A Hybrid Classification Approach of Network Attacks using Supervised and Unsupervised Learning

Rahaf Hamoud R. Al-Ruwaili, Osama M. Ouda
Department of Computer Science-College of Computer and Information Sciences,
Jouf University, Al-Jouf, Saudi Arabia

***Abstract*—The increasing scale and sophistication of network attacks have become a major concern for organizations around the world. As a result, there is an increasing demand for effective and accurate classification of network attacks to enhance cyber security measures. Most existing schemes assume that the available training data is labeled; that is, classification is based on supervised learning. However, this is not always the case since the available real data is expected to be unlabeled. In this paper, this issue is tackled by proposing a hybrid classification approach that combines both supervised and unsupervised learning to build a predictive classification model for classifying network attacks. First, unsupervised learning is used to label the data available in the dataset. Then, different supervised machine learning algorithms are utilized to classify data with the labels obtained from the first step and compare the results with the ground truth labels. Moreover, the issue of the unbalanced dataset is addressed using both over-sampling and under-sampling techniques. Several experiments have been conducted, using the NSL-KDD dataset, to evaluate the efficiency of the proposed hybrid model and the obtained results demonstrate that the accuracy of our proposed model is comparable to supervised classification methods that assume that all data is labeled.**

Keywords*—Network attacks; supervised learning; unsupervised learning; machine learning

I. INTRODUCTION

The extensive use of the Internet and its continuous development benefit many network users in many aspects. However, in recent years, cyber-attacks have become a growing concern due to their increasing complexity and diversity posing a major threat to governments, businesses, and networks[1]. As a result, network security becomes more important with the widespread use of the network. The purpose of network security is to provide protection and defense against misuse such as modification and unauthorized access.

The task of detecting anomalies in network traffic is experiencing growing demand due to the expanding internet accessibility among individuals[2]. The potential consequences of an intrusion on a computer network encompass a wide range of concerns, including but not limited to the compromise of confidentiality, integrity, and accessibility. These issues can manifest in various ways, such as breaches of privacy or compromise of systems. The primary classifications of intrusion detection systems encompass signature-based detection systems and anomaly-based detection systems[3]. Signature-based systems predominantly depend on established attack signatures to identify and detect unauthorized activities. In contrast, when encountering unfamiliar attack signatures, the identification of abnormal network activity is mostly conducted through the utilization of anomaly-based technologies.

There are many mechanisms trying to protect the network from outsiders, but these mechanisms can be hacked because the attacker spends enough time and resources to penetrate this perimeter, which may be mostly successful. Despite the multitude of mechanisms implemented to safeguard networks from external threats, determined attackers often invest significant time and resources to breach these defensive perimeters, leading to a high success rate. While firewalls are renowned as one of the most widely used network defense systems, they alone are insufficient in providing comprehensive protection against cyber-attacks. While access control policies play a crucial role in ensuring network security, they can be circumvented through passive authentication methods, thereby undermining their effectiveness. Passive authentication attacks pose a significant challenge to network security because they exploit the trust established within the network. By leveraging legitimate user credentials or session information, attackers can effectively bypass the access control policies implemented by firewalls.

This highlights the need for additional security measures beyond traditional firewall systems. To mitigate the risks associated with passive authentication attacks, organizations should consider implementing supplementary security measures such as strong user authentication protocols, encryption mechanisms, and intrusion detection systems. These layers of defense can help detect and prevent unauthorized access attempts, even if attackers manage to bypass the firewall's access control policies. By adopting a multi-faceted approach to network security, organizations can enhance their resilience against evolving cyber threats and minimize the potential impact of successful attacks[4]. On the other hand, encrypting stored data is a way to achieve data-centric security. However, encrypting stored data is not appropriate for all environments and contexts. Despite the many ways of protection, the attackers always find an entrance to fulfill their desires[5]. Thus, there is a need to identify methods for extracting security information from network data.

Most of the existing methods for classifying network attacks assume that the available datasets are labeled. Hence, they utilize supervised learning techniques to classify network packets. However, in real scenarios, network data are not labeled, and hence supervised learning-based classification methods might not be practically useful. Another issue with existing datasets is that most of the available datasets are

highly unbalanced in the sense that the training samples for some classes are much smaller than the samples available for other types. Ignoring this class imbalance increases the chances that the developed model will learn more about classes with large samples in the data set than about classes with fewer samples. This paper aims to address the class balance problem and use machine learning techniques to identify and extract useful security information from network data to classify different types of network attacks. Network attack classification plays a vital role in detecting and mitigating potential threats to network security. Traditional signature-based methods have limitations in identifying new and sophisticated attacks. Thus, the need for predictive models that can effectively identify attack patterns and classify them with a high degree of accuracy emerges [4] which affects the protection of the network.

Machine learning plays a pivotal role in the development and advancement of several domains within the field of IDS. Statistical methodologies and methods are utilized in order to train the model using a set of training data. When faced with unfamiliar data, the system extracts distinctive and hidden patterns from the dataset in order to provide predictions or classifications, thereby forecasting future trends based on the available data[6]. There are two primary categories of machine learning algorithms: unsupervised learning and supervised learning. In the training phase of Supervised Learning, the model is trained using data that includes both the dependent variable and its corresponding outcome. Conversely, in Unsupervised Learning, the model is trained on unlabeled data, which consists of input data without any associated output information[7].

In this study, an unsupervised pooling technique has been devised, utilizing the K-means method for the purpose of detecting and grouping network intrusion. Next, a supervised learning technique was employed using three distinct algorithms: Random Forest (RF), K-Nearest Neighbor (KNN), and Support Vector Machines (SVM), in order to classify attacks. The NSL-KDD dataset was utilized in this study, employing two distinct methodologies, namely oversampling and under sampling, to ensure the maintenance of data balance.

The main contributions of this work are outlined as follows:

- A data preprocessing strategy is provided as well as data sampling techniques that aim at achieving a more accurate representation of the dataset's features, with the ultimate goal of reducing model bias.
- Introduce and compare the utilization of three models, namely Random Forest (RF), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM), in the context of classifying network intrusion attempts. The classification task is performed using unsupervised learning through the application of the K-means algorithm.
- Presenting a complete review of network intrusion attacks, focusing on the datasets used in previous research. The analysis evaluates the accuracy of four different models, identifying the most realistic model among them.

The subsequent sections of the paper are structured in the following manner: The literature review is presented in Section II. The suggested method is explicated in Section III. Section IV discusses the evaluation metrics while the discussion of the outcomes is presented in Section V. The final Section VI contains future work and the conclusion.

II. Related Works

In [1], machine learning was used to detect the occurrence of malicious attacks and introduce a feature-based transfer learning framework and transfer learning approach. They also introduced the feature-based learning approach using a linear transformation, called HeTL. A cluster-enhanced transfer learning approach, called CeHTL, has been proposed to make it more potent to detect unknown attacks and evaluation of learning transfer approaches on shared workbooks. The results show that transfer learning methods improve the detection performance of unknown network attacks compared to baselines.

In [5], the authors cover most of the papers that have been released on the attack and defense of membership inference on ML models. They familiarize MIAs with ML models and present current attack methods. They also rated all MIAs papers next to discuss why MIAs work on ML models and summarize the most current evaluation metrics, datasets, and open-source applications of common approaches.

In [4], the researchers proposed a machine learning approach to classify and predict types of DDoS attacks. The authors also used Random Forest and XGBoost classification algorithms. The UNWS-np-15 dataset was extracted from the GitHub repository and Python was used as a simulation. After applying the machine learning models, they generated a confusion matrix to determine the performance of the model. For the Random Forest algorithm, the results show that both Precision (PR) and Recall (RE) are ~89%. For the XGBoost algorithm, the results show that both Precision (PR) and Recall (RE) are about 90%.

The researchers in [8] proposed an efficient framework that learns minimal temporal preferential attack targeting the LSTM model with electronic medical record inputs, they also proposed an efficient and effective framework that identifies sensitive locations in medical records using adversarial attacks on deep predictive models. The results showed weakness in the deep models, as it was more than half of patients can be successfully attacked by changing only 3% of the recording sites with maximum perturbation less than 0.15 and mean perturbation less than 0.02.

In [9], the researchers suggested a stack-based ensemble approach to obtain reliable predictions by combining different algorithms. A powerful processing model called Graphlab Create (GC) was used to perform experiments involving many cases. Recent datasets consisting of attack types were compiled from the UNSW NB-15 and UGR'16 datasets.

In [10], SVM models detect malicious behavior within low-power, low-speed, and short-range networks. They evaluated two SVM approaches, namely C-SVM and OC-SVM. Actual network traffic was used along with the specific network layer attacks that they have implemented to generate and evaluate

VPM detection models. They show that C-SVM achieves a classification accuracy of 100% when evaluated with unknown data taken from the same network topology in which it was trained and an accuracy of 81% when running in unknown topologies.

In [11] the authors proposed the first survey of its kind on adversarial attacks on machine learning in network security. They discussed aggressive attacks against deep learning in computer vision only. They introduced a new classification of adversarial attacks based on machine learning applications in network security and developed a matrix to correlate different types of adversary attacks with a classification-based classification to determine their effectiveness in causing misclassification. A new idea of the adversarial risk network map concept was presented for machine learning in network security.

In [12] they compared different classifiers in the NSL-KDD dataset for binary and multiclass classification. SVM, random forest, and LSTM-RNN model were considered. They show that the proposed model produced the highest accuracy rate of 96.51% and 99.91% for binary classification using 122 features and an optimal set of 99 features, respectively. LSTM-RNN obtained higher accuracy than SVM in binary classification.

In [13], the researchers presented an exploration of how adversarial learning can be used to target supervised models by generating adversarial samples using the Jacobian-based Saliency Map attack and an exploration of classification behaviors. An authentic power system dataset was used to support the experiments presented. The classification performance of two widely used classifiers, Random Forest and J48, decreased by 6 and 11 percentage points when hostile samples were present.

In [14], the authors aimed to detect distributed denial-of-service (DDOS) attacks on financial institutions by using banking datasets. They used multiple classification models to predict DDOS attacks. Some complexity has been added to the architecture of the generic models to enable them to perform well and application of support vector machine (SVM), k-nearest neighbors (KNN), and random forest (RF) algorithms. SVM showed an accuracy of 99.5%, while KNN and RF recorded an accuracy of 97.5% and 98.74%, respectively, for detecting DDoS attacks. When compared, it is concluded that SVM is more powerful compared to KNN, RF, and existing machine learning (ML) and deep learning (DL) approaches.

In [15], the authors applied the MeanShift algorithm to detect an attack in a network traffic dataset and evaluated the performance of the MeanShift algorithm by two metrics. These metrics are detection rate and detection accuracy. The results of this study showed that the detection rate of the MeanShift algorithm was 79.1 percent, and the detection accuracy of the MeanShift algorithm was 81.2 percent.

In [16], the authors proposed a method for infiltration detection based on deep neural networks. They trained the encoder block based on self-supervised variance learning using unclassified training patterns. Then they inserted the resulting representation into the classification header which was trained using a labeled data set.

In [17] they implemented the machine learning-based detection, classification, and investigation of flood DDoS attacks. They used four supervised learning methods (CART, K-NN, QDA, GNB) and implemented them well, but CART outperforms others based on the investigations that have been conducted.

In [18] they performed a comparative study to analyze the performance of ML algorithms for intrusion detection on the NAL-KDD dataset. They selected only the relevant features. They concluded a reliable identity detection system capable of real-time intrusion detection using different. Table I summarize some related works and provide a comparison between different approaches.

TABLE I. SUMMARY AND COMPARISON OF THE RELATED WORKS

Ref.	Author & year	Study name	Method or Technique	Dataset	Accuracy	Notes
[1]	Zhao,Juan et al. 2019	Transfer Learning for Detecting Unknown Network Attacks	transfer learning approach HeTL and CeHTL	NSL-KDD	0.93%	Assumed The Data Is Labelled.
[4]	Mohmand, Muhammad Ismail et al.2022	A Machine Learning-Based Classification and Prediction Technique for DDoS Attacks	Was Used Random Forest and XGBoost Classification Algorithms	KDD, UNWS-np-15	90%	They Got Good Accuracy Using XGBoost Algorithm But by Using Other Algorithms The Accuracy Was Low
[9]	Rajagopal, Smitha et al.2020	A predictive Model for Network Intrusion Detection Using Stacking Approach	model Graphlab Create (GC) was used	UNSW NB-15 and UGR' 16	95%	It Is Only Limited To Use Of The Graphlab Construct (GC) Model.
[10]	Ioannou, Christiana et al.2021	Network Attack Classification in IoT Using Support Vector Machines	C-SVM and OC-SVM	-	100%	
[12]	Muhuri, Pramita Sree et al.2020	Using a Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) to Classify Network Attacks	SVM, random forest, and LSTM-RNN model were considered.	NSL-KDD	99.91 %	LSTM-RNN performs Poorly. In this Experiment The Training Time Was Not Recorded
[14]	Islam, Umar et al.2022	Detection of Distributed Denial of Service (DDoS) Attacks in IOT Based Monitoring System of Banking Sector Using Machine Learning Models	Has been used multiple classification models to predict DDOS attacks and SVM, KNN, RF algorithms	Bank Dataset	99.5 %	This Model Is Limited To Offline Datasets

[15]	Kumar, Avinash et al.2020	Network Attack Detection Using an Unsupervised Machine Learning Algorithm	MeanShift algorithm	KDD 99	81.2 %	MeanShift Algorithm used Did Not Detect The R2L and U2R Attack Types.
[16]	Lotfi, S et al.2022	Network Intrusion Detection with Limited Labeled Data Using Self-supervision	Supervised and Self-supervised	UNSW-NB15	%94.05	Detect Intrusion With Limited Number Of Labeled Data
[17]	Sangodoyin, Abimbola O. et al.2021	Detection and Classification of DDoS Flooding Attacks on Software-Defined Networks: A Case Study for the Application of Machine Learning	Supervised Learning Methods and (DA,NB,DT, k-NN)	Dataset For The SDN Classes Of Events (Normal, TCP,HTTP, UDP)	%98	Only DDos Attack Was Used, This Study Was Limited To The Supervised Learning Method, CART and k-NN Their Hyperparameters Have Not Been Tuned
[18]	Masoodi Faheem et al.2021	Machine Learning for Classification analysis of Intrusion Detection on NSL-KDD Dataset	Comparative Study of Performance Analysis Of Various ML Algorithms	NSL-KDD	%100	U2L Attacks Did Not Produce Enough Results. There Is No Solution To The Problem Of Security Vulnerabilities in Machine Learning Algorithms

III. Proposed Method

For developing a hybrid classification model for classifying network attacks, a standard approach was followed, relying on the NSL-KDD dataset. Fig. 1 illustrates the workflow that is followed to achieve the goal of detecting intrusions. Initially, the NSL-KDD dataset is acquired, after which data pre-processing takes place. The pre-processing procedure involves many steps that are required to render the data suitable for the algorithms that will be used later. For instance, in this study, data pre-processing includes removing null and duplicated values, in addition to data normalization and fixing the oversampling and under-sampling issues. After that, the data is fed to unsupervised machine learning model, where a K-means algorithm is used, followed by a supervised learning model where three different algorithms are implemented: Random Forest, Support Vector Machine, and K-Nearest Neighbor. Finally, the performance of each of these algorithms is evaluated according to F1 score, involving recall and precision.

A. Dataset

NSL-KDD dataset has been significantly popular in the field of intrusion detection due to its qualities, among many other datasets that are frequently used, whether they are private, public, or simulated network traffic datasets. Initially, Tavallaee et al. [19] proposed the NSL-KDD dataset as an enhancement of the KDD-99 cup dataset, overcoming its numerous issues, such that the enhanced dataset only contains the selected records from the complete KDD dataset. Despite the enhancement process, the NSL-KDD dataset still suffers from minor issues such as its lack of representation of the low-footprint attacks [20].

The choice fell upon the NSL-KDD dataset since it has less data points than KDD-99, the number of selected records from each difficult level group is inversely proportional to the percentage of records in the original KDD dataset, and it includes no duplicate records in the test set, ultimately leading to better reduction rates. Furthermore, the selected dataset provides less computational expenses for training ML models. Overall, the NSL-KDD dataset contains 41 attributes and one class attribute [21]. The class attribute indicates the type of network traffic, which can be one of five classes: normal, DoS, Probe, R2L, and U2R. The label counts for the NSL-KDD dataset are as follows: normal: 76967, DoS: 52985, Probe: 13954, R2L: 3749, U2R: 252 The dataset has a total of 147,907 rows and 42 columns, with the additional column being the class attribute.

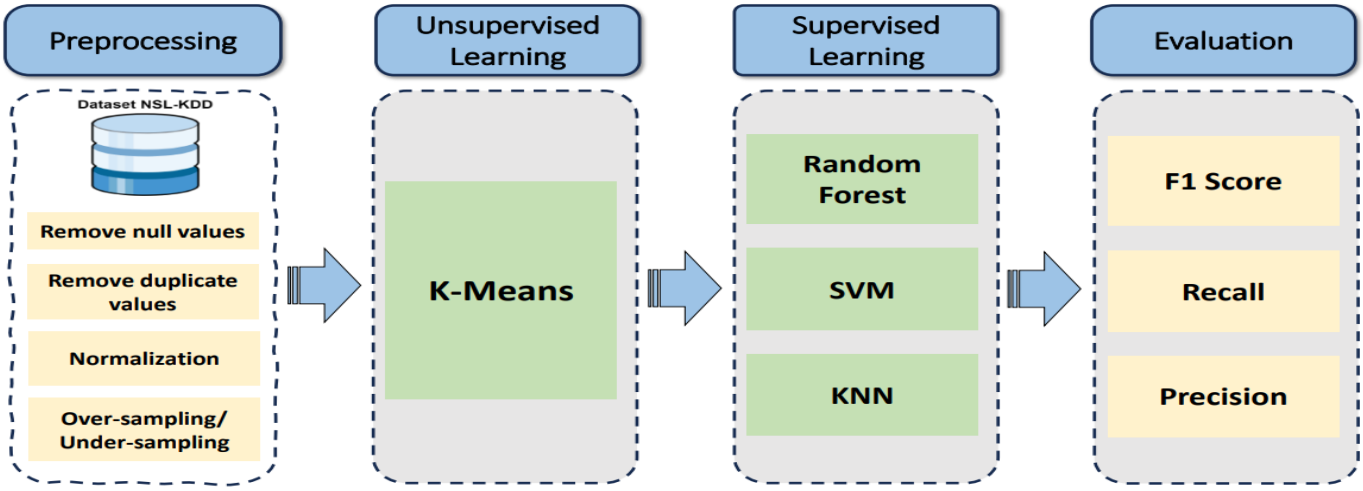

Fig. 1. Proposed framework for the hybrid classification model.

B. Exploratory Data Analysis

Exploratory data analysis EDA is one of the essential steps on any given dataset, as it allows the understanding of the data through observing and analyzing its characteristics, usually by charts. EDA also helps in identifying patterns, possible anomalies, and possible outliers in the data.

The total number of labels within the dataset is 147907 distributed over the following labels: normal, DoS, Probe, R2L, and U2R. The distribution of these labels is presented in Fig. 2 such that the percentage of each label among the whole data is given respectively. Upon observation, the normal label takes up 52% of the total labels which is the highest percentage, followed by 35.8% taken by the DoS label (from the attack labels). Additionally, the Probe labels take up 9.4% of the total labels, while R2L and U2R labels have the lowest percentages.

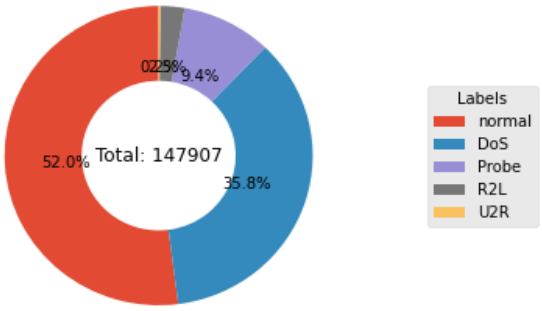

Fig. 2. Labels distribution in the NSL-KDD dataset.

In addition, there are three protocol types within the dataset, namely tcp, udp, and icmp. The percentage of each of these protocol types is shown in Fig. 3. The majority of the protocols are represented by the tcp protocol (82.1%) followed by the udp protocol (11.9%) and the icmp protocol (6.1%).

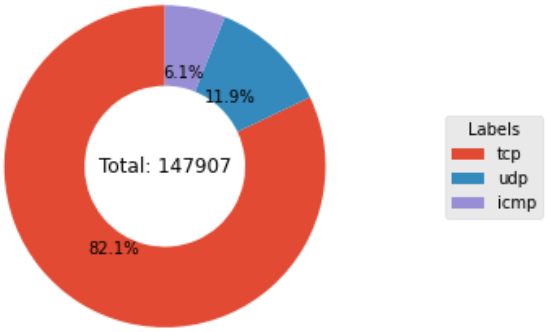

Fig. 3. Protocol type distribution in the NSL-KDD dataset.

It is also possible to know the distribution of the different labels within the dataset over the protocol types that are present. This data is given in Fig. 4. Fig. 4 shows the relationship between the labels and the protocols, where the count plot shows that most of the attacks are carried out using the TCP protocol, with DoS attacks being the most prevalent in this protocol category. It can also be seen that DoS attacks are most prevalent in the UDP protocol, even though UDP doesn't present that many attacks compared to the TCP protocol. Finally, the probe attacks are the most prevalent label within the icmp protocol.

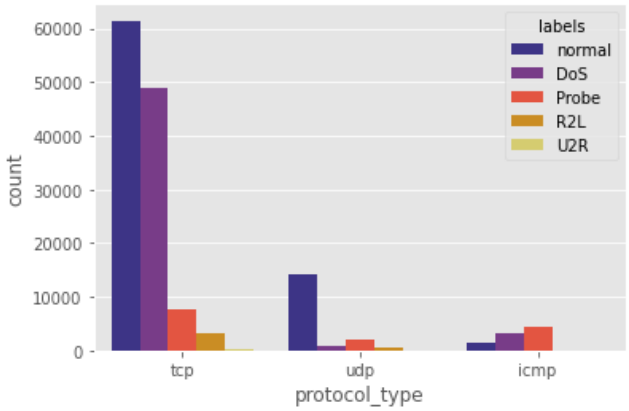

Fig. 4. Label-Protocol distribution count plot.

The relationship between flags and labels can also be acquired from EDA, as presented in Fig. 5. There are numerous flags, namely normal REJ, SF, S0, RSTO, RSTR, RSTOS0, S1, S3, S2, SH, and OTH. It appears that most of the attacks were carried out through the SF flag, where the other dominant flags are REJ flag and S0 flag. The SF flag shows a high count of normal label, whereas the S0 flag shows the highest count of attacks, namely the DoS attacks. It's noteworthy that the DoS attack dominates the REJ flag as well, but in a less prominent way.

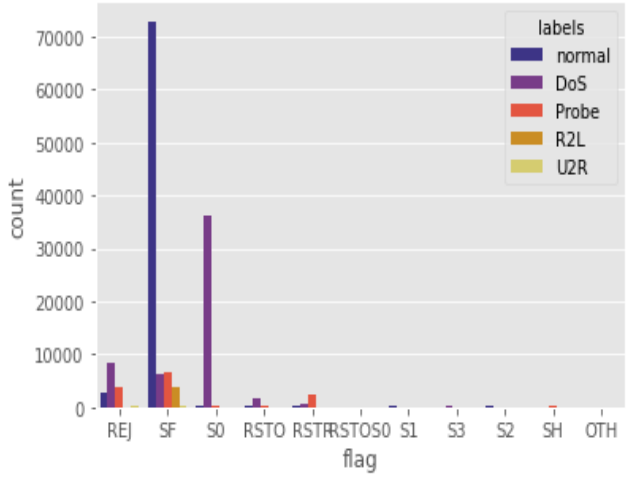

Fig. 5. Label-Flag distribution count plot.

C. Data Pre-processing

Data pre-processing is yet another essential step to prepare the data for use on different machine learning algorithms. The purpose of applying data pre-processing techniques is to improve the quality of data by removing noise and dealing with missing values, for example. It also enhances the efficiency of data analysis and interpretation, while also improving the overall performance of the model. In this study, three main steps were followed as data pre-processing procedures.

1) Missing values: The NSL-KDD dataset was checked for missing values or duplicate values. After inspection, it was clear that the dataset does not contain any missing values or duplicate values, which renders it of high quality.

2) Normalization: Data normalization works on transforming the data into the same scale without interfering with the relationship between variables or their distribution. This step helps in improving the efficiency and accuracy of the ML model. In this study, the MinMaxScaler function was used to normalize the data, scaling them to a range between 0 and 1.

3) Class imbalance: In cases of class imbalance, such as the case of the NSL-KDD dataset, the end results of the ML can be biased. For this reason, the NSL-KDD dataset was subjected to over-sampling and under-sampling to fix the class imbalance issue. Under-sampling is a method used to decrease the number of samples in classes that are overrepresented in a dataset. This can be achieved by randomly selecting a portion of the samples or eliminating samples that have a high degree of similarity to other samples in the dataset. Conversely, oversampling is a technique that aims to increase the number of samples in minority classes by creating synthetic data points.

TABLE II. COUNTS OF EACH CLASS BEFORE AND AFTER SAMPLING METHODS

Sampling Method	Class	Count Before	Count After
Over-sampling	normal	76967	76967
	DoS	52985	76967
	Probe	13954	76967
	R2L	3749	76967
	U2R	252	76967
Under-sampling	normal	76967	252
	DoS	52985	252
	Probe	13954	252
	R2L	3749	252
	U2R	252	252

Table II shows the count of each of the five classes before and after the class imbalance procedures, which are over-sampling and under-sampling. For instance, the normal class contained 76967 instances before under-sampling, and that number became 252 after the procedure was done. Another example is the R2L class which contained 3749 instances before over-sampling, and after the procedure the count became 76967.

The shape of the data before and after the two sampling methods (over- and under-sampling) can be visualized in Table III.

TABLE III. SHAPE OF DATA BEFORE AND AFTER SAMPLING METHODS

Sampling Method	Shape Before	Shape After
Over-sampling	(147907, 122)	(384835, 122)
Under-sampling	(147907, 122)	(1260, 122)

D. Classification Methods

Since there are five different classes or labels within the dataset, this means that the classification problem is a five-class classification problem, where the classes are: benign (or normal), u2r, r2l, dos, and probe. In addition, multiple algorithms exist such that they support this kind of multi-class classification task. Yet, selecting the suitable ML algorithm is the obvious challenge in this case. In this study, initially the cases where labeled data can be used will be considered, which means that supervised machine learning techniques will be used. After that, semi- and un-supervised machine learning techniques will be considered as well.

1) Supervised classification: For classifying attacks through supervised classification, three supervised machine learning algorithms were chosen, namely: Random Forest RF, K-Nearest Neighbor KNN, and Support Vector Machine SVM. The dataset is divided by a 80:20 ration for training and testing respectively, upon which these three ML algorithms will be trained and evaluated.

a) Random forest: RF is one of the supervised machine learning algorithms, and its concept is randomly creating a forest of decision trees such that the number of the trees directly correlates with the accuracy of performance. Yet, it is noteworthy to consider that creating the forest is different from constructing a decision tree using the information gain or gain index approach. The main difference between Random Forest and Decision Tree algorithms is that in Random Forest, the processes of finding the root node and splitting the feature nodes occur randomly [22]. Two stages are required for RF classification: the forest creation stage where decision trees are created, and the prediction stage where predictions take place.

The Random Forest algorithm creation method involves the following steps:

- (a) Randomly selecting "K" features from the total "m" features, where k << m.
- (b) Calculating the node "d" among the "K" features using the best split point.
- (c) Splitting the node into daughter nodes using the best split.
- (d) Repeating steps a to c until "l" number of nodes has been reached.
- (e) Building the forest by repeating steps a to d for "n" number of times to create "n" number of trees.

b) K-Nearest Neighbor: KNN is described as a non-parametric and lazy learning method. Non-parametric indicates that no assumptions are made about the underlying data distribution, whereas a lazy algorithm indicates the no need for any training data points to achieve model construction. K-nearest neighbors (K-NNs) classifier depends on Manhattan or Euclidean distances to evaluate similarities or differences between instances in the dataset. Often, the Euclidean distance is the metric of choice in KNN classifiers. In KNN, k represents the number of nearest neighbors used

for classification. The algorithm finds the data point with the minimum distance to the test point and assigns it to the same class [23].

Even though KNN is a simple algorithm to implement, it still has the disadvantage of slow prediction time because of calculating the distance between each data point.

The KNN algorithm is implemented by following these steps:

- Loading the data
- Initializing the value of k
- Iterating from 1 to the total number of training data points (*to obtain the predicted class*).
  - Calculating the distance between the test data and each row of training data using a distance metric such as Euclidean, Chebyshev or cosine.
  - Sorting the calculated distances in ascending order based on distance values.
  - Retrieving the top k rows from the sorted array.
  - Determining the most frequent class of these rows
  - Returning to the predicted class.

c) Support Vector Machine: SVM is a supervised type of machine learning. algorithm in which, given a set of training examples, each marked as belonging to one of the many categories, an SVM training algorithm builds a model that predicts the category of the new example. SVM has the greater ability to generalize the problem, which is the goal in statistical learning. The statistical learning theory provides an outline for studying the problem of gaining knowledge, making predictions, and making decisions from a set of data. SVM is a type of linear and non-linear classifier, which is a mathematical function that can distinguish between two different classes of objects [24]. VM has the benefit of being capable of managing high-dimensional data and data with non-linear decision boundaries. Nevertheless, its drawback is that it can be computationally demanding and necessitates careful adjustment of the hyperparameters, such as C and the kernel function, to achieve the best possible performance.

Training an SVM algorithm can be achieved with the following pseudocode:

Require: X and y containing the labeled training data, α<= 0 or α<= partially trained SVM

- C<= some value (10 for example)
- repeat
- for all {xi, yi}, {xj,yj} do
- Optimize α_i and αj
- end for
- until no changes in α or other resource constraint criteria met

Ensure: Retain only the support vectors ($\alpha_i > 0$)

In SVM, the C value is a regularization parameter that manages the balance between maximizing the margin and minimizing the classification error. The algorithm progressively improves the values of α_i and α_j to locate the support vectors, which are the data points nearest to the decision boundary. After the algorithm concludes, only the support vectors with $\alpha_i > 0$ are maintained.

2) Unsupervised classification: The NSL-KDD dataset was also clustered using an unsupervised ML algorithm. K-Means clustering is employed to group similar instances together and new labels are predicted for the instances. The predicted labels will then be used as the target variable, and the instances will be classified using the same supervised ML algorithms (KNN, SVM, and RF). The performance of each algorithm will be evaluated using the same performance metrics utilized in the supervised classification.

a) K-Means: K-Means is an unsupervised clustering technique that is frequently employed for partitioning data into k-clusters. The algorithm is iterative and aims to obtain the optimal value for each iteration. Initially, a preferred number of clusters is chosen, and the data points are distributed into k clusters. A greater k produces smaller groups with finer detail, while a lower k results in larger groups with less detail.

The K-Means algorithm can be summarized in two steps that are repeated until the clusters and their means are stable:

i. Assign each data item to the nearest cluster center. The nearest distance can be calculated using distance algorithms.
ii. Calculate the mean of the cluster with all data items [23].

IV. Evaluation Metrics

The performance of the proposed algorithms is evaluated based on their results in the testing dataset. There are several metrics that can be used to evaluate the performance of the models such as precision, recall, and f1 score. Recall is another term used for sensitivity, which resembles the true positive value, which is also the portion of the correctly classified inputs as positive among the entire inputs that should have been classified as positive. Precision is the portion of the true positive classifications over the entirety of the positive results. F-measure is the harmonic mean of the precision and recall and sums up the predictive performance of a model.

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

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

$$F - Measure = \frac{2\,(Precision \times Recall)}{Precision + Recall}$$

True positive is designated by TP. True negative is designated by TN. False positive is designated by FP. False negative is designated by FN.

V. RESULTS

In computer security, intrusion detection encompasses monitoring computer systems and network to look out for any potential threats embodied by malicious activities, security breaches, or unauthorized access. For intrusion detection models, usually NSL-KDD dataset is used since it comprises many network connections that can be classified as normal traffic or attacks of different types. In this study, two different approaches are implemented to detect intrusion, namely the supervised learning approach (through RF, KNN, and SVM), and the unsupervised clustering followed by supervised learning approach. Both approaches are compared according to their performance in terms of accuracy, precision, recall, f1 score, and confusion matrix. By examining the results of these approaches, the strength, and limitations of each one of them becomes clearer, and it can be considered as an insight into building effective intrusion detection systems and identifying areas for future research.

A. Supervised Learning Approach

1) Random Forest algorithm after over-sampling: After applying oversampling techniques, the random forest algorithm was able to score a perfect accuracy rate of 100% on the NSL-KDD dataset. Among the five different classes within the dataset (DoS, Probe, R2L, U2R, and normal), the RF model also scored high precision, recall, and f1 score values as can be seen in Table IV. In fact, RF achieves perfect performance when taking into consideration all the evaluation metrics. These results prove that this model accurately detects intrusions and normal connections.

TABLE IV. RANDOM FOREST CLASSIFICATION REPORT / OVER-SAMPLING

Algorithm	Accuracy	Precision	Recall	F1-Score
	100%	1.00	1.00	1.00

Aside from the perfect recall, precision, and f1 score, the RF algorithm also correctly predicted most of the instances except for a few misclassifications that can be seen in the confusion matrix in Table V. The instances in the attacks category (DoS, Probe, R2L, and U2R attacks) were all perfectly classified, yet the misclassifications fall in the normal category. More specifically, 10 normal connections were classified as DoS attacks, 14 as Probe attacks, and 40 as R2L attacks. These misclassifications may be due to the similarities in network traffic patterns between normal connections and certain types of attacks.

TABLE V. RANDOM FOREST CONFUSION MATRIX / OVER-SAMPLING

	DoS	Probe	R2L	U2R	normal
DoS	15602	1	0	0	2
Probe	0	15447	0	0	0
R2L	0	0	15286	0	0
U2R	0	0	0	15400	0
normal	10	14	40	0	15165

2) K-Nearest Neighbor algorithm after over-sampling: After over-sampling on the NSL-KDD dataset, the KNN algorithm was able to score a perfect 100% accuracy in classifying intrusions. Similarly, the precision, recall, and F1 score values were very high as can be seen in Table VI. These results indicate that the KNN algorithm can identify attack classes with ease, while finding some difficulties in correctly identifying all of the normal connections.

TABLE VI. KNN CLASSIFICATION REPORT / OVER-SAMPLING

Algorithm	Accuracy	Precision	Recall	F1-Score
	100%	1.00	1.00	1.00

As for the confusion matrix depicted in Table VII, the results show that the KNN algorithm can correctly identify most of the instances as attacks and normal connections, with a few errors in the normal class. For instance, 21 normal connections were identified as DoS attacks, 46 as Probe attacks, 150 as R2L, and 15 as U2R attacks. This reflects the poor ability of KNN to classify normal connections. On the other hand, KNN was able to correctly identify all of the instances within the DoS, Probe, R2L, and U2R classes.

TABLE VII. KNN CONFUSION MATRIX / OVER-SAMPLING

	DoS	Probe	R2L	U2R	normal
DoS	15596	3	0	0	6
Probe	9	15425	1	0	12
R2L	0	0	15286	0	0
U2R	0	0	0	15400	0
normal	21	46	150	15	14997

3) Support Vector Machine algorithm after over-sampling: The achieved accuracy level by the SVM algorithm after over-sampling of the NSL_KDD dataset was 96%. Similarly, all the other metrics reached high values as shown in Table VIII. for all of the attack classes as well as the normal classes. Even in terms of precision, the SVM model achieved lower values scoring 0.97, with 0.96 recall and 0.96 F1-score.

TABLE VIII. SVM CLASSIFICATION REPORT / OVER-SAMPLING

Algorithm	Accuracy	Precision	Recall	F1-Score
	96%	0.97	0.96	0.96

Table IX describing the confusion matrix of SVM algorithm shows that the model achieves acceptable results in the attack classes, where only a few misclassifications took place. on the other hand, it was demonstrated that the model performs poorly in identifying the normal classes, where a lot of misclassifications can be seen. 76 normal connections were mistakenly identified as DoS, 159 were mistakenly identified as Probe, 122 were mistakenly identified as U2R, and 713 were mistakenly identified as R2L. Another class that shows a rather poor performance of SVM is the U2R class, were 1031 instances were mistakenly identified as R2L attacks. The SVM model rather shows a better performance in the other classes.

TABLE IX. SVM CONFUSION MATRIX / OVER-SAMPLING

	DoS	Probe	R2L	U2R	normal
DoS	15511	8	0	0	86
Probe	26	15315	4	17	85
R2L	0	17	14859	149	261
U2R	0	0	1031	14369	0
normal	76	159	713	122	14159

4) Random forest algorithm after under-sampling: When under-sampling techniques were followed, the RF model scored 98% accuracy on the NSL-KDD dataset.

Table X illustrates the high values of accuracy, precision (0.98), recall (0.97), and F1 score (0.98) achieved on all the attack classes as well as the normal class. The achieved results, however, were lower than those scored by RF in over-sampling.

TABLE X. RANDOM FOREST CLASSIFICATION REPORT / UNDER-SAMPLING

Algorithm	Accuracy	Precision	Recall	F1-Score
	98%	0.98	0.97	0.98

Interestingly, the confusion matrix of RF after under-sampling, shown in Table XI demonstrates nearly perfect classifications in all classes, including the normal class. There is 1 misclassification only in the DoS class (classified as Probe), and 1 misclassification only in the Probe class, identified as R2L attack. Other than that, the RF forest after under-sampling is so far the only algorithm that perfectly classified all of the normal connections.

TABLE XI. RANDOM FOREST CONFUSION MATRIX / UNDER-SAMPLING

		DoS	Probe	R2L	U2R	normal
DoS		29	1	0	0	0
Probe		0	26	1	0	0
R2L		0	0	17	0	0
U2R		0	0	0	28	0
normal		0	0	0	0	24

5) K-Nearest Neighbor algorithm after under-sampling: After performing under-sampling on the dataset, KNN was able to achieve an overall of 96% accuracy in predicting the classes. Tabel XII shows that KNN achieved a good precision (0.97), good recall (0.96) and goof F1-score (0.96).

TABLE XII. KNN CLASSIFICATION REPORT / UNDER-SAMPLING

Algorithm	Accuracy	Precision	Recall	F1-Score
	96%	0.97	0.96	0.96

Furthermore, the confusion matrix for KNN after under-sampling in Table XIII shows that KNN perfectly classified U2R and Normal classes, while it misclassified DoS in 1 occasion (as Probe) only. Probe class was also misclassified only once by KNN as R2L. The most misclassifications achieved by KNN were in the R2L class, where it misclassified 2 of them as DoS.

TABLE XIII. KNN CONFUSION MATRIX / UNDER-SAMPLING

	DoS	Probe	R2L	U2R	normal
DoS	29	1	0	0	0
Probe	0	26	1	0	0
R2L	2	0	15	0	0
U2R	0	0	0	28	0
normal	0	0	0	0	24

6) Support Vector Machine Algorithm after under-sampling: After under-sampling, SVM was able to achieve a total of 96% accuracy on the NSL-KDD dataset. Table XIV shows that SVM has good precision (0.97), recall (0.96), and F1 score (0.96).

TABLE XIV. SVM CLASSIFICATION REPORT / UNDER-SAMPLING

Algorithm	Accuracy	Precision	Recall	F1-Score
	96%	0.97	0.96	0.96

In addition, the confusion matrix for SVM shows in (Table XV) that it can perfectly classify normal and U2R labels without any misclassifications. On the other hand, SVM misclassifies DoS in 1 instance as probe, it also misclassifies Probe in 1 instance as R2L. SVM has 2 misclassifications in the R2L label, where 2 instances are falsely classified as DoS.

TABLE XV. SVM CONFUSION MATRIX / UNDER-SAMPLING

	DoS	Probe	R2L	U2R	normal
DoS	29	1	0	0	0
Probe	0	26	1	0	0
R2L	2	0	15	0	0
U2R	0	0	0	28	0
normal	0	0	0	0	24

B. Overall Performance in Supervised Classification

The scores achieved by all of the supervised algorithms "RF, KNN, and SVM" are shown in Fig. 6. The performances are shown in terms of accuracy, Precision, Recall, and F1-score in both cases of over-sampling and under-sampling.

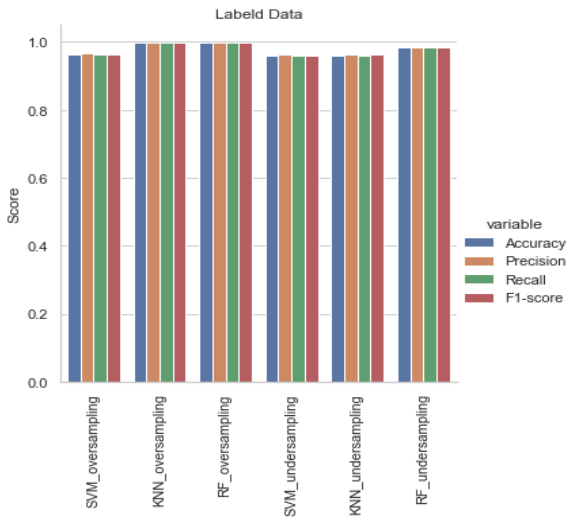

Fig. 6. Performance of supervised ML algorithms in over- and under-sampling.

The results of the three algorithms in terms of accuracy through over-sampling and under-sampling of the NSL-KDD dataset are shown in Table XVI.

TABLE XVI. ACCURACY RESULTS FOR SUPERVISED LEARNING ALGORITHMS IN OVER-SAMPLING AND UNDER-SAMPLING

	KNN	SVM	Random Forest
Over Sampling Accuracy	100%	96%	100%
Under Sampling Accuracy	96%	96%	98%

C. Overall Performance in Unsupervised Classification

The outcomes of the unsupervised categorization employing K-Means indicated that the precision of the supervised algorithms fluctuated based on the sampling method utilized. Following oversampling, the SVM and Random Forest algorithms attained an accuracy of 0.94, whereas KNN attained an accuracy of 0.92. In contrast, following under-sampling, KNN achieved an accuracy of 0.92, while SVM and Random Forest attained an accuracy of 0.94. These results are presented in

XVII.

TABLE XVII. PERFORMANCE OF ALGORITHMS AFTER K-MEANS CLUSTERING AS UNSUPERVISED CLASSIFICATION

	KNN	SVM	Random Forest
Over Sampling	92%	94%	94%
Under Sampling	92%	94%	93%

In addition, the same results can be visualized in Fig. 7 which shows the accuracy, Precision, Recall, and F1-score values for SVM, KNN, and RF after K-means clustering, in both over- and under-sampling techniques on the NSL-KDD dataset.

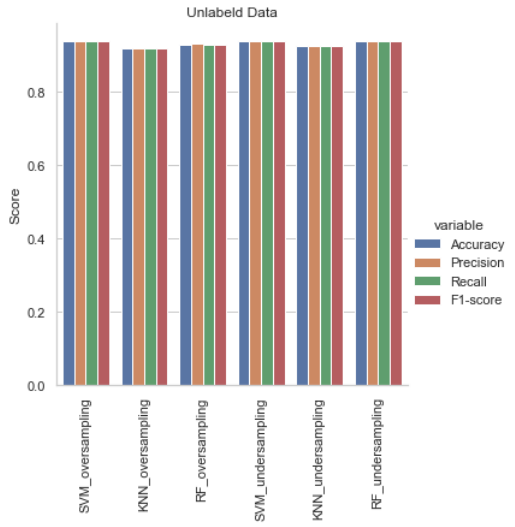

Fig. 7. Performance of unsupervised classification in over- and under-sampling.

VI. CONCLUSION AND FUTURE SCOPE

Intrusion detection is an essential component of cybersecurity and organizations of all sizes to protect their networks and systems from attacks. Effective intrusion detection requires a combination of technical tools and expertise and a thorough understanding of the organization's potential threats and vulnerabilities. This paper proposed a hybrid intrusion detection method that employs both unsupervised and supervised learning to address those issues of unlabeled and unbalanced datasets. Several supervised learning techniques including Random Forest, K-Nearest Neighbor, and Support Vector Machine were tested along with the K-means unsupervised classification technique. The main task was to perform intrusion detection by classifying traffic data as Normal, DoS, Probe, R2L, and U2R after training the ML algorithms on the NSL-KDD dataset. The obtained results showed that all algorithms can achieve high accuracy, recall, and F1 score. In the future, the integration of ensemble models for classification [25] can be explored. Moreover, the utilization of federated learning to maintain data integrity and privacy [26], and the adoption of transformer ViT models[27] to enhance network attack defense across many datasets can be considered.

REFERENCES

[1] J. Zhao, S. Shetty, J. W. Pan, C. Kamhoua, and K. Kwiat, "Transfer learning for detecting unknown network attacks," EURASIP J. Inf. Secur., vol. 2019, no. 1, p. 1, Dec. 2019, doi: 10.1186/s13635-019-0084-4.

[2] A. Devarakonda, N. Sharma, P. Saha, and S. Ramya, "Network intrusion detection: a comparative study of four classifiers using the NSL-KDD and KDD'99 datasets," J. Phys. Conf. Ser., vol. 2161, no. 1, p. 012043, Jan. 2022, doi: 10.1088/1742-6596/2161/1/012043.

[3] I. P. Saputra, E. Utami, and A. H. Muhammad, "Comparison of Anomaly Based and Signature Based Methods in Detection of Scanning Vulnerability," in 2022 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Oct. 2022, pp. 221–225. doi: 10.23919/EECSI56542.2022.9946485.

[4] Y. Wu, D. Wei, and J. Feng, "Network Attacks Detection Methods Based on Deep Learning Techniques: A Survey," Secur. Commun. Networks, vol. 2020, pp. 1–17, Aug. 2020, doi: 10.1155/2020/8872923.

[5] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, "Membership Inference Attacks on Machine Learning: A Survey," ACM Comput. Surv., vol. 54, no. 11s, pp. 1–37, Jan. 2022, doi: 10.1145/3523273.

[6] H. Alazzam, A. Sharieh, and K. E. Sabri, "A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer," Expert Syst. Appl., vol. 148, p. 113249, Jun. 2020, doi: 10.1016/j.eswa.2020.113249.

[7] et al. Rahim, Rahila, "nalysis of IDS using feature selection approach on NSL-KDD dataset," 2022.

[8] O. A. Alimi, K. Ouahada, A. M. Abu-Mahfouz, S. Rimer, and K. O. A. Alimi, "A Review of Research Works on Supervised Learning Algorithms for SCADA Intrusion Detection and Classification," Sustainability, vol. 13, no. 17, p. 9597, Aug. 2021, doi: 10.3390/su13179597.

[9] S. Rajagopal, P. P. Kundapur, and H. Katiganere Siddaramappa, "A predictive model for network intrusion detection using stacking approach," Int. J. Electr. Comput. Eng., vol. 10, no. 3, p. 2734, Jun. 2020, doi: 10.11591/ijece.v10i3.pp2734-2741.

[10] C. Ioannou and V. Vassiliou, "Network Attack Classification in IoT Using Support Vector Machines," J. Sens. Actuator Networks, vol. 10, no. 3, p. 58, Aug. 2021, doi: 10.3390/jsan10030058.

[11] et al. Ibitoye, Olakunle, "The Threat of Adversarial Attacks on Machine Learning in Network Security--A Survey," arXiv, vol. arXiv:1911, 2019.

[12] J. Kumar, R. Goomer, and A. K. Singh, "Long Short Term Memory Recurrent Neural Network (LSTM-RNN) Based Workload Forecasting Model For Cloud Datacenters," Procedia Comput. Sci., vol. 125, pp. 676–682, 2018, doi: 10.1016/j.procs.2017.12.087.

[13] E. Anthi, L. Williams, M. Rhode, P. Burnap, and A. Wedgbury, "Adversarial attacks on machine learning cybersecurity defences in Industrial Control Systems," J. Inf. Secur. Appl., vol. 58, p. 102717, May 2021, doi: 10.1016/j.jisa.2020.102717.

[14] U. Islam et al., "Detection of Distributed Denial of Service (DDoS) Attacks in IOT Based Monitoring System of Banking Sector Using Machine Learning Models," Sustainability, vol. 14, no. 14, p. 8374, Jul. 2022, doi: 10.3390/su14148374.

[15] and R. B. Kumar, Avinash, William Glisson, "Network attack detection using an unsupervised machine learning algorithm," Hawaii Int. Conf. Syst. Sci. 2020, 2020.

[16] et al. Lotfi, S., "Network intrusion detection with limited labeled data," arXiv, vol. 2209.03147, 2022.

[17] A. O. Sangodoyin, M. O. Akinsolu, P. Pillai, and V. Grout, "Detection and Classification of DDoS Flooding Attacks on Software-Defined Networks: A Case Study for the Application of Machine Learning," IEEE Access, vol. 9, pp. 122495–122508, 2021, doi: 10.1109/ACCESS.2021.3109490.

[18] F. Masoodi, "Machine learning for classification analysis of intrusion detection on NSL-KDD dataset," Turkish J. Comput. Math. Educ., vol. 12, no. 10, pp. 2286–2293, 2021, doi: https://doi.org/10.17762/turcomat.v12i10.4768.

[19] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Jul. 2009, pp. 1–6. doi: 10.1109/CISDA.2009.5356528.

[20] J. McHugh, "Testing Intrusion detection systems," ACM Trans. Inf. Syst. Secur., vol. 3, no. 4, pp. 262–294, Nov. 2000, doi: 10.1145/382912.382923.

[21] B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," in 2015 International Conference on Signal Processing and Communication Engineering Systems, Jan. 2015, pp. 92–96. doi: 10.1109/SPACES.2015.7058223.

[22] M. W. Liaw, Andy, "Classification and regression by randomForest," R news 2.3, pp. 8–22, 2002.

[23] K. Taunk, S. De, S. Verma, and A. Swetapadma, "A Brief Review of Nearest Neighbor Algorithm for Learning and Classification," in 2019 International Conference on Intelligent Computing and Control Systems (ICCS), May 2019, pp. 1255–1260. doi: 10.1109/ICCS45141.2019.9065747.

[24] and C.-J. L. Hsu, Chih-Wei, Chih-Chung Chang, "A practical guide to support vector classification," Taipei, Taiwan, pp. 1396–1400, 2003.

[25] A. M. Al-Hejri, R. M. Al-Tam, M. Fazea, A. H. Sable, S. Lee, and M. A. Al-antari, "ETECADx: Ensemble Self-Attention Transformer Encoder for Breast Cancer Diagnosis Using Full-Field Digital X-ray Breast Images," Diagnostics, vol. 13, no. 1, p. 89, Dec. 2022, doi: 10.3390/diagnostics13010089.

[26] S. Pandya et al., "Federated learning for smart cities: A comprehensive survey," Sustain. Energy Technol. Assessments, vol. 55, p. 102987, Feb. 2023, doi: 10.1016/j.seta.2022.102987.

[27] R. M. Al-Tam et al., "A Hybrid Workflow of Residual Convolutional Transformer Encoder for Breast Cancer Classification Using Digital X-ray Mammograms," Biomedicines, vol. 10, no. 11, p. 2971, Nov. 2022, doi: 10.3390/biomedicines10112971.