = 1 for class 1 and Yi = 1 for class 2. For linearly separable data, a hyperplane can be defined.

$$\mathrm{F(x)=W.X+p=} \sum_{i=1}^{n} WiXi + p = 0 \qquad (3)$$

The decision function is Sgn (f(x)), W is an n-dimensional vector, and p is a scalar in Eq. (3). These determine the position of the hyperplane that completely separates the space, and it must adhere to the following constraints:

$$\mathrm{Yi\ (W.Xi+p)\text{-}1} \geq 0 \Rightarrow \begin{pmatrix} f(Xi)=W,Xi+p\geq 1 \ \ Yi=+1 \\ f(Xi)=W,Xi+p\leq -1 \ \ Yi=-1 \end{pmatrix} \qquad (4)$$

An ideal hyperplane is a hyperplane that produces the maximum limit. The independent variable is Si, and the error penalty is C in the following equation. The hyperplane's minimal solution is:

$$\emptyset(W,S) = \frac{1}{2}(W.W) + C(\sum_{i=0}^{I} Si \qquad (5)$$

Based on:

$$\mathrm{Yi\ [(W.X)+p]\geq 1\text{-}Si}, \quad \mathrm{i=1,2,3.....I} \qquad (6)$$

The distance between the sample xi and the limit on the other side of the limit is measured by Si. This calculation can be made easier by using the following formula:

$$\mathrm{V}(\emptyset) = \sum_{i=1}^{I} \propto i - \frac{1}{2}.\sum_{i,j=1}^{I} \propto i \propto jYiYjker(Xi,Xj) \qquad (7)$$

Based on:

$$\sum_{i=1}^{I} XiYi = 0 \ \ C \geq \propto \geq 0 \ \ i = 1,2,3 \dots . I \qquad (8)$$

The dot product of the feature space mappings of the original data points is returned by the kernel function Ker(XiXj)[20].

## 4. Proposal worms Detection System

To increase the ability to detect worms in networks, we propose an efficient data mining model for worm detection in which both the misuse and anomaly detection techniques are used in the detection of worms. Each instance in a dataset is labeled as "normal" or "attack"(the worms are one type of attack), and a learning algorithm is trained over the labeled data. Figure 1 shows the proposed worm detection model's framework. Which is divided into four main phases:

1) Dataset preprocessing: First, we apply preprocess steps to the original datasets to make data fit for the classification algorithm.
2) Dimensionality reduction: The feature selection approach based on correlation and chi-square features selection is used to select the most relevant features and reduce the dimensionality of the dataset.
3) Classifiers training: We use the ensemble Ada boosting classifier algorithm to build classifiers to improve the accuracy of worm detection.
4) Classification (testing) to predict the results of our model.

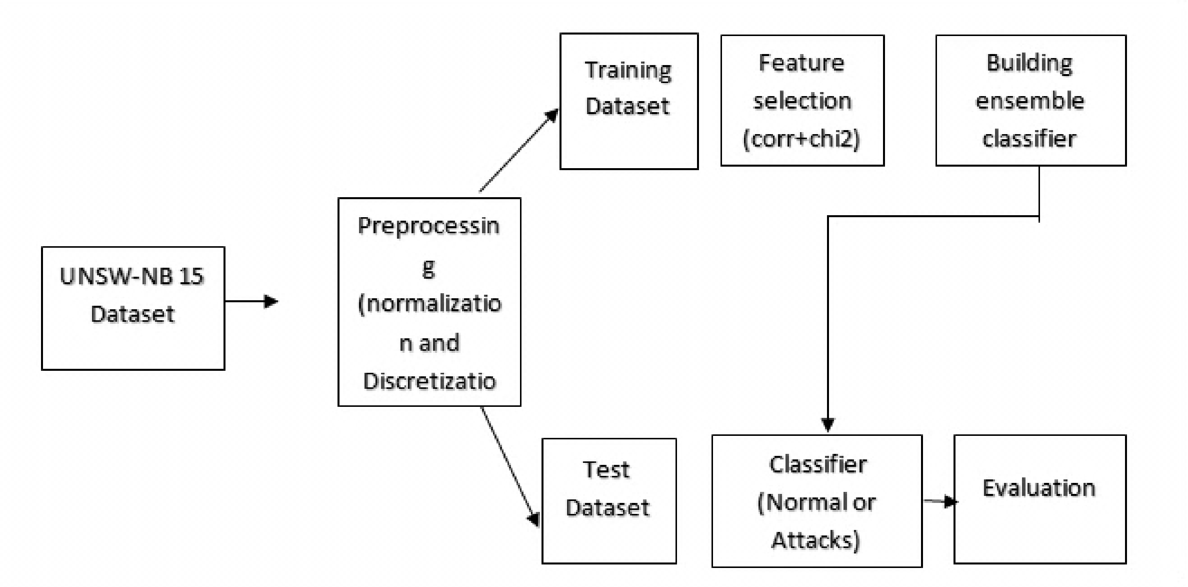
Training Dataset
Feature selection (corr+chi2)
Building ensemble classifier
UNSW-NB 15 Dataset
Preprocessin g (normalizatio n and Discretizatio
Test Dataset
Classifier (Normal or Attacks)
Evaluation

**Figure 1:** flowchart of proposed worm detection

## 4.1 Preprocessing

The UNSW-NB15 dataset was created by the Australian Center for Cyber Security (ACCS) in collaboration with several international researchers to use for IDS. The IXIA PerfectStorm program was used to create a rich hybrid collection of normal and abnormal contemporary network traffic. This dataset is used to train the proposed model. This dataset contains 2,540,044 instances [21]. These records are distributed in 4 large CSV files. In addition to those files, there are separate training and testing sets. The train contains 175,341records, and the testing contains 82,332 records. It contains 45 columns, 1 for id and 44 for features. The names and Descriptions of the dataset features are shown in Table 1. 5000 records from mentioned training and testing records were selected to train the proposed model, contain 154 worms, and the Remaining contains normal and other types of attacks. The UNSW-NB15 dataset is described in detail in Table 2. Normal data and nine types of attacks are included in these training and testing datasets: Backdoors, DoS attack, Exploits attack, Fuzzers attack, Generic attack, Reconnaissance attack, Shellcode attack, and Worms attack. Because the algorithm deals with the binary classification, I changed the last column (45) to column no. (44). Because the latter represents the binary classification, 0 refers to the normal and 1 to the rest of the types of attacks, including the worm.

Because the UNSW-NB15 dataset contains both continuous and discrete features, it is necessary to convert the continuous attributes to discrete to ensure the system's efficiency and deal with the issue of new values appearing in the test dataset that are not present in the training dataset. Following discretization, we used the Min-Max normalization process to improve the model's efficiency and effectiveness by placing attribute values between 0 and 1. After discretization and normalization, we will use correlation feature selection and chi-square feature selection (the work's originality lies in combining these two methods) to exclude unused and redundant features from the dataset (See Algorithm 1). We used these methods in particular because we found them in experiments to be the most effective methods to select the features that fit our proposal and lead to increased accuracy and reduced false alarm rate. After all, the selected features are more relevant to our problem, which enhances the results.

**Table 1:** The UNSW-NB15 dataset's attack and normal categories are described in detail [22]

| Traffic Type | Description |
|---|---|
| Normal | Not-dangerous traffic |
| Analysis | Generic kind that shows the penetration of HTML file, port scanning, and spam |
| Fuzzing | Finding 'hackable' software issues via an automated technique that includes randomly feeding many permutations of data into a target application until one of the variations reveals a vulnerability. |
| Backdoor | The malware allows remote access to databases and file servers without standard authentication methods. |
| DOS | Invalid authentication attempts flood the network/server, causing it to crash or stall, preventing genuine users from accessing online services. |
| Exploits | Code that takes advantage of a flaw or weakness in software. Frequently used in malware, allowing for a rapid and easy propagation. |
| Generic | Collision attack on the secret keys of ciphers. It's effective against all block ciphers. |
| Reconnaissance | Methods for obtaining information about the target network/server that are simple to implement |
| Shellcode | A collection of instructions/statements injected and executed by a defective program. Directly manipulates the registers and functionalities of a program. |
| Worms | Malicious code that replicates itself. There is an excessive amount of system memory and network bandwidth being consumed. As a result, the system's availability has been lowered. |

**Table 2:** Details on UNSW NB-15 Dataset [14]

| Attacks types | Testing dataset (percent) | | Training dataset(percent) | |
|---|---|---|---|---|
| Normal | 56.000 | 31,94% | 31,94 | 44,94% |
| Analysis | 2.000 | 1,14% | 1,14 | 0,82% |
| Backdoor | 1.746 | 1,00% | 1,00 | 0,71% |
| Dos | 12.264 | 6,99% | 6,99 | 4,97% |
| Exploits | 33.393 | 19,04% | 19,04 | 13,52% |
| Fuzzers | 18.184 | 10,37% | 10,37 | 7,36% |
| Generic | 40.000 | 22,81% | 22,81 | 22,92% |
| Reconnaissance | 10.491 | 5,98% | 5,98 | 4,25% |
| Shellcode | 1.133 | 0,65% | 0.65 | 0,46% |
| Worms | 130 | 0,07% | 0.07 | 0,05% |
| Total | 175.341 | 100% | 100 | 100% |

## 4.2 Classification

After feature selection techniques, we will split the dataset into two parts training containing 67% from a total number of records in the dataset and testing containing 33% from a total number of records in the dataset. The two parts are used to train and test the proposed model. It's worth noting that the selection process was selected at random. Then I will use the Ada-Boost algorithm (See algorithm 2) with SVM as the base estimator. Algorithm 2 describes an Ada boost with an SVM classifier algorithm for detecting normal or attack records in the UNSW-NB15 dataset. To train the SVM base classifier via weighted sample, the weight function is measured based on the error value for each training sample. Suppose the weighted sample exceeds the threshold value $\emptyset$ . Then, using the Ada booster technique, the SVM classifier becomes more powerful. The strong classifier output results classify it as normal or attack based on the objective function.

## 5. Experimental work and Results

As previously stated, the goal of this paper is to create a high-accuracy worm detection system. A model called CFS-chi2 that combines CFS and chi2 is used to determine a subset of the original features to eliminate irrelevant features and improve classification efficiency. In addition, an Ada boost ensemble classifier is trained and tested during the classification stage based on the UNSW NB15 dataset. The experiments are carried out on a desktop PC equipped with a 1.80 GHz Intel Core i3-3217U processor and 4GB RAM.

The two classification models are built after the two chosen classifiers (SVM and DT) have been trained on the training dataset using ensemble Ada Boosting. Then, to ensure that the built models are valid and accurate, apply these two models to test dataset records. True positive (TP) normal, true negative (TN) attack, or false positive (FP) not normal, false negative (FN), not an attack, or unknown, such as user behavior or a new attack are the categorization findings of testing. The results of testing SVM and DT classifiers to classify the testing dataset records are shown in Table3. These results show that the TP is greater than TN, FP, and FN and unknown when selecting all features and feature selection methods. In the SVM classifier, the FP rates decrease when using feature selection methods from 15 to 10 when using CFS and 6 when using Chi2, and 0 when using CFS+Chi2fs.

**Table 3:** Classification Results of SVM and DT Classifiers

| Classifier | FS method | TP | TN | FP | FN | Unknown |
|---|---|---|---|---|---|---|
| SVM | ALL | 914 | 721 | 15 | 0 | 0 |
| | CFS | 925 | 715 | 10 | 0 | 0 |
| | Chi2FS | 921 | 723 | 6 | 0 | 0 |
| | CFS+Chi2fs | 940 | 710 | 0 | 0 | 0 |
| DT | ALL | 902 | 719 | 4 | 25 | 0 |
| | CFS | 905 | 721 | 1 | 23 | 0 |
| | Chi2FS | 932 | 698 | 3 | 17 | 0 |
| | CFS+Chi2fs | 923 | 714 | 0 | 13 | 0 |

To evaluate our proposed system, we will use several metrics shown in Table 4. These metrics are such as Accuracy ((TP + TN)/(TP + TN + FP + FN)), detection rate (DR=TP/(TP+FN)), false alarm rate (FAR=FP/(TN+FP)), true negative rate (TNR=TN/TN+FP), positive predictive value (PPV=TP/TP=FP), false-positive rate (FPR=FP/FP+TN), false discovery rate (FDR=FP/FP+TP), error rate, and area under the curve AUC which is a summary of the ROC curve that measures the ability of a classifier to distinguish between classes. We will apply all metrics on four different subsets of the UNSW-NB-15 dataset features (all features, correlation select 33 features, chi2 select 33 feature, 27 features from combining chi2 with correlation).

**Table 4:** Metrics to evaluate ensemble Ada boost with SVM

| Metrics | All | CFS | Chi2FS | CFS+Chi2fs |
|---|---|---|---|---|
| n. of .feature | 44 | 33 | 33 | 27 |
| Accuracy | 99.15 | 99.38 | 99.89 | 100 |
| DR | 1.00 | 1.00 | 1.00 | 1.00 |
| FAR | 0.0203 | 0.0137 | 0.0082 | 0.0 |
| TNR | 0.9796 | 0.9862 | 0.9917 | 1.00 |
| PPV | 0.9838 | 0.9893 | 0.9935 | 1.00 |
| FPR | 0.0203 | 0.0137 | 0.0082 | 0.0 |
| FDR | 0.0161 | 0.0106 | 0.0064 | 0.0 |
| Error Rate | 0.0090 | 0.0060 | 0.0036 | 0.0 |
| AUC | 99.23 | 99.55 | 99.88 | 100 |

Table 4 highlights the findings using the UNSW-NB15 dataset, including the ensemble SVM classifier results. It is suggested that without feature selection, the ensemble classifier is not optimum enough in several criteria. When feature selection methods are used, however, performance improves to the best possible scenario. in detail, our proposed system exhibits the accuracy of 99.15, FAR of 0.0203, TNR of 0.9796, PPV of 0.9838, FPR of 0.0203, FDR of 0.0161, the error rate of 0.0090, AUC of 99.23 without using feature selection methods. The results are optimized when using feature selection methods and reach the best possible case when using CFS+Chi2fs with the highest accuracy of 100, FAR of 0.0, TNR of 1.00, PPV of 1.00, FPR of 0.0, FDR of 0.0, error rate of 0.0, AUC of 100.

Table 5 shows the results of the evaluation Ada boost that uses DT with four different subsets of the UNSW-NB15 features(all features, correlation select 33 features, chi2 select 33 feature,27 features from combining chi2 with correlation).

**Table 5:** Metrics to evaluate ensemble Ada boost with DT

| Metrics | All | CFS | Chi2FS | CFS+Chi2fs |
|---|---|---|---|---|
| n. of .feature | 44 | 33 | 33 | 27 |
| Accuracy | 0.982 | 0.985 | 0.987 | 0.992 |
| DR | 0.973 | 0.975 | 0.982 | 0.989 |
| FAR | 0.0055 | 0.0013 | 0.0042 | 0.0 |
| TNR | 0.994 | 0.998 | 0.995 | 1.00 |
| PPV | 0.995 | 0.998 | .0.996 | 1.00 |
| FPR | 0.0055 | 0.0013 | 0.004 | 0.0 |
| FDR | 0.004 | 0.001 | 0.003 | 0.0 |
| Error Rate | 0.017 | 0.014 | 0.012 | 0.007 |
| AUC | 0.983 | 0.986 | 0.988 | 0.993 |

When applying DT classifier with ensemble Ada boosting algorithm Without using feature selection methods, our proposed system has an accuracy of 0.982, DR of 0.973, FAR of 0.0055, a TNR of 0.994, a PPV of 0.995, an FPR of 0.0055, an FDR of 0.004, an error rate of 0.017, and an AUC of 0. 983. When using feature selection methods, the results are optimized. Therefore, when using CFS+Chi2fs, the best case is reached with the highest accuracy of 0.992, DR of 0.989, FAR of 0.0, TNR of 1.00, and PPV of 1.00, an FPR of 0.0, an FDR of 0.0, an error rate of 0.07, and an AUC of 0. 993. To better understand the benefits of the suggested methodology, we compare our suggested system to the related work discussed in section 2. The results of the comparison are shown in Table 6.

**Table 6:** Compression between the proposed system and the related work

| Method | Dataset | Feature selection | n. of features | ACC | DR | FAR |
|---|---|---|---|---|---|---|
| Native Bayes | KDD-cup99 | Info- gain | 20 | 91.0 | 98.0 | N/A |
| Part | KDD-cup99 | Info-gain | 20 | 99.0 | 99.0 | N/A |
| Adaptive Boost | KDD-cup99 | Info-gain | 20 | 97.0 | 96.0 | N/A |
| C4.5 | CIC-IDS2017 | CFS-BA | 10 | 0.98 | 0.996 | 0.011 |
| RF | CIC-IDS2017 | CFS-BA | 10 | 0.993 | 0.995 | 0.003 |
| Forest-PA | CIC-IDS2017 | CFS-BA | 10 | 0.988 | 0.993 | 0.006 |
| SVM | Train and Test UNSW-NB15 | N/A | 44 | 0.85 | N/A | 15.26 |
| Bagging (SVM) | UNSW-NB15 | N/A | 49 | 0.659 | 0.935 | N/A |
| STACKING (SVM) | UNSW-NB15 | N/A | 49 | 0.628 | 0.935 | N/A |
| Boosting (SVM) | UNSW-NB15 | N/A | 49 | 0.947 | 0.882 | N/A |
| Ada boosting (random tree) | diabetes dataset | N/A | 17 | 0.961 | N/A | N/A |
| Boosting | The payment data in Taiwan | N/A | 23 | 0.71 | N/A | N/A |
| Proposed System | Subset of Train and Test UNSW-NB15 | CFS+Chi2s | 27 | 1.00 | 1.00 | 0.0 |

As shown in Table 6, the comparison comprises the method of classification, the selected dataset, the feature selection techniques, the number of selected features, accuracy, FAR, and DR for intrusion detection. When we compare our system with Naive Bayes, Part, Adaptive Boost, C4.5, RF, Forest-PA, and other comparable approaches in section 2, our suggested system gives the highest accuracy and detection rate, DR equal to 0.100. When comparing our suggested system to the SVM, we can observe that the ensemble method has advantages because the SVM is a single classifier with a high variance. As a result, ensembles frequently minimize the variance component of prediction mistakes made by contributing models, resulting in a significant increase in accuracy (from 0.85 to 0.100) and a decrease in the False alarm rate, FAR (from 15.26 to 0.0). When SVM was employed as a base estimator, our suggested system likewise achieved the highest accuracy and detection rate when compared to other ensemble approaches such as Bagging(SVM), stacking(SVM), and Boosting(SVM). Table 7 compares the work with the contribution and without the contribution.

**Table 7:** A comparison between the work with the contribution and without the contribution

| Method | Dataset | Feature selection | No. of features | ACC | DR | FAR |
|---|---|---|---|---|---|---|
| Ada boosting (DT) | Subset of Train and Test UNSW-NB15 | N/A | 44 | 0.98 2 | 0.97 3 | 0.0055 |
| Proposed System | Subset of Train and Test UNSW-NB15 | CFS+Chi2s | 27 | 1.00 | 1.00 | 0.0 |

The results show that the SVM algorithm performs better than the DT algorithm in classification. The SVM method using CFS+Chi2fs has a total accuracy of 100%, a DR of 0.100, and a FAR of 0.0. The DT method has an overall accuracy of 0.982 percent, a DR of 0.973, and a FAR of 0.0055.

## 6. Conclusions

The proposed system emphasizes the importance of using intrusion detection systems (IDS) in networks to detect worm attacks, which are considered the most dangerous attacks in a network and impact resource availability. Furthermore, the proposed system is more efficient due to the normalization and discretization processes. To improve the accuracy of the proposed system and reduce the amount of time required, the correlation and chi2 algorithms are suggested as feature selection methods. Using these algorithms improves classification accuracy, as shown in Tables IV and V. The accuracy of the Ada boost classifier that uses SVM supported by chi2+corr with 27 features is better than using all features or using the Ada boost Classifier with Corr or chi2 with 33 features. In addition, Chi2+corr reduces the false alarm rate compared to CFS or CHI2, as shown in Table IV. In contrast, when using a decision tree classifier is a base estimator in Ada boost(without our contribution ), the system will be less accurate, detect less, and have a false alarm rate, as shown in Table 7.

**Algorithm (1) Preprocessing**

```
input : subset unsw-nb15 Datasets
Output: data values ranging from zero to one, independent features with a strong connection to the class, and class-dependent features.

Start
Step 1: min-max normalization
establish upper and lower bounds (P, Q)                    // a particular range
determine the minimum and maximum values ($X_{min}$,$X_{max}$)
      for each data item, do
            $Value\_of(y) = \frac{X-X_{min}}{X_{max}-X_{min}}$ .(P-Q)+P       // x the value to be normalized
         End For
Step 2:  correlation CFS
         For each class column
             Extract  the correlation of class with all features
             Choose features that have a strong relationship with class.
             Remove the remainder
      End For
      For each feature in the subset you've chosen,
             Extract the correlation of feature with all features
         Remove the remainder
      End For
Step 3: Chi-square feature selection
       For each unsw-nb15 Dataset feature
            seek for $X_c^2$with class. See equation(1)
            alpha=0.05
            from the chi2 table, find  X_c^2' where alpha=0.05 and match it to X_c^2
          If $X_c^2 < X_c^{2'}$ the feature is independent (dropped)
            Else it depends on class (not drop)
      End For
End
```

**Algorithm(2)Ensemble of Ada Boost with SVM classifier**

```
Input: {(X1, Y1), (X2, Y2), … (Xn, Y)} is a set of the UNSW-NB15 training dataset.
Output: classification decisions(normal , attacks).
Start
the weight of the training subset(1/N) is Initialized.
For each UNSW-NB sample.
Error=$\sum^{n}_{i=1} W_{ti}(t)$          //Calculate the error to get the best weight.
$Bt = \frac{1}{2} Ln\left(\frac{1-Error}{Error}\right)$          // After the error is analyzed, set the sample weight.
Di+1(j)=$\frac{Di(j)}{Qi} * \begin{bmatrix} exp(-Wi) if h(Xi) = Yi \\ exp(-Wi) if h(Xi) \neq Yi \end{bmatrix}$          // update the weight value
If Di+1(j) <Ø then
train an SVM classifier to detect the intrusion.
If the objective function of a strong classifier is Yi = +1, then
The sample was deemed abnormal.
Else
The sample is normal
End if
End if
End for
End
```

## Author contribution

All authors contributed equally to this work.

## Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

## Data availability statement

The data that support the findings of this study are available on request from the corresponding author.

## Conflicts of interest

The authors declare that there is no conflict of interest.

## References

**[1]** N. Ochieng, W.Mwangi, I.Ateya, Optimizing computer worm detection using ensembles, Secur. Commun. Netw., 2019 (2019)10. https://doi.org/10.1155/2019/4656480

**[2]** V. Tasrilet al., Threats of computer system and its prevention, Int. J. Sci. Res. Sci. Technol., 3 (2017) 448-451.

**[3]** S.H. Hashem, I. A. Abdulmunem, A proposal to detect computer worms (malicious codes) using data mining classification algorithms, Eng. Technol. J., 31 (2013) 142- 155 .

**[4]** S. O. Al-Memory, H.Zhang, A. R. Abbas, IDS alarms reduction using data mining, 2008 IEEE International Joint Conference on Neural Networks, IEEE World Congress , Comput. Intell., (2008) 3564-3570. https://doi.org/10.1109/IJCNN.2008.4634307

**[5]** S. Hashim, Intrusion detection system based on data mining techniques to reduce false alarm rate, Eng. Technol. J., 36 (2018) 110-120. https://doi.org/10.30684/etj.36.2B.3

**[6]** S. Hashim, Proposed Hybrid Classifier to Improve Network Intrusion Detection System using Data Mining Techniques. Eng. Technol. J., 38 (2020) 6-14. https://doi.org/10.30684/etj.v38i1B.149

**[7]** S. Hashem, Efficiency of Svm and Pca to enhance intrusion detection system, J. Asian Sci. Res., 3 (2013) 381–395.

**[8]** S. Hashem, Enhance Network Intrusion Detection System by Exploiting BR Algorithm as an Optimal Feature Selection, Source Title: Handbook of Research on Threat Detection and Countermeasures in Network Security (2015) 16 https://doi.org/10.4018/978-1-4666-6583-5.ch002

**[9]** S. K. Majeed, S.H. Hashem, I.K. Gbashi, Propose hmnids hybrid multilevel network intrusion detection system, Int. J. Comput. Sci. Issues, 10 (2013)200-208.

**[10]** S.K. Singh, Machine-learning based stacked ensemble model for accurate analysis of molecular dynamics simulations, J. Phys. Chem., A, 123 (2019) 5190-5198.

**[11]** R.K.S.Gautam, E. A. Doegar, An ensemble approach for intrusion detection system using machine learning algorithms, Int. Conf. Cloud Comp. Data Sci. Eng., (2018) 14-15. https://doi.org/10.1109/CONFLUENCE.2018.8442693

**[12]** Y. Zhou, Building an efficient intrusion detection system based on feature selection and ensemble classifier, Comp.Netw. 174 (2020) 107247. https://doi.org/10.1016/j.comnet.2020.107247

**[13]** D. Jing, H.B. Chen, SVM based network intrusion detection for the UNSW-NB15 dataset, IEEE Int.Conf. Chong. China., (2019) 1-4. https://doi.org/10.1109/ASICON47005.2019.8983598

**[14]** H. N. Thanh, T. V. Lang, Evaluating Effectiveness of Ensemble Classifiers When Detecting Fuzzers Attacks on The Unsw-Nb15 Dataset, J. Comput. Sci .Cybernetics, 36 (2020)173-185.https://doi.org/10.15625/1813-9663/36/2/14786

**[15]** P. Y.Taser, Application of Bagging and Boosting Approaches Using Decision Tree-Based Algorithms in Diabetes Risk Prediction, Proc., 74 (2021) 6. https://doi.org/10.3390/proceedings2021074006

**[16]** S. Hamori , Ensemble learning or deep learning? Application to default risk analysis, J. Risk. Financ. Manag., 11 (2018) 12. https://doi.org/10.3390/jrfm11010012

**[17]** A.Wosiak, D.Zakrzewska, Integrating correlation-based feature selection and clustering for improved cardiovascular disease diagnosis, Cplxy., 2018 (2018)11. https://doi.org/10.1155/2018/2520706

**[18]** L. Ali. Reliable Parkinson's disease detection by analyzing handwritten drawings: Construction of an unbiased cascaded learning system based on feature selection and adaptive boosting model, IEEE Access., 7 (2019) 116480-116489. https://doi.org/10.1109/ACCESS.2019.2932037

**[19]** N.K. Korada , N. S.P. Kumar,Y. V. N. H. Deekshitulu, Implementation of naïve Bayesian classifier and ada-boost algorithm using maize expert system, J. Inf. Sci. Tech., 2 (2012) 13.

**[20]** A. Alkan, M. Günay, Identification of EMG signals using discriminant analysis and SVM classifier, Expert. Syst. Appl., 39 (2012) 44-47. http://dx.doi.org/10.1016/j.eswa.2011.06.043

**[21]** N. Moustafa , J. Slay, unsw-nb15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set) , Military Commun. Inf. Syst. Conf., (2015) 1-6. https://doi.org/10.1109/MilCIS.2015.7348942

**[22]** M. K.Hooshmand, Using Ensemble Learning Approach To Identify Rare Cyber-Attacks In Network Traffic Data, Int. Conf. Comput. Sci. Inf. Syst., (2020) 141-146. https://doi.org/10.1109/ICACSIS51025.2020.9263111