University of Baghdad
College of Engineering
JOURNAL OF ENGINEERING

Journal of Engineering

journal homepage: www.jcoeng.edu.iq

Volume 30 Number 7 July 2024

Efficient Intrusion Detection Through the Fusion of AI Algorithms and Feature Selection Methods

Marwa Mohammad Obaid[1, *], Muna Hadi Saleh[2]

Department of Electrical Engineering, College of Engineering, University of Baghdad, Baghdad, Iraq
marwa.obaid2302m@coeng.uobaghdad.edu.iq[1], dr.muna.h@coeng.uobaghdad.edu.iq[2]

ABSTRACT

With the proliferation of both Internet access and data traffic, recent breaches have brought into sharp focus the need for Network Intrusion Detection Systems (NIDS) to protect networks from more complex cyberattacks. To differentiate between normal network processes and possible attacks, Intrusion Detection Systems (IDS) often employ pattern recognition and data mining techniques. Network and host system intrusions, assaults, and policy violations can be automatically detected and classified by an Intrusion Detection System (IDS). Using Python Scikit-Learn the results of this study show that Machine Learning (ML) techniques like Decision Tree (DT), Naïve Bayes (NB), and K-Nearest Neighbor (KNN) can enhance the effectiveness of an Intrusion Detection System (IDS). Success is measured by a variety of metrics, including accuracy, precision, recall, F1-Score, and execution time. Applying feature selection approaches such as Analysis of Variance (ANOVA), Mutual Information (MI), and Chi-Square (Ch-2) reduced execution time, increased detection efficiency and accuracy, and boosted overall performance. All classifiers achieve the greatest performance with 99.99% accuracy and the shortest computation time of 0.0089 seconds while using ANOVA with 10% of features.

Keywords: Intrusion Detection System (IDS), Machine learning, Naïve bayes, K-Nearest Neighbor (KNN), Decision tree, Feature selection.

*Corresponding author
Peer review under the responsibility of University of Baghdad.
https://doi.org/10.31026/j.eng.2024.07.11

Article received: 29/01/2024
Article accepted: 24/05/2024
Article published: 01/07/2024

كشف التسلل من خلال دمج خوارزميات الذكاء الاصطناعي وطرق اختيار الميزات

مروة محمد عبيد *، منى هادي صالح

قسم الهندسة الكهربائية، كلية الهندسة، جامعة بغداد، بغداد، العراق

الخلاصة

مع انتشار الوصول إلى الإنترنت وحركة البيانات، سلطت الخروقات الأخيرة الضوء على الحاجة إلى أنظمة كشف التسلل إلى الشبكة (NIDS) في حماية الشبكات من الهجمات السيبرانية الأكثر تعقيدًا. من أجل التمييز بين عمليات الشبكة العادية والهجمات المحتملة، غالبًا ما تستخدم أنظمة كشف التسلل (IDS) تقنيات التعرف على الأنماط واستخراج البيانات. يمكن اكتشاف عمليات التطفل على الشبكة ونظام المضيف والاعتداءات وانتهاكات السياسة وتصنيفها تلقائيًا بواسطة نظام كشف التسلل (IDS). تظهر نتائج هذه الدراسة أن تقنيات تعلم الآلة مثل Decision Tree (DT)، و Naïve Bayes (NB)، و K-Nearest Neighbor (KNN) يمكن أن تعزز فعالية نظام كشف التسلل. يتم قياس النجاح من خلال مجموعة متنوعة من المقاييس، بما في ذلك الدقة والاستدعاء والدقة ودرجة F1 ووقت التنفيذ. أدى تطبيق أساليب اختيار الميزات مثل تحليل التباين (ANOVA)، والمعلومات المتبادلة (MI)، ومربع كاي (Ch-2) إلى تقليل وقت التنفيذ، وزيادة كفاءة الكشف ودقته، وتعزيز الأداء العام. تحقق جميع المصنفات أفضل أداء بدقة تصل إلى 99.99% وأقصر وقت حسابي يبلغ 0.0089 مللي ثانية أثناء استخدام ANOVA مع 10% من الميزات.

الكلمات المفتاحية: نظام كشف التسلل، التعلم الآلي، Naïve bayes، K-Nearest neighbor، Decision tree، اختيار الميزات.

1. INTRODUCTION

Current solutions, despite significant progress in the field of network security, remain inadequate to completely safeguard networks of computers against hostile attacks. Traditional security measures like firewalls, user authentication, and data encryption are inadequate for safeguarding network security against evolving infiltration techniques. Preventative techniques like Intrusion Detection systems (IDSs) are being created to improve the safety of systems (**Tapiador et al., 2013**).

An IDS is designed to identify threats based on monitoring network traffic. It is essential to monitor and review regular computer operations to detect intrusions and weaknesses in security. IDSs are often categorized as either signature-based (misuse-based) detection systems or anomaly-based detection systems. One of the major issues with IDSs is the production of the production of too many false signals, which generate more data for the system than it can handle. The UNSW-NB15 is a popular dataset in the security breach detection field (**Zeeshan *et al.*, 2021)**. The scale of a massive dataset can slow down the classification process and even compromise the accuracy of a classifier due to a limited memory size. In addition to that, big data is full of duplicates and noisy information which can be a serious problem to knowledge discovery and data modeling. The research conducted demonstrates that ML methods such as NB, KNN, and DT can increase the effectiveness of IDSs. The performance of IDS can be improved by including a preprocessing

stage in it. Feature selection is one of the preprocessing strategies that works efficiently in solving IDS problems by selecting important features and at the same time removing any redundant features **(Ambusaidi *et al.*, 2016)**.

The research analyzed previous studies to identify methods that had been used to enhance the effectiveness and accuracy of the IDS. The research **(Gu and Lu, 2021)** presented a system for intrusion detection using Support Vector Machines (SVMs) with NB features embedded. To train an SVM classifier, the framework first modified the original features using the NB feature transformation method, which produced new, high-quality data. The suggested detection method has been tested on various intrusion detection datasets and has proven to be very accurate, with accuracies of 93.75% on the UNSW-NB15 dataset, 98.92% on the CICIDS2017 dataset, 99.35% on the NSL-KDD dataset, and 98.58% on the Kyoto 2006+ dataset. In this work **(Arik and Çavdaroğlu, 2024)**, the authors introduced ROGONG-IDS, which is divided into three parts: data primer, the classification imbalance process, and classification solution. In the data collection section, hot coding, labeling, feature selection, and data normalization are addressed. A pair revising strategy of the unbalanced class challenge in which nearmiss-1 under-sampling complemented SMOTE over-sampling approaches are offered. Data size increases necessitate more computer power and time to address imbalanced class issues. Thus, ROGONG-IDS demonstrated sustained performance over the UNSW-NB15 dataset, reaching a 97.30% malware detection rate and over 97.65% of F1- score.

In **(More *et al.*, 2024),** the UNSW-NB15 network traffic dataset was used to improve IDS through the use of Logistic Regression (LR), SVM, DT, and Random Forest (RF) techniques. LR, DT, and linear SVM were among the ML models that underwent hyperparameter tuning to improve their accuracy. When it came to detecting cyber-attacks, the RF model performed the best, with an accuracy rate of 98.63% and an F1-Score of 97.80%. Reproducibility may be affected since the computational resources and hardware parameters utilized by the ML models are not specified**.** In **(Kocher and Kumar, 2021)**, researchers used the UNSW-NB15 dataset for training ML classifiers such as KNN, Stochastic Gradient Descent (SGD), RF, LR, and NB. They used the Ch-2 filter-based feature selection method to eliminate unwanted features from the dataset. The RF predictor outperforms prior algorithms in terms of accuracy of 99.57, Mean Squared Error (MSE) of 0.004, and a true-positive rate of 0.997. The NB classifier, on the other hand, had the highest MSE of 0.234 and the smallest accuracy of 76.59 among the tested classifiers. The research has certain limitations, such as its reliance on the UNSW-NB15 dataset for training ML classifiers. As a result, the findings may not apply to other datasets. In **(Hussein, 2022),** the study employed supervised ML techniques including KNN, SVM, NB, DT, RF, SGD, GB, and AB classifiers for intrusion detection. Evaluation of performance was conducted using the confusion matrix. Information Gain, Pearson, and F-test methods for selecting features were compared with models using all three. The KDD99 dataset was utilized to assess the effectiveness of models based on machine learning. The RF classifier achieved the highest accuracy of 99.96% with a margin of error of 0.038%. Problems and difficulties with using the KDD99 dataset in IDSs are not addressed in the article.

In **(Fuat, 2023)**, the use of ML algorithms as a viable solution for an IDS is mentioned in the study. For assault detection and categorization, the Multi-Layer Perception (MLP) neural network architecture, Long-Short-Term Memory (LSTM), LR, KNN, DT, and RF were utilized. Using the UNSW-NB15 and NSL-KDD datasets. The study does not go into depth about the ML and DL algorithms used. The UNSW-NB15 dataset had two-class and multi-class

classification accuracies of 98.6% and 98.3%, respectively. The accuracy ratings in the NSL-KDD dataset were 93.4% and 97.8%, respectively.
This work has the following contributions:

- Using ML algorithms NB, KNN, and DT to increase the accuracy of IDS.
- Unable to directly apply feature selection due to numerous instances of duplicate data in the UNSW-NB15 dataset. Label encoding and normalization helped us examine the intrusion detection dataset by bringing all of the features' value ranges together and removing any bias.
- To eliminate features that detract from model performance and prolong execution time. Therefore, use three feature selection methods: ANOVA, MI, and Ch-2.

2. INTRUSION DETECTION SYSTEM

The IDS tracks and monitors computer system events. It discovers security flaws through the use of event-based approaches and security data. More computers and clients are connected to data and computer networks as Internet-based infrastructure grows. These gadgets cater to both public and private online consumers as well as organizations. The ever-increasing variety of assaults needs the use of an efficient intrusion detection system that recognizes recorded forms as well as learns to recognize novel forms **(Ahmad et al., 2022)**. IDS can be categorized into specific groups based on the places where data analysis takes place **(Alkanhel *et al.*, 2023).**

2.1 According to the Detection Location

The IDSs are categorized as follows based on the location of the detection **(Pradhan et al., 2020)**:

1. Host Intrusion Detection System (HIDS): this sort of IDS may be installed on network devices or workstations. This IDS can prevent attacks on a single device but not the entire network.
2. Network Intrusion Detection System (NIDS): this IDS can detect and categorize all network traffic from all devices in a protected network, as shown in **Fig. 1.**

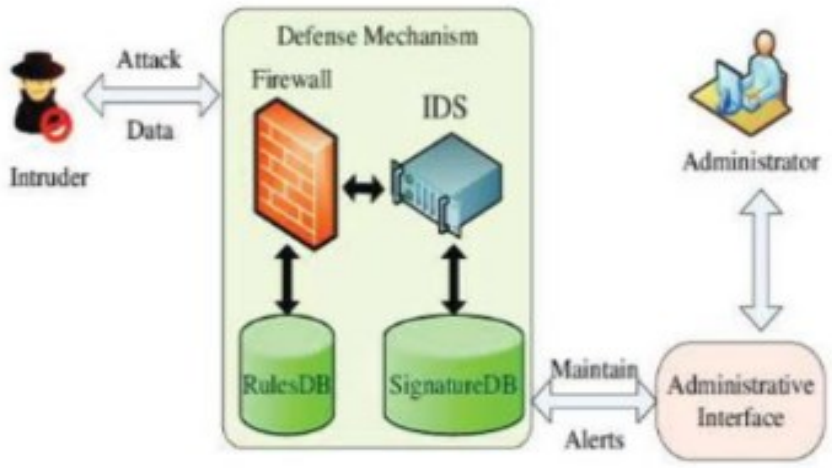


Figure 1. Intrusion Detection System **(Gupta and Agrawal, 2020)**

3. Hybrid Intrusion Detection System (HYIDS): hybrid IDS adds the capability to monitor network traffic entering and leaving a particular host. A comprehensive picture of the network is produced by combining host agency data with network information, creating a combination of HIDS and NIDS.

2.2 According to The Detection Method

The classification of intrusion detection systems is based on the type of detection mechanism employed:

1. Signature-based IDS: approaches identify abnormalities by comparing preset attack signatures**(Hwang *et al.*, 2007)** . The key benefits of these approaches are their ease of use and low false positive rates; nevertheless, they cannot identify novel mimicking assaults **(Kabir et al., 2018)**.
2. Anomaly-based detection: techniques are based on the presumption that the intruder's action departs beyond regular network behavior **(Kabir et al., 2018)**. These technologies monitor the typical traffic of a network and identify any abnormal behavior as malicious activity. This method can identify both unidentified and identified attacks. This method's primary drawback is its exceptionally high rate of false alarms **(Pietraszek, 2004)**.

3. DATASET UNSW-NB15

The UNSW-NB15 intrusion detection dataset was developed in 2015 at the cyber range Laboratory of the Australian Centre for Cyber Security at the University of New South Wales (UNSW) using the IXIA perfect storm tool in a simulated environment **(Moustafa and Slay, 2015b; Moustafa and Slay, 2016)**. Deficits in current datasets like KDD-98, KDD-CUP99, and NSL-KDD prompted the development of this dataset, which depicts a more complex and modern threat environment. A new dataset contains synthetic contemporary assaults that mirror real-world modern regular behavior. There are nine separate contemporary attack categories in the UNSW-NB15 dataset (backdoors, fuzzers, reconnaissance, exploits, worms, DoS, analysis, generic, and shellcode) **(Moustafa and Slay, 2015a)** and one normal kind, compared to 14 in the KDD'99 dataset. There are other 49 characteristics, like the class label, and a large range of actual everyday activities. The entire dataset is divided into two groups testing and training. This dataset can be accessed via the link https://research.unsw.edu.au/projects/unsw-nb15-dataset.

4. METHODOLOGY

Fig. 2 represents the proposed system. Starting with step 1 of the input process, which involves label encoding and data normalization, as a result of their initial variation in amounts, it is essential to standardize all feature values onto a uniform scale. Before passing them on to the classification approach, the dataset's numeric and textual features into a common format. Feature reduction is the second step of the method. Three feature selection models are used to pass pre-processed features, and each model returns a distinct collection of features. The features that make it into the classification model are chosen from among these three sets based on the results of the selection models. Three feature selection models ANOVA, MI, and Ch-2 were employed in this investigation. After the second phase, the original dataset is reduced to a single feature subset. Classification is the main focus of step

3, and ML is employed for this objective. The ML models are fed the final subset of pre-processed features. In the algorithm (1) steps of the method proposed in this work.

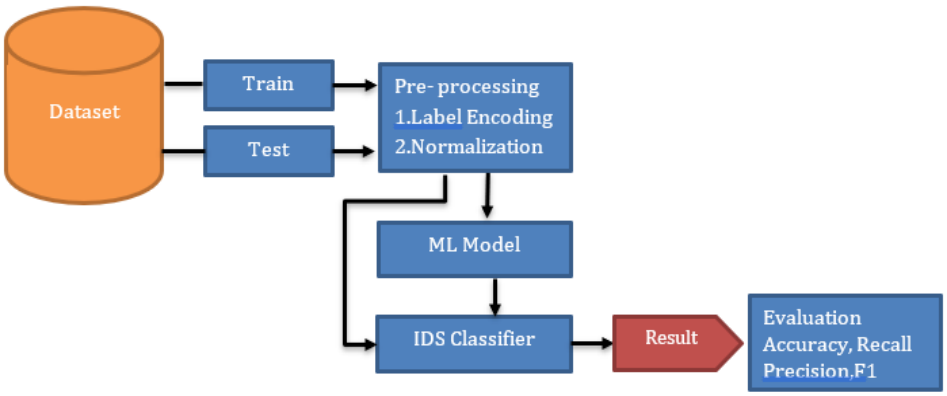


Figure 2. Architecture for the proposed model without feature selection

Algorithm 1: ML-based Framework for feature extraction and classification for intrusion detection in network
Input: UNSW-NB15 dataset **Output:** Accuracy, Precision, Recall, F1-Score, Time **Begin:** Initialization: Data = features of UNSW-NB15 dataset, nfeatures = Numeric features, tfeatures = Textual features, ds_f1= features from ANOVA, ds_f2 = features from MI, ds_f3 = features from Ch-2. **Step1:** Data preprocessing **Step 2:** Features selection ds_f1 = ANOVA ds_f2 = MI ds_f3 = Ch-2 End step **Step 3:** Classification by using (NB, KNN, and DT) The model is trained and teased on UNSW-NB15 dataset binary classification. **End step** **Return the classification result**

4.1 Workflow without Feature Selection Method

The strategy entails developing an intrusion detection system without reliance on feature selection approaches. This strategy stresses employing full datasets and incorporating all accessible information in the analysis procedure.

4.1.1 Pre-Processing Dataset

In this section, the exploration strategy works without using feature selection.

1. Dataset dividing: The dataset has been divided into a training dataset and a testing dataset with a ratio of 70% for training and 30% for testing.

2. Labeling encoding**:** We began by reading the data found in the training and testing data files, which will be used to train and test the proposed classifier. The characteristics were then manually encoded by converting the labels from text to letters and numbers, allowing them to be processed. (f1, f2..., f42) was the order in which the characteristics were encoded.
3. Normalizer: The function takes an array as an argument and standardizes its values to a range from 0 to 1. The output array is generated to have the same dimensions as the input to counteract the impact of feature scaling during model training. Therefore, our system can reach optimal weights and improve its accuracy**.** According to Eq. (1), the updated value is determined as the difference between the minimum value and the scale size **(Farhana et al., 2020).**

$$X'_i = \frac{X_i - \min(X_i)}{\max(X_i) - \min(X_i)} \quad (1)$$

Where X_i represents the ith feature vector, min (X_i) calculates the vector's minimal value and max (X_i) calculates the vector's maximum value

4.2 Machine Learning (ML) Classifiers

The use of supervised machine learning includes the goal of classification. Intrusion detection and categorization are made possible by the model's training. A binary classification (normal class and an abnormal class (attack)) is created from the input data. Diverse supervised ML methods are used to construct classifier models, with the following foundations:

1. Probability approach: NB
2. Distance approaches: KNN
3. Rule approaches: DT

4.2.1 Naïve Bayes (NB) Algorithm

Constructed using the principle of Bayes, this supervised ML classification is famous for being easy to use. Bayes rules are used to calculate the posterior probability P(c, x) in the following way **(Kachavimath et al., 2020)**:

$$P(c, x) = \frac{P(x,c)\, P(c)}{P(x)} \quad (2)$$

Where P(c, x): denotes the probability of the posterior class.
P(*c)*: denotes the preceding class's probability.
P(x*, c)*: is the predictor class's provided probability.
P(*x)*: is the prior predictor's probability.

4.2.2 K-Nearest Neighbors (KNN) Algorithm

In ML the KNN is utilized for issues related to regression and classification **(Larose and Larose, 2014)**. A distance function is used to implement KNN. When the number of classifiers K=1, the instance is allocated to the class of its nearest neighbor.

4.2.3 Decision Tree (DT) Algorithm

DT is a rule-based classifier that ranks data according to its attribute values. Every node in the tree represents an input characteristic, and each branch reflects the value of that feature. The method of classification begins at the root level based on feature values and divides the data into several measures utilized in the sample identification, such as entropy and information gain**(Mousavi et al., 2022).** Entropy fluctuates between zero and one. Its value is optimal at 0 and deteriorates at 1, indicating that the closer it is to 0, the better. Information gain is the reciprocal of entropy, with higher values indicating better performance **(Charbuty and Abdulazeez, 2021)**.

4.3 Workflow with Feature Selection Method

To remove irrelevant and redundant data from the dataset, a feature selection technique is required. Feature selection is a strategy for picking a subset of relevant features while preserving the presentation. The presence of superfluous features in the intrusion dataset frequently impeded accurate detection. Some reasons were investigated as to why it might be necessary to limit the features. Here, it added the feature selection stage before putting the dataset into the classification algorithms, as presented in **Fig. 3**. The strategy for operation includes integrating feature selection techniques into the IDS. In this work, we have used three methods of feature selection: ANOVA, MI, and Ch-2.

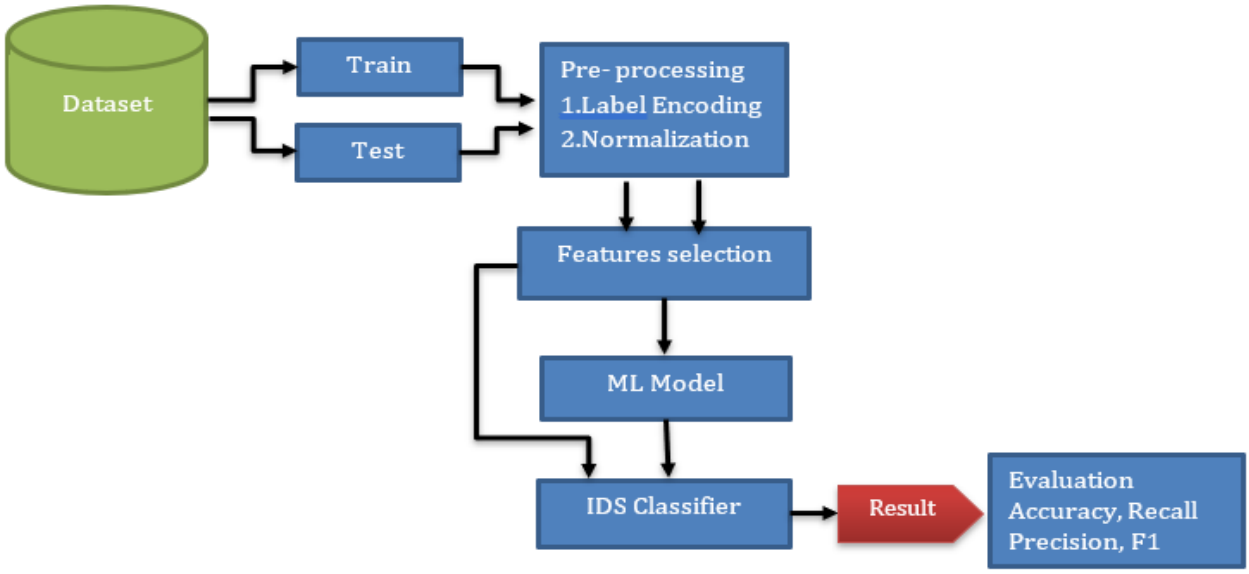


Figure 3. Architecture for the proposed model with feature selection

4.3.1 Feature Selection Methods

It is one of the most significant steps in data preparation in ML **(Pathak and Pathak, 2020)**. It preserves only the important qualities and discards the rest. Any characteristic that does not contribute to forecasting the target value is discarded.

4.3.1.1 *Analysis of Variance (ANOVA)*

Ronald Fisher developed ANOV as a statistical approach for analyzing data variations within the mean. The ANOVA test is used in regression studies to measure the level of significance between independent variables. The ANOVA approach compares many groups at the same time to identify the relationship between them **(Siraj et al., 2022).** The ANOVA implementing result, known as the f-statistic or f-ratio, may be used to examine variability across and within the sample. The ANOVA may be computed using Eq. (3):

$$F = \frac{MST}{MSE} \tag{3}$$

where F represents the ANOVA coefficient, MST is the Mean Sum of Squares value, and MSE illustrates the Mean Sum of Squares Error value.

4.3.1.2 *Mutual Information (MI)*

Information theory was first established to quantify the quantity of information conveyed in data **(Song et al., 2014)**. In this theory, entropy is a crucial measure of information. It can properly assess the uncertainty of random variables and scale the quantity of information they exchange. Let X be a random variable with discrete values; its entropy can be expressed as Eq. (4):

$$H(X) = -\sum_{x \in X} p(x) \log p(x) \tag{4}$$

p(x)=Pr (X=x) is the probability density function of X, whereas H (·) represents entropy. It is important to note that entropy is determined by the random variable's probability distribution.

4.3.1.3 *Chi-square (Ch-2)*

The Ch statistic measures the degree of independence between the feature a_i and the class label y_j by comparing it to the Ch-2 distribution with a single level of free. Thus, the Chi-Square statistic has been defined as follows in Eq. (5) **(Krishnaveni *et al.*, 2021)**:

$$X^2(a_i, y_{j)} = \sum_{i=1}^{k} \left[\frac{(O-E)^2}{E}\right] \tag{5}$$

where X^2 is represents Chi-Square, O is a category's observed frequency, E is the expected frequency, and k is the number of observations in the sample (or categories in the dataset).

5. RESULTS AND DISCUSSION

5.1 Setting up the Hardware and the Environment

Evaluating the proposed system's behavior, performance, and operation relies heavily on the implementation environment description. Here will offer the tables that will show how the suggested system will be implemented and for this proposal, used a DELL laptop computer loaded with Windows 10 Pro, version 22H2. A Core (TM)i7 processor and 8 GB of RAM make up the laptop's components. Our experimental environment was built on top of Python 3.6. TensorFlow, pandas, numpy, matplotlib, and other programs. These libraries provided functionalities for data processing, feature selection, and visualization.

5.2 Evaluation Specifications

The ground truth value is necessary in assessment to estimate the various statistical measures. In the instance of binary classification, the ground truth is a series of connection data labeled either normal or attack. The following words have been utilized to assess the quality of categorization models**:**

- True Positive 'TP': the number of connection records that were accurately classified as normal.
- True Negative 'TN': the number of connection records that were accurately classified as an attack**.**
- False Positive 'FP': the number of normal connection records that were incorrectly classed as attack connection records**.**
- False Negative 'FN': The number of attack connection records that had been wrongly determined to be normal.

The following most widely used assessment measures are examined based on the terms above **(Ali and Dawood, 2023)**.

1. Accuracy: It determines the ratio of correctly detected connection records to the entire test dataset. ML algorithm performs better when it achieves a higher accuracy score within the range of 0 to 1. The accuracy of the test dataset with balanced classes is defined in Eq. (6):

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{6}$$

2. Precision: It determines the ratio of correctly detected attack connection records to all identified attack connection records. ML model is considered superior when its Precision value is higher (ranging from 0 to 1). Below is a definition of precision in Eq. (7):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

3. F1-Score: F1-Measure is another name for F1-Score. Precision and recall are harmonic means. The ML model is better if the F1-score is greater (F1-score [0, 1]). The following is the definition of F1-Score in Eq. (8).

$$\text{F1-score} = 2 * \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

4. Recall: It estimates the ratio of the correctly classified attack connection records to the total number of attack connection records. If the recall is higher, the ML model is better (recall ∈ [0, 1]). Recall is defined as Eq. (9):

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

5.3 Test Performance without Feature Selection

Through the framework illustrated in **Fig. 2**, we achieved results for the UNSW-NB15 dataset binary classification with normal and attack problems utilizing three classifiers: NB, KNN, and DT. The binary classifier's performance is evaluated using accuracy, precision, recall, F1-Score, and time metrics. The results in **Table 1** show that all classifiers achieved high detection accuracy without feature selection, but the computational time of the KNN method is relatively high.

Table 1. Performance measures classifiers without feature selection

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Time(sec)
NB	0.9634	0.97	0.97	0.97	0.0799
KNN	0.9520	0.96	0.96	0.96	35.627
DT	0.974	0.975	0.97	0.97	0.1795

In **Fig. 4**, the performance measures (accuracy, precision, recall, and F1-Score) were good for the proposed machine learning algorithms.

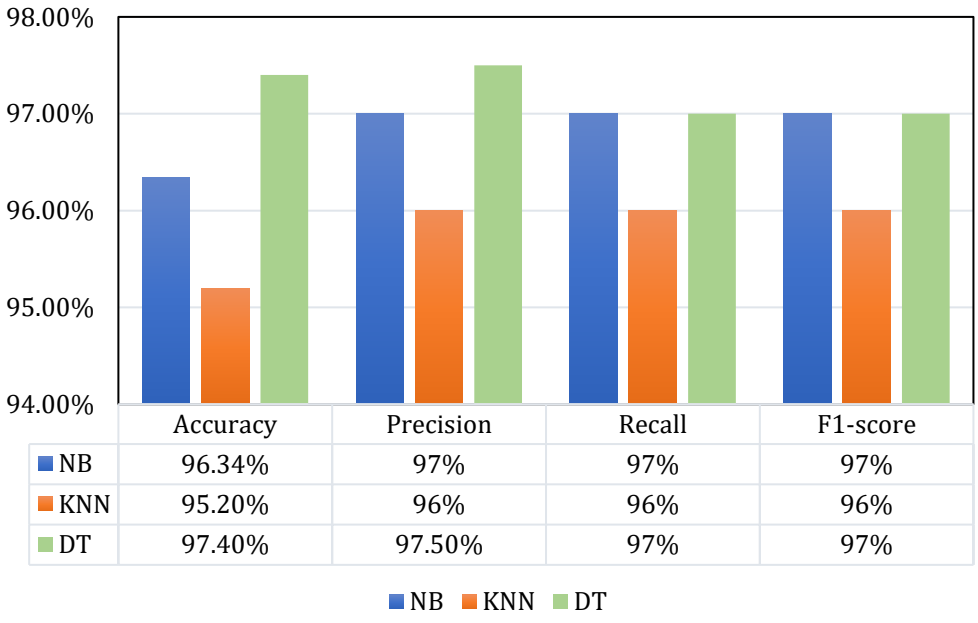


Figure 4. Performance measures classifiers without feature selection

5.4 Test Performance with Feature Selection

This section contains the different outcomes collected from the experimentation procedure shown in **Fig. 3**. The results were obtained using the UNSW-NB15 dataset for binary classification with normal and attack problems. As described in the system above, feature selection was carried out using ANOVA, MI, and Ch-2. Pick different proportions of all features, such as 10%, 30%, and 50%. The following are some of the reasons why various dimensional feature selection reduction approaches have been used **(Mebawondu *et al.*, 2020):**

- Reduces overfitting from occurring.
- Get a basic model for a classifier that can effectively generalize data.
- Simplifying calculations, reducing memory storage requirements, and accelerating the training time.
- Enhancing the rates of false alarms and learner performance.

The following **Table 2** displays the outcomes of the suggested method ANOVA when utilizing the binary classification of the UNSW-NP15 dataset with feature selections of 50% and 30% from all features. Using 50%, 30%, and 10% of the features when utilizing feature selection methods gave better results in terms of accuracy and time than using all the features, as shown in the figures and tables below.

Table 2. Performance measures using ANOVA with 50% and 30% from features

Percentage	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Time (sec)
50%	NB	0.98	0.99	0.98	0.98	0.0468
	KNN	0.99	0.99	0.99	0.99	32.062
	DT	0.9999	0.9999	0.99	0.99	0.0469
30%	NB	0.98	0.99	0.99	0.99	0.0299
	KNN	0.9999	0.9999	0.9999	0.9999	4.0910
	DT	0.9999	0.9999	0.9999	0.9999	0.0299

Table 3 shows that using ANOVA with a 10% feature ratio results in a significant drop in KNN execution time, with all algorithms achieving detection accuracy, precision, recall, and F1-Score of 99.99%.

Table 3. Performance measure using ANOVA with 10% from features

Classifier	Time (sec)
NB	0.0139
KNN	3.5066
DT	0.0089

In **Table 4**, it can be seen that using the feature selection method MI with a 50%, 30%, and 10% feature ratio results on the same dataset in a considerable reduction in KNN execution time, which exceeds the drop reported in ANOVA. All algorithms have detection accuracy, precision, recall, and F1-Score of 99.99%.

Table 4. Performance measure using MI with 50%, 30%, and 10% from features

Percentage	Classifier	Time (sec)
	NB	0.0458
50%	KNN	32.8292
	DT	0.0658
	NB	0.0312
30%	KNN	2.5463
	DT	0.0312
	NB	0.0249
10%	KNN	1.1252
	DT	0.0144

There are multiple reasons why integrating ML algorithms with feature selection techniques in IDS design can reduce computational complexity:

- Enhanced Efficiency: The system can learn and predict more efficiently by focusing on the features that hold the most relevant information.
- Dimensionality Reduction: By assisting in the reduction of the dataset's dimensions, feature selection techniques can cut down on the amount of time needed for analysis and learning.
- Cost Reduction: Time and effort can be saved by minimizing operational and computational expenses by limiting the number of features used

Table 5 gives the results of using the Ch-2 method. Note that the decrease in detection accuracy for the NB, KNN, and DT methods, is accompanied by long execution times for the KNN algorithm with features of 50%, 30%, and 10%, but the computational time for the rest of the algorithms was small.

Table 5. performance measure using Ch-2 with 50%, 30%, and 10% from features

Percentage	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Time (sec)
50%	NB	0.98	0.99	0.99	0.99	0.0377
	KNN	0.997	0.9999	0.9999	0.9999	58.7074
	DT	0.9999	0.9999	0.9999	0.9999	0.2182
30%	NB	0.6902	0.82	0.69	0.72	0.0298
	KNN	0.8448	0.84	0.84	0.84	3.0882
	DT	0.9205	0.92	0.92	0.92	0.3694
10%	NB	0.6846	0.84	0.68	0.72	0.0081
	KNN	0.81	0.81	0.81	0.81	1.2498
	DT	0.889	0.89	0.89	0.89	0.2597

Figs. 5 to **7** show the Ch-2 results for the feature selection method and show the decrease in accuracy, precision, recall, and F1-score.

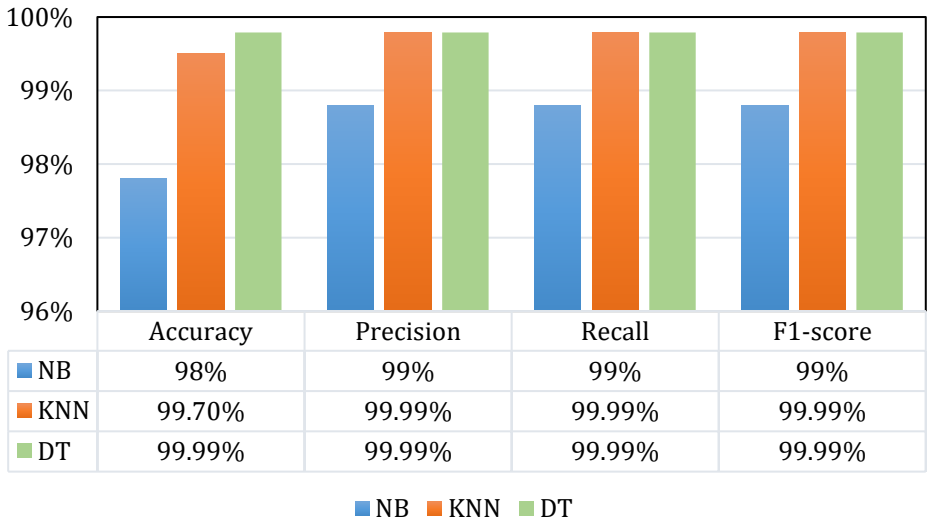
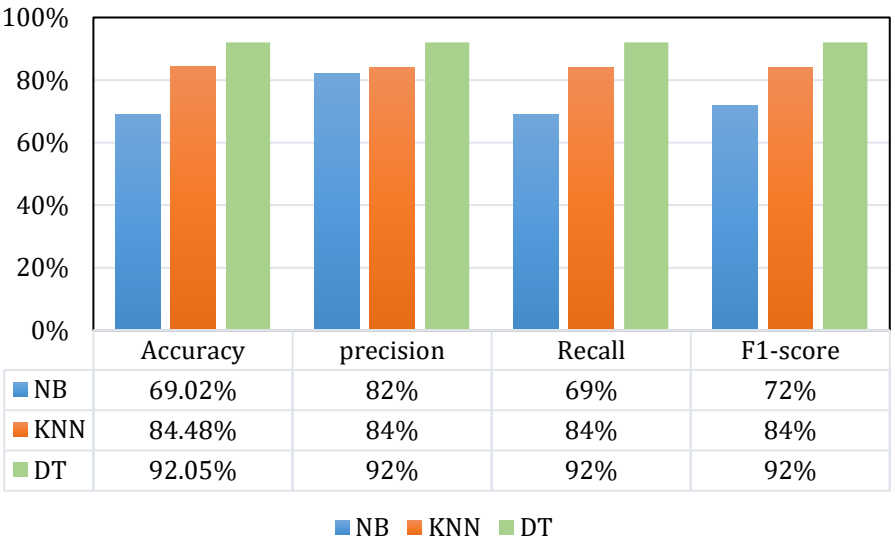
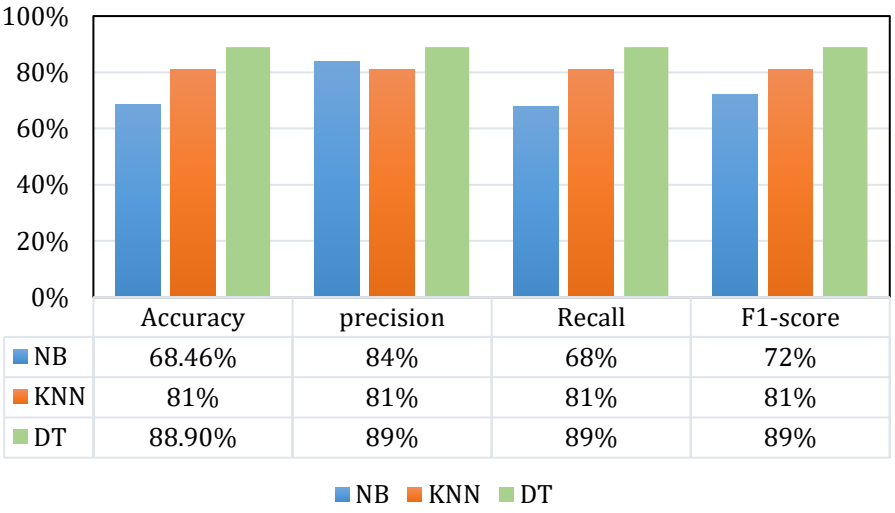


Figure 5. Performance measure using Ch-2 with 50% of features.



	Accuracy	precision	Recall	F1-score
NB	69.02%	82%	69%	72%
KNN	84.48%	84%	84%	84%
DT	92.05%	92%	92%	92%

Figure 6. Performance measure using Ch-2 with 30% of features.



	Accuracy	precision	Recall	F1-score
NB	68.46%	84%	68%	72%
KNN	81%	81%	81%	81%
DT	88.90%	89%	89%	89%

Figure 7. Performance measure using Ch-2 with 10% of features.

For intrusion detection, a binary classification procedure is employed, using multiple classifier models based on an ML-supervised algorithm. Many assessment measures are used to determine each model's performance, including accuracy, precision, recall, F1 score, and time. The experiment findings are presented in tables for several models based on different rules, and the key points are given below:

- Various dimension selection procedures were employed to decrease the input feature space. This involved using 10%, 30%, and 50% of all features from the UNSW-NB15 dataset. Selected characteristics had a notable effect on the performance of the model while minimizing time and memory usage. MI yields somewhat superior performance outcomes compared to the other approaches employed in this investigation.

- The nature of Chi-Square (Ch-2) feature selection may be explained by the decrease in intrusion detection accuracy and the rise in execution time. Its application may not be compatible with the features of the data or the algorithmic specifications.

- **Table 6** and **Fig. 8** illustrate a comparison between the results of this work and the results of **(More et al., 2024; Arik and Çavdaroğlu, 2024; Khan et al., 2020; Fuat, 2023**), emphasizing its contributions. The second column (The proposed work) displays the higher attained values achieved. It was found that our proposed system achieved high accuracy compared to previous works, and the computational time achieved was short.

Table 6. Comparison with some previous work

Type	The proposed work	(More *et al.*, 2024)	(Arik and Çavdaroğlu, 2024)	(Khan et al., 2020)	(Fuat, 2023)
Accuracy	99.99%	98.63%	97.30%	98.6%	98.6%
Precision	99.99%	--	--	--	97%
Recall	99.99%	--	--	--	98%
F1-Score	99.99%	97.80%	97.65%	98%	98.%
Time (sec)	0.0089	--	--	--	15.1

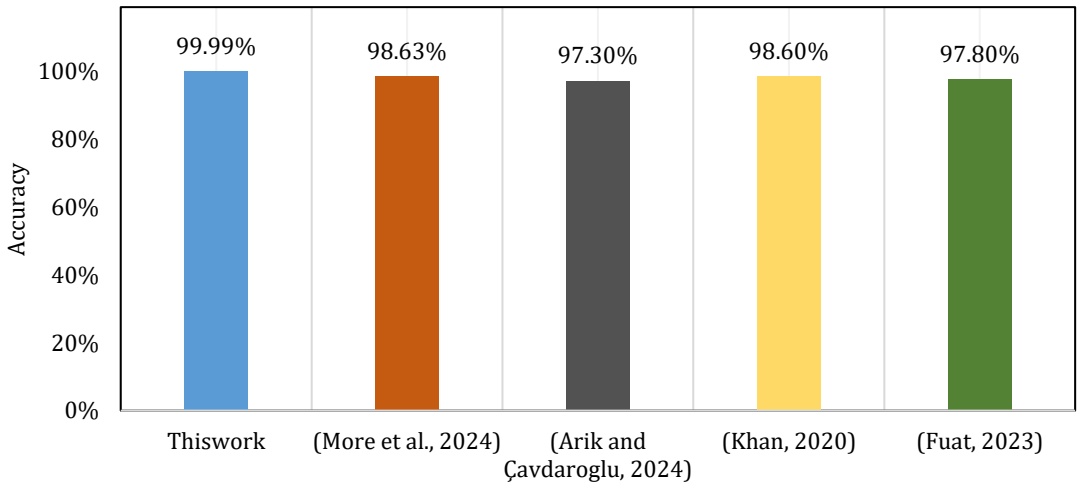


Figure 8. Comparison of present work with previous work

6. CONCLUSIONS

With the developments in Intrusion and attack activities, classification based on ML has become critical, which has led to the development of this work to improve the performance of IDSs.

1. The results of this work show that when the features are reduced using feature selection methods, the detection accuracy of the system increases with a decrease in the computational time of all the ML algorithms used.
2. When using the MI feature selection method, the computational time of the KNN algorithm is reduced from 32.8292 msec to 1.1252 msec with a high accuracy of 99.99%.
3. The use of the Ch-2 approach significantly reduces the accuracy of detection of all methods with a 30% and 10% advantage selection. It reaches 68.46% with NB, 81% with KN, and 88.9% with DT.

The future works suggested for this work are:

1. The specialized DL design allows for excellent discovery accuracy across multiple feature selection approaches while seamlessly adapting to new datasets, all without the need for a big data infrastructure.
2. The dataset can also be divided into training and test datasets by (80%: 20%).
3. Additional data sets can be used to build a model that has great flexibility against any possible aggressions in a real-time situation.

NOMENCLATURE

Symbol	Description	Symbol	Description
E	Expected frequency	O	Category's observed frequency
F	ANOVA coefficient	P(c, x)	Probability of posterior class
FP	False Negative	P(x, c)	Predictor classes provide probability
FN	False Negative	P(c)	Preceding class's probability
H	Entropy	P(x)	Probability density
K	Number of the sample	TP	True Positive
Max (X_i)	Maximum value	TN	True Negative
Min (X_i)	Minimal value	X_i	Feature vector
MSE	Mean Square Error	X^2	Chi-Square
MST	Mean sum of Square value		

Credit Authorship Contribution Statement

Marwa Mohammad Obaid: undertook this research as part of her M.Sc. study. This approach included Machine Learning (ML) algorithms and feature selection methods to develop an Intrusion Detection System (IDS). Marwa wrote the draft version. Muna Hadi Saleh is the supervisor and guided the research development, reviewing, and improvement of the manuscript.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

Ahmad, I. Ul Haq, Q. E., Imran, M., Alassafi, M. O., and AlGhamdi, R. A. 2022. An efficient network intrusion detection and classification system, *Mathematics*, 10(3), p. 530. Doi:10.3390/math10030530

Ali, A.A. and Dawood, F.A.A. 2023. Deep learning of diabetic retinopathy classification in fundus images, *Journal of Engineering*, 29(12), pp. 139–152. Doi:10.31026/j.eng.2023.12.09

Alkanhel, R. El-kenawy, E. S. M., Abdelhamid, A. A., Ibrahim, A., Alohali, M. A., Abotaleb, M., and Khafaga, D. S. 2023. Network intrusion detection based on feature selection and hybrid metaheuristic optimization., *Computers, Materials & Continua*, 74(2). Doi:10.32604/cmc.2023.033273

Ambusaidi, M.A. He, X., Nanda, P., and Tan, Z. 2016. Building an intrusion detection system using a filter-based feature selection algorithm, *IEEE Transactions on Computers*, 65(10), pp. 2986–2998. Doi:10.1109/TC.2016.2519914

Arik, A.O. and Çavdaroğlu, G.Ç. 2024. An intrusion detection approach based on the combination of oversampling and undersampling algorithms, *Acta Infologica*, 7(1), pp. 125–138. Doi:10.26650/acin.1222890

bhai Gupta, A.R. and Agrawal, J. 2020. A comprehensive survey on various machine learning methods used for intrusion detection systems, in *2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT)*. IEEE, pp. 282–289. Doi:10.1109/CSNT48778.2020.9115764

Charbuty, B. and Abdulazeez, A. 2021. Classification based on decision tree algorithm for machine learning, *Journal of Applied Science and Technology Trends*, 2(01), pp. 20–28. Doi:10.38094/jastt20165

Farhana, K., Rahman, M. and Ahmed, M.T. 2020. An intrusion detection system for packet and flow based networks using deep neural network approach, *International Journal of Electrical & Computer Engineering (2088-8708)*, 10(5). Doi:10.11591/ijece.v10i5.pp5514-5525

Fuat, T. 2023. Analysis of intrusion detection systems in UNSW-NB15 and NSL-KDD datasets with machine learning algorithms, *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, 12(2), pp. 465–477. Doi:10.17798/bitlisfen.1240469

Gu, J. and Lu, S. 2021. An effective intrusion detection approach using SVM with naïve Bayes feature embedding, *Computers & Security*, 103, p. 102158. Doi:10.1016/j.cose.2020.102158

Hussein, M.A. 2022. Performance analysis of different machine learning models for intrusion detection systems, *Journal of Engineering*, 28(5), pp. 61–91. Doi:10.31026/j.eng.2022.05.05

Hwang, K., Cai, M., Chen, Y., and Qin, M. 2007. Hybrid intrusion detection with weighted signature generation over anomalous internet episodes, *IEEE Transactions on dependable and secure computing*, 4(1), pp. 41–55. Doi:10.1109/TDSC.2007.9

Kabir, E., Hu, J., Wang, H., and Zhuo, G. 2018. A novel statistical technique for intrusion detection systems, *Future Generation Computer Systems*, 79, pp. 303–318. Doi:10.1016/j.future.2017.01.029

Kachavimath, A. V, Nazare, S.V. and Akki, S.S. 2020. Distributed denial of service attack detection using naïve bayes and k-nearest neighbor for network forensics, in *2020 2nd International conference on innovative mechanisms for industry applications (ICIMIA)*. IEEE, pp. 711–717. Doi:10.1109/ICIMIA48430.2020.9074929

Khan, S., Sivaraman, E. and Honnavalli, P.B. 2020. Performance evaluation of advanced machine learning algorithms for network intrusion detection system, in *Proceedings of International Conference on IoT Inclusive Life (ICIIL 2019), NITTTR Chandigarh, India*. Springer, pp. 51–59. Doi:10.1007/978-981-15-3020-3_6

Kocher, G. and Kumar, G. 2021. Analysis of machine learning algorithms with feature selection for intrusion detection using UNSW-NB15 dataset, *Available at SSRN 3784406* [Preprint]. Doi:10.2139/ssrn.3784406

Krishnaveni, S., Sivamohan, S., Sridhar, S. S., and Prabakaran, S. .2021. Efficient feature selection and classification through ensemble method for network intrusion detection on cloud computing, *Cluster Computing*, 24(3), pp. 1761–1779. Doi:10.1007/s10586-020-03222-y

Larose, D.T. and Larose, C.D. 2014. K-nearest neighbor algorithm. Doi:10.1002/9781118874059.ch7

Mebawondu, Alowolodu, O. D., Mebawondu, J. O., and Adetunmbi, A. O. 2020. Network intrusion

detection system using supervised learning paradigm, *Scientific African*, 9, p. e00497. Doi:10.1016/j.sciaf.2020.e00497

More, S., Idrissi, M., Mahmoud, H., and Asyhari, A. T. 2024. Enhanced intrusion detection systems performance with UNSW-NB15 data analysis, *Algorithms*, 17(2), p. 64. Doi:10.3390/a17020064

Mousavi, S.M., Majidnezhad, V. and Naghipour, A. 2022. A new intelligent intrusion detector based on ensemble of decision trees, *Journal of Ambient Intelligence and Humanized Computing*, 13(7), pp. 3347–3359. Doi:10.1007/s12652-019-01596-5

Moustafa, N. and Slay, J. 2015a. The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems, in *2015 4th international workshop on building analysis datasets and gathering experience returns for security (BADGERS)*. IEEE, pp. 25–31. Doi:10.1109/BADGERS.2015.014

Moustafa, N. and Slay, J. 2015b. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), in *2015 military communications and information systems conference (MilCIS)*. IEEE, pp. 1–6. Doi:10.1109/MilCIS.2015.7348942

Moustafa, N. and Slay, J. 2016. The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set, *Information Security Journal: A Global Perspective*, 25(1–3), pp. 18–31. Doi:10.1080/19393555.2015.1125974

Pathak, A. and Pathak, S. 2020. Study on decision tree and KNN algorithm for intrusion detection system, *International Journal of Engineering Research & Technology*, 9(5), pp. 376–381. Doi:10.17577/IJERTV9IS050303

Pietraszek, T. 2004. Using adaptive alert classification to reduce false positives in intrusion detection, in *Recent Advances in Intrusion Detection: 7th International Symposium, RAID 2004, Sophia Antipolis, France, September 15-17, 2004. Proceedings 7*. Springer, pp. 102–124. Doi:10.1007/978-3-540-30143-1_6

Pradhan, M., Nayak, C.K. and Pradhan, S.K. 2020. Intrusion detection system (IDS) and their types, in *Securing the internet of things: Concepts, methodologies, tools, and applications*. IGI Global, pp. 481–497. Doi:10.4018/978-1-5225-9866-4.ch026

Relan, N.G. and Patil, D.R. 2015. Implementation of network intrusion detection system using variant of decision tree algorithm, in *2015 International Conference on Nascent Technologies in the Engineering Field (ICNTE)*. IEEE, pp. 1–5. Doi:10.1109/ICNTE.2015.7029925

Siraj, M.J., Ahmad, T. and Ijtihadie, R.M. 2022. Analyzing ANOVA F-test and sequential feature selection for intrusion detection systems., *International Journal of Advances in Soft Computing & Its Applications*, 14(2). Doi:10.15849/IJASCA.220720.13

Song, J., Zhu, Z. and Price, C. 2014. Feature grouping for intrusion detection based on mutual information, *Journal of Communications*, 9(12), pp. 987–993. Doi:10.12720/jcm.9.12.987-993

Tapiador, J.E., Orfila, A., Ribagorda, A., and Ramos, B. 2013. Key-recovery attacks on KIDS, a keyed anomaly detection system, *IEEE Transactions on Dependable and Secure Computing*, 12(3), pp. 312–325. Doi:10.1109/TDSC.2013.39

Zeeshan, M., Riaz, Q., Bilal, M. A., Shahzad, M. K., Jabeen, H., Haider, S. A., and Rahim, A. 2021. Protocol-based deep intrusion detection for dos and ddos attacks using unsw-nb15 and bot-iot data-sets, *IEEE Access*, 10, pp. 2269–2283. Doi:10.1109/ACCESS.2021.3137201