Enhancing the Intrusion Detection Efficiency using a Partitioning-based Recursive Feature Elimination in Big Cloud Environment

Hesham M. Elmasry[1], Ayman E. Khedr[2], Hatem M. Abdelkader[3]
Management Information Systems Department-Faculty of Commerce and Business Administration, Future University in Egypt (FUE), Cairo, Egypt[1]
Information Systems Department-Faculty of Computers and Information Technology, Future University in Egypt (FUE) Cairo, Egypt[2]
Information Systems Department-Faculty of Computers and Information, Menoufia University, Menoufia, Egypt[3]

***Abstract*—In the era of cloud computing, the effectiveness of utilizing supervised machine-learning-based intrusion detection models for categorizing and detecting malicious network attacks depends on the preparation, extraction, and selection of the optimal subset of features from the dataset. Therefore, before beginning the training phase of the machine learning classifier models, it is required to remove redundant data, manage missing values, extract statistical features from the dataset, and choose the most valuable and appropriate attributes using the Python Jupyter Notebook. In this study, partitioning-based recursive feature elimination (PRFE) method was suggested to decrease the complexity space and training time for machine learning models while increasing the accuracy rate of detecting malicious attacks. On the information security and object technology cloud intrusion dataset (ISOT-CID), some of the most popular supervised machine learning classification techniques, including support vector machines (SVM) and decision trees (DT), have been assessed using the suggested PRFE technique. In comparison to some of the most popular filter and wrapper-based feature selection strategies, the results of the practical experiments demonstrated an improvement in accuracy, recall, F-score, and precision rate after using the PRFE technique on the ISOT-CID dataset. Additionally, the time required to train the machine-learning models was reduced.**

Keywords*—Machine learning models; big cloud environment; intrusion detection system (IDS); Jupyter Notebook; feature selection; ISOT-CID introduction

I. Introduction

In the era of cloud computing and with the steady growth in the volume of transmitted and received data, machine learning models have emerged as one of the most significant contemporary techniques used to recognize and categorize dangerous assaults from network traffic. Preprocessing techniques on the data are therefore necessary in order to increase the precision and effectiveness of these models. A fundamental set of sub processes known as data preparation comprises steps including deleting duplicate data, filling in missing values, and turning some categorical data into numerical data so that machine learning models can interpret it [1], [2]. In a machine-learning process, incoming data is analyzed by computers to create patterns that foretell learning outcomes with a minimum of human input [3]. Based on how the learning algorithm is implemented, machine learning models can be divided into three groups.

The supervised machine learning (SML) model is used when the data available for the training phase is labelled, which means that some dataset attributes contain the correct answer that will be used at the end of the learning process to evaluate the final outputs. This model can be developed using either classification or regression algorithms [4], [5]. When dealing with a dataset that lacks labelled features, the unsupervised machine learning (UML) model is used and relied on; the model instead relies on trial and error to evaluate the learning process's outcomes. Moreover, this model can be developed using clustering algorithms [6]. While the reinforcement machine learning (RML) model evaluates the outcomes of the learning process based on the existence of an entity that performs a set of actions in a specific environment, a reward is given if the action matches the desired result. This model can be done with value-based, policy-based, and model-based algorithms [7].

One of the key elements influencing how well supervised machine learning models can detect and categorize harmful intrusions is feature engineering [8]. This can be done by selecting the dataset's most significant and connected features to the model outputs, a process known as feature selection, and then creating a new feature from the already accessible ISOT-CID dataset, a process known as feature extraction [9], [10].

In the feature selection phase, duplicated features and features that were not related to the outcomes of the learning process were excluded from the ISOT-CID dataset. The focus is only on the features that are most influential in building the detection model and are related to the results of the learning process, which would reduce the time required in the data training process and improve the quality of the outputs. The methods for selecting features can be divided into three types. In the filtering method, the degree of variance is calculated for each feature in the dataset, and higher or equal features are selected by the user based on a predetermined variance threshold [11]. One disadvantage of this method is that it does not consider the relationship between the selected features and target variables.

The wrapper technique uses a sophisticated search algorithm to analyze every feature combination in the dataset, then uses a machine-learning algorithm to evaluate the learning outcomes and choose the feature set that produces the best output. The high rate of classification accuracy for malicious attacks is one of this method's key benefits in terms of selecting the best features. The exorbitant expense and complexity of this technology are also disadvantages. Forward selection (FS), backward selection (BS), and recursive feature elimination (RFE) are three of the most significant algorithms utilized in this strategy [12]. The filter and wrapper methods' issues are addressed by the hybrid approach. There are two sections to the process. The features of the dataset were first created using a filtering technique. In the subsequent step, wrapper techniques were used to select the best features. Two of the most crucial algorithms used in this method are random forest importance (RFI) and LASSO regularization (LR) [13], [14].

The remainder of this paper is organized as follows: The literature review is summarized in Section II. Section III presents the research methodology and the five steps involved in this investigation. A detailed description of the ISOT-CID datasets is provided, along with information on the proposed partitioning-based recursive feature elimination (PRFE) technique, model flowchart, algorithm, performance metrics, research findings, and building ML classifier models. Section IV describes the discussion of results. The final Section V of this paper presents conclusions and suggestions for future research.

II. Literature Review

This section consists of two main parts. The first part focuses on reviewing and analyzing some of the previous work on supervised machine learning classifier algorithms, such as support vector machines, decision trees, naive Bayes, and k-nearest neighbor algorithms, and their improvements. In addition, the main limitations of each approach were identified. In the second part, we look back at some of the previous studies on feature selection techniques and analyze them. We explain the main improvements and limitations of each technique.

A. Machine Learning Classifier Algorithms

Without human interaction, intrusion detection systems can recognize new assaults using machine learning (ML). The IDS is able to modify its execution plan by using ML and taking into account recently acquired data. The two main categories of learning strategies are supervised and unsupervised strategies. In supervised learning, examples with input and output labels provided during training are used to "train" algorithms [15]. The unsupervised learning algorithms are allowed to make their own interpretations of the data because the training dataset is empty of labelled data. Unsupervised learning employs clustering and association algorithms to find patterns and distinctions in the data [16].

Peng et al. suggested using supervised machine-learning methods to categorize harmful attacks in a cloud environment and develop a decision tree-based model for intrusion detection. In order to guarantee the efficacy, excellence, and accuracy of the proposed models for categorizing hostile assaults, researchers have relied on a variety of preprocessing methods to prepare and clean enormous datasets. The suggested model was found to be more efficient and effective than naïve Bayesian and k-nearest neighbor models in laboratory trials using the Knowledge Discovery and Data Mining (KDDCUP99) dataset. The decision tree's training duration is not ideal, though, and this is its biggest drawback. Additionally, only the k-nearest neighbor and naive Bayesian models were contrasted with the decision tree model [17].

Using the Apache Spark machine learning library, Belouch et al. conducted a comparison study to assess the effectiveness and detection accuracy of support vector machines, random forests, decision, and naive Bayes algorithms (MLLIB). The results of the lab tests performed on the UNSW-NB15 dataset demonstrated the random forest algorithm's efficiency and effectiveness in comparison to other models. The greatest issue is that the model takes a long time to create and train because the feature selection technique isn't used [18].

Belavagi and Muniyal offered classification and predictive models for intrusion detection using machine learning classification techniques as logistic regression, support vector machine, naive Bayes, and random forest (RF). The techniques are evaluated using the Network Security Laboratory- Knowledge Discovery in Databases (NSL-KDD) dataset. The testing findings demonstrated that, with a peak value of 99%, the Random Forest classifier outperformed the other techniques in all criteria. However, the use of feature selection strategies to choose the best features from the dataset in order to minimize dimensionality is not examined in this paper [19].

An intrusion detection-based big data model was suggested by Azeroual and Nikiforova using unsupervised machine learning and the K-means clustering technique. The correlation-based filter method is used by the author to select the attributes that have the greatest impact on the results of the learning process. The Synchro Phasor dataset used in the laboratory tests revealed a high degree of classification accuracy for harmful attacks. However, the fundamental issue is that the lack of test support in the Apache Spark framework prevented the authors from comparing the suggested model to other solutions [20].

Souhail et al. suggested a two-stage network-based IDS (NIDS) technique to recognize network threats. The proposed approach combines LR, gradient boost machine (GBM), support vector machine (SVM), recursive feature elimination (RFE), and random forest feature selection methods for the complete UNSW-NB15 dataset. The results showed that the accuracy rate of multi-classifiers using decision trees was about 86.04%. Due to the usage of the recursive feature elimination-based feature selection technique, the key restriction is the amount of time needed to create and train the model [21].

A detection framework using an ML model was proposed by Alshammari and Aldribi to feed IDS and detect abnormal network traffic in cloud environments. An ISOT-CID dataset containing both malicious and normal traffic is used in this detection method. Six machine-learning models were trained

using this dataset, and they were then tested using split- and cross-validation techniques. Only two of the results were satisfactory, but the other four were accurate enough to be useful. The model's reliance on a large dataset or considerable dataset, which has an impact on how well the system is fitted and evaluated, is the biggest drawback, though [22].

B. Feature Selection Algorithms

Using evolutionary algorithms and support vector machines, Ashahri et al. suggested an embedding-based feature selection technique to minimize the number of dataset features from 45 to 10. The ten traits that had been chosen were then divided into three groups based on their level of significance in the following stage. Laboratory tests revealed that the suggested hybrid algorithm has a true-positive value of 0.973 and a false-positive value of 0.017 [23].

Based on 70% of the DDOS dataset from NSL-KDD, Mohammed and Gyasi suggested an intrusion detection system for distributed denial-of-service (DDOS). Random forest (RF) and multilayer perceptron (MLP) were utilized for the detection tasks, and recursive feature elimination (RFE) was used to choose the top 10 features. With receiver operating characteristic (ROC) ratings of 91% and 97%, respectively, their binary classification findings were precise. However, the accuracy and ROC score of our suggested binary classification were 99.86% and 99.7%, respectively. Furthermore, all of the assaults in the sample were found using our intrusion detection technology. However, due to the usage of the recursive feature elimination-based feature selection technique, the key restriction is the amount of time required to create and train the model. Additionally, a sizable dataset must be used to assess the proposed model's efficacy [24].

The stratified k-fold cross-validation (SKCV) method was proposed by Prusty et al. to improve classification accuracy by removing redundant and weak features whose deletion had the least impact on the training error while retaining an independent and strong feature to improve the generalization performance of the model and address the overfitting problem (RFE). This method creates a model with the whole set of features before prioritizing them based on relevance. The model was then rebuilt with the lowest priority feature deleted, and the feature importance estimate was revised. However, developing and refining the model using the SKCV method takes a lot of time [25].

For the NIDS methodology, Kasongo and Sun combined the filter-based feature selection technique of the XGBoost algorithm with five classification algorithms: logistic regression (LR), k-nearest neighbors, artificial neural network (ANN), decision tree, and support vector machines. This study uses binary and multiclass classification using the UNSW-NB15 dataset. Multiclass classification performed poorly, with the maximum accuracy being just 82.66%, while binary classification using the k-nearest neighbor classifier did well, with an accuracy of 96.76%. However, a sizable dataset must be used to assess the suggested model's efficacy. The classification model is unaffected by the filter-based strategy of selecting characteristics as well [26].

The intrusion detection model proposed by Thaseen and Kumar uses a multi-class SVM classifier and a rank-based chi-square feature selection technique. The chi-squared test can be used to determine the deviation from the predicted distribution when the feature event is thought to be independent of the class value. A multi-class SVM is used to categorize the various sorts of attacks in the NSL-KDD dataset. Using the proposed model, 31 features were selected from a total of 41. The accuracy rate of the suggested system was 98%, while the false positive rate was 0.13% [27].

III. Proposed Methodology

The methodology of this experimental study consists of five stages. First, the flow features were extracted using a CIC flow meter tool. In the second stage, dataset preprocessing was performed, and in the third stage, the best subset features were selected using the proposed PRFE technique. In the fourth step, a machine learning detection model is made. Finally, using the proposed PRFE technique and the ISOT-CID dataset, some supervised machine learning techniques are tested. Fig. 1 illustrates the five phases of the experimental study, which will be discussed in the remainder of this research.

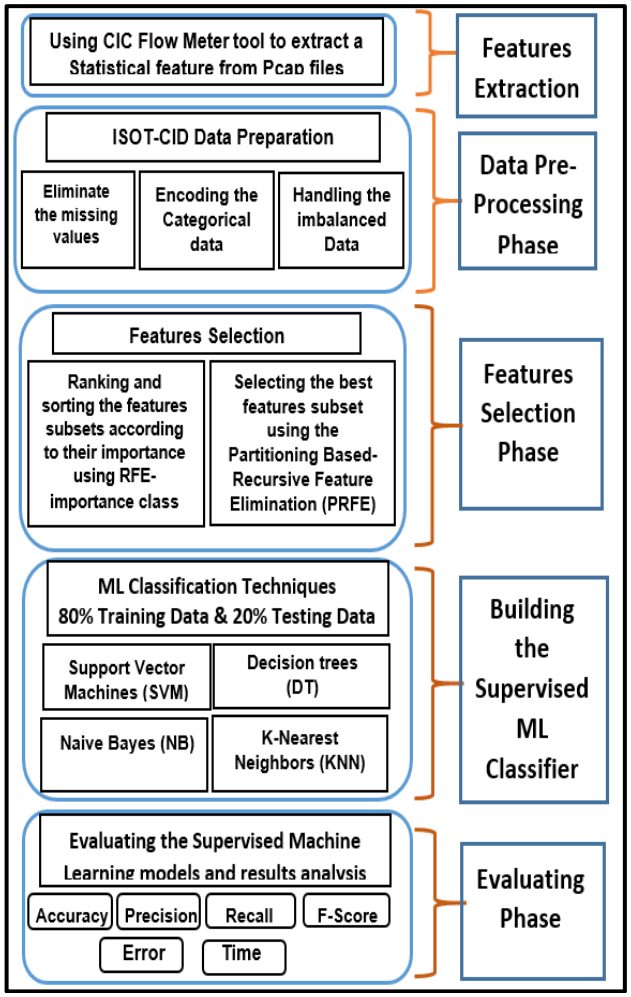


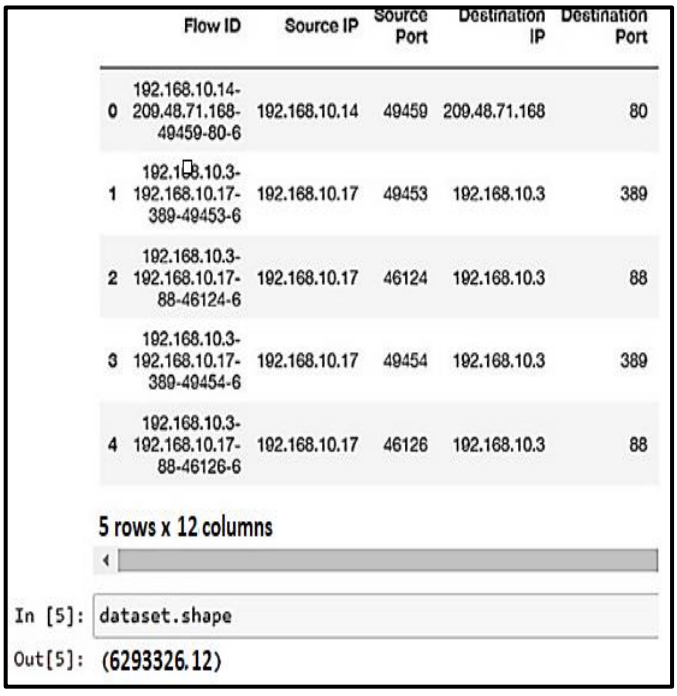
Fig. 1. Research methodology phases.

A. Features Extraction

As shown in Fig. 2, the original ISOT-CID dataset consists of 12 attributes and 6.293.326 records [28]. In this study, a CIC flow meter was used to extract statistical and analytical features from the network flow. The CIC flow meter is a Java open-source tool that can generate and extract row attributes from huge Packet Capture (pcap) files and save the results in the comma-separated value (CSV) file format. Eighty-five network flow attributes were extracted from the ISOT-CID dataset using the CIC Flow Meter tool. The list of extracted features will be reduced to 80 after eliminating the five features that contained more than 90% missing data. Then, the proposed PRFE-based feature selection technique is used to choose the best, most important and most influential features of the learning process output [29]. Fig. 3 illustrates the number of features and records in the ISOT-CID dataset after the feature extraction stage.

B. Dataset Preprocessing

The dataset used in the actual experiments in this study was ISOT-CID, which is considered the first huge, public, and labelled cloud intrusion detection dataset. The size of the ISOT-CID dataset is greater than 2.5 TB, and it consists of normal and malicious traffic activity collected from different cloud tiers, virtual machine hosts, and hypervisors. ISOT-CID data is collected in two phases and consists of different data formats, such as network traffic, CPU utilization, memory dumps, and event logs. The dataset consists of several types of attacks, including remote to local (R2L), input validation, backdoors, Denial of Service (DOS) and probing.

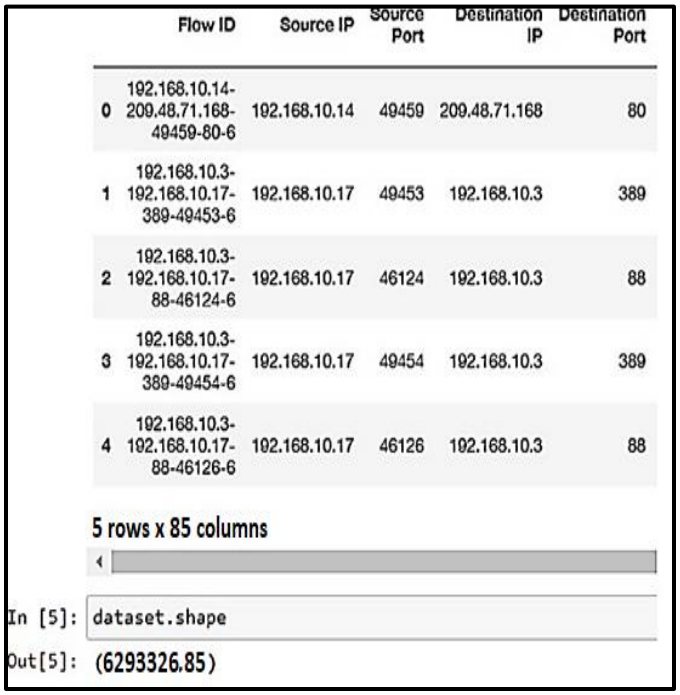


	Flow ID	Source IP	Source Port	Destination IP	Destination Port
0	192.168.10.14-209.48.71.168-49459-80-6	192.168.10.14	49459	209.48.71.168	80
1	192.168.10.3-192.168.10.17-389-49453-6	192.168.10.17	49453	192.168.10.3	389
2	192.168.10.3-192.168.10.17-88-46124-6	192.168.10.17	46124	192.168.10.3	88
3	192.168.10.3-192.168.10.17-389-49454-6	192.168.10.17	49454	192.168.10.3	389
4	192.168.10.3-192.168.10.17-88-46126-6	192.168.10.17	46126	192.168.10.3	88

5 rows x 12 columns

```
In [5]: dataset.shape
Out[5]: (6293326.12)
```

Fig. 2. ISOT-CID dataset before features extraction.



	Flow ID	Source IP	Source Port	Destination IP	Destination Port
0	192.168.10.14-209.48.71.168-49459-80-6	192.168.10.14	49459	209.48.71.168	80
1	192.168.10.3-192.168.10.17-389-49453-6	192.168.10.17	49453	192.168.10.3	389
2	192.168.10.3-192.168.10.17-88-46124-6	192.168.10.17	46124	192.168.10.3	88
3	192.168.10.3-192.168.10.17-389-49454-6	192.168.10.17	49454	192.168.10.3	389
4	192.168.10.3-192.168.10.17-88-46126-6	192.168.10.17	46126	192.168.10.3	88

5 rows x 85 columns

```
In [5]: dataset.shape
Out[5]: (6293326.85)
```

Fig. 3. ISOT-CID dataset after features extraction.

1) Eliminate the missing values: The dataset always needs to be reprocessed to remove duplicate and missing data before being used in the training phase of machine learning models, because relying on this dataset without processing would affect the quality of the learning results. A Python Jupyter Notebook was used to process and eliminate missing values in the ISOT-CID dataset. As shown in Fig. 4, some values in the flag, protocol, and fragment columns are missing.

```
In [7]: # check for the Missing values

In [8]: dataset.isnull().sum()

Out[8]: timestamp            0
        protocol          1455
        sourceIp             0
        sourcePort           0
        destIp               0
        destPort             0
        size                 0
        fragmented         950
        seqNumber            0
        ackNumber            0
        flags             3713
        classification       0
        dtype: int64
```

Fig. 4. ISOT-CID before handling missing values.

```
In [11]: dataset.dropna(inplace=True)
         dataset.isnull().sum()

Out[11]: timestamp          0
         protocol           0
         sourceIp           0
         sourcePort         0
         destIp             0
         destPort           0
         size               0
         fragmented         0
         seqNumber          0
         ackNumber          0
         flags              0
         classification     0
         dtype: int64
```

Fig. 5. ISOT-CID after handling missing values.

Fig. 5 shows the ISOT-CID dataset after the missing values were taken care of by dropping them using the Python Jupyter Notebook. This improved the machine learning model's ability to classify and find malicious traffic.

2) Label encoding for categorical data: Dealing with machine learning algorithms to detect and classify malicious assaults in a big cloud environment requires encoding and converting some of the textual data that exists in the dataset into digital and numeric data to enhance and increase the level of accuracy of the learning results. As shown in Fig. 6, categorical attributes were converted to (0, 1) instead of (benign or malicious) attributes using the label encoding method.

```
In [11]: dummy = pd.get_dummies(dataset['classification'])

In [12]: dummy.head()

Out[12]:
```

	benign	malicious
0	1	0
1	1	0
2	1	0
3	1	0
4	1	0

Fig. 6. Encoding the categorical into numeric data.

3) Handling the imbalanced-labelled data: Class imbalance is a machine learning issue where the classes are not evenly represented in the data. This can cause issues while training machine learning models because the models may be biased towards the more prevalent class. The model will be more likely to pick up on and predict the majority class if there are more samples of one class than the other. As a result, when the model is used to analyze data that is more evenly distributed, it may produce erroneous conclusions. Addressing unbalanced classification difficulties is a challenge in developing models with good performance. Fig. 7 illustrates the number of benign and malicious objects in the classification class before handling the imbalanced labelled data using the oversampling technique.

Synthetic Minority Oversampling Technique (SMOTE) is a method of oversampling, in which artificial samples are produced for the minority class. This method solves the overfitting problem caused by random oversampling. By interpolating nearby positive examples, we focused on the feature space to create new examples. Fig. 8 illustrates the number of benign and malicious objects in the classification class after handling the imbalanced labelled data.

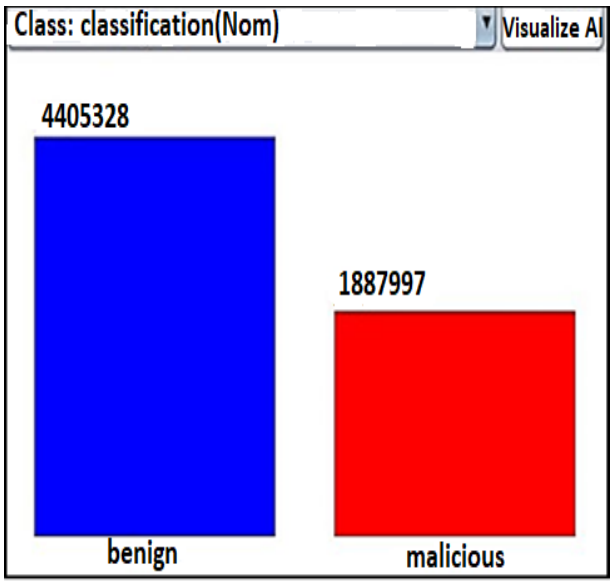


Fig. 7. Classification attribute before using SMOT.

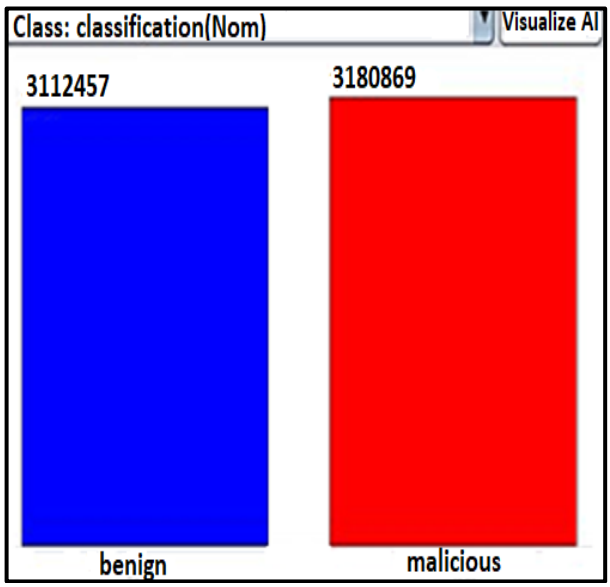


Fig. 8. Classification attribute after using SMOT.

C. Proposed Features Selection Algorithm (PRFE)

Feature selection is a technique used to improve the accuracy of malicious attack classification by selecting the attributes that are the most important and significant to the outcome of the learning process and eliminating the least informative attributes. In this study, the ISOT-CID dataset was used to find the most useful attributes. To deal with labelled datasets, a supervised-based feature selection method was used [30]. Recursive feature elimination is a wrapper-based feature selection approach that assesses the significance of features using a machine-learning algorithm. All dataset features were used to train and fit the ML classifier model in the initial stages of recursive feature elimination, and the feature importance was calculated for each feature. The recursive feature elimination model is used repeatedly, with the least important features being thrown out and the most important ones being saved for the next round, until the best features are found. The partitioning-based recursive feature elimination (PRFE) technique was proposed in this study to improve the accuracy rate of classifying and detecting malicious attacks while reducing the complexity space and time required training the ML classifier models. Algorithm 1 illustrates the Partitioning-based Recursive Feature Elimination (PRFE) algorithm for selecting optimal features from the ISOT-CID dataset.

Algorithm 1: Partitioning based Recursive Feature Elimination (PRFE)

#Input

F: set of features where F= {f1, f2 … f n}

N: number of the required features

G: the number of groups where: 1<G≤ N

i = 1

#Output

O: ordered ranked features.

R: ordered ranked groups.

Step 1: Train the model using all attributes

Step 2: Compute the model's accuracy

Step 3: Calculate and ranking the feature importance using the RFE-impotence class Fi^{rank} where i = 1...N (N is the number of features).

Step 4: Divided the feature into equal number of groups (G), where numbers of features in each group are equal.

Note: G[i] contains number of features.

Step 5: Ranking and sorting the groups (G) in ascending order based on their features weight.

Step 6: Eliminating the lowest weighted group.

Step 7: Build ML classifier model and calculate model performance.

Step 8: i = i + 1

Step 9: If i! = G-1

Repeat step 4 to step 7 until i = G-1

End

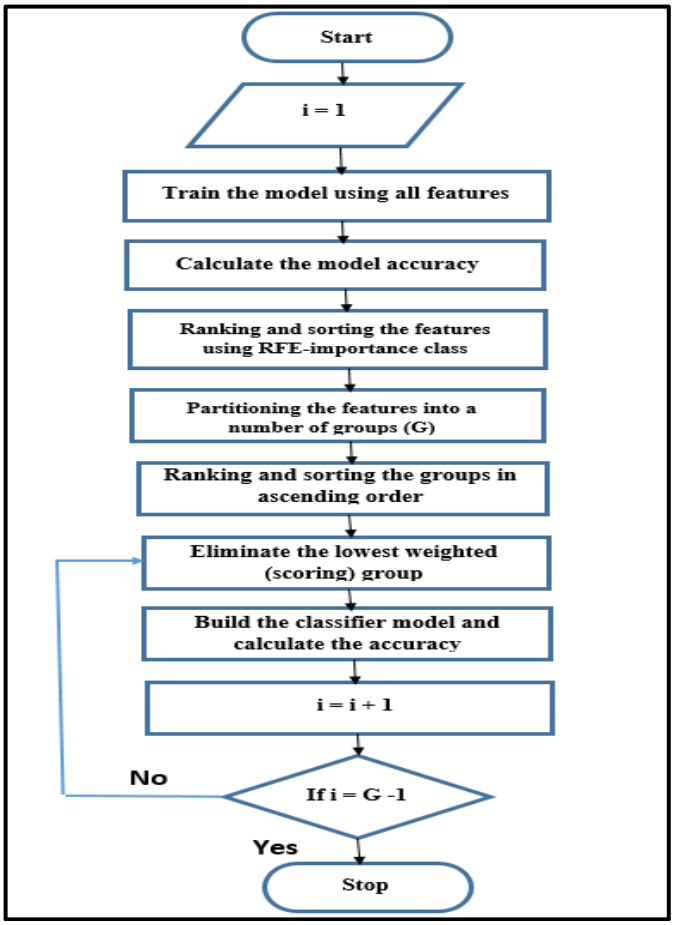


Fig. 9. Partitioning based Recursive Feature Elimination (PRFE) flowchart.

As shown in Fig. 9, the process of selecting optimal features from the ISOT-CID dataset using the PRFE technique occurs in two stages. In the first stage, the RFE importance class was used to rank and sort the feature subsets individually according to their importance and how strongly they were related to the outcomes of the learning process. In the second stage, the features are partitioned into groups, with an equal number of features in each group. For example, if we have 100 attributes, we can divide them into ten groups, each with ten attributes. Subsequently, the groups are ranked and sorted in ascending order, the lowest weighted group is eliminated in each iteration, and the training procedure for the remaining groups is repeated to obtain the best group of features. So, the number of tests went from 100 to 10, and the space and time needed to train the machine learning models became less complicated.

D. Building the Machine Learning Classifier Models

For supervised learning, the ISOT-CID dataset was divided into training and testing sets. As a result, 80 % of the data were chosen at random and used to train machine-learning models, whereas the remaining 20% were utilized to evaluate the classifier's performance. Table I illustrates the statistics of the ISOT-CID Dataset used in this study.

TABLE I. STATISTICS OF THE ISOT-CID DATASET

Traffic Type	Total	Training 80%	Testing 20%
benign	3112457	2489965.6	622491.4
malicious	3180869	2544695.2	636173.8
Total	6293326	5034660.8	1258665.2

To evaluate the performance of the machine learning classifier models with different sets of selected features, accuracy, precision, recall, F-score, and error performance measurements were utilized. A confusion matrix was used to calculate the classifier performance indicators. "True positive" (TP) denotes benign instances that are correctly predicted, true negative (TN) denotes malicious instances that are correctly identified, false positive (FP) denotes malicious instances that are incorrectly assumed to be normal, and false negative (FN) denotes malicious instances that are incorrectly detected as normal [28]. Table II illustrates the five metrics that are commonly used to measure and evaluate the effectiveness of machine learning classification models.

TABLE II. PERFORMANCE METRICS FOR ML CLASSIFICATION MODELS

Metric	Formula	Interpretation
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Overall performance of model
Precision	$\frac{TP}{TP+FP}$	How accurate the positive predictions are
Recall Sensitivity	$\frac{TN}{TN+FP}$	Coverage of actual positive sample
F1 score	$\frac{2TP}{2TP + FP+FN}$	Hybrid metric useful for unbalanced classes
Error Rate	$\frac{FP+FN}{TP+TN+FP+FN}$	the percentage of the classification that is done wrongly

E. Experiment Findings and Analysis

1) The experiment setup: Using the HP Z Book G3 workstation with Microsoft Windows 11 64-bit Enterprise edition and an Intel Core i7-6820HQ CPU @ 2.7GHz, 32GB RAM, the novel proposed PRFE approach was created using the Python version 3 code, which was implemented using the Jupyter Notebook platform and Anaconda virtual environment for Windows to execute Scikit-learn, NumPy, and Panda's libraries.

2) The experiment findings: Accuracy is one of the most important performance metrics in intrusion detection. The accuracy of the four supervised machine-learning classifiers using the proposed PRFE method outperformed the RFECV and RFE techniques in terms of overall performance. When PRFE-based selected features were used instead of RFECV- and RFE-based selected features, accuracy improved by approximately 0.75% and 2.25%, respectively. As shown in Fig. 10, with PRFE-based selected features, the Support Vector Machine (SVM) classifier achieved the highest accuracy percentage of 99.25%. The k-nearest neighbor (KNN) classifier performed the worst in this trial. In general, the four machine-learning classifier models were more accurate after they used PRFE-based feature selection.

As shown in Fig. 11, the precision findings, which demonstrate the classifier's percentage of accurately identified instances, which is one of the key markers of excellent models. Classifiers trained with PRFE-based selected features outperformed those trained with RFECV and RFE-based selected features. When compared to other classifiers, the support vector machine (SVM) has the highest precision percentage of 98.80%. In contrast to prior results, the k-nearest neighbor (KNN) classifier has the lowest precision percentage in this trial, with a value of 96.50%. In general, the four machine-learning classifier models were more accurate when they used the PRFE-based feature selection method.

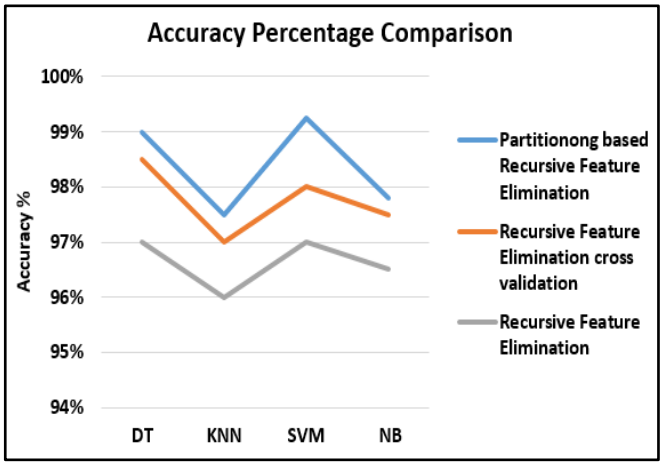


Fig. 10. Accuracy percentage comparison among PRFE, RFECV and RFE algorithm.

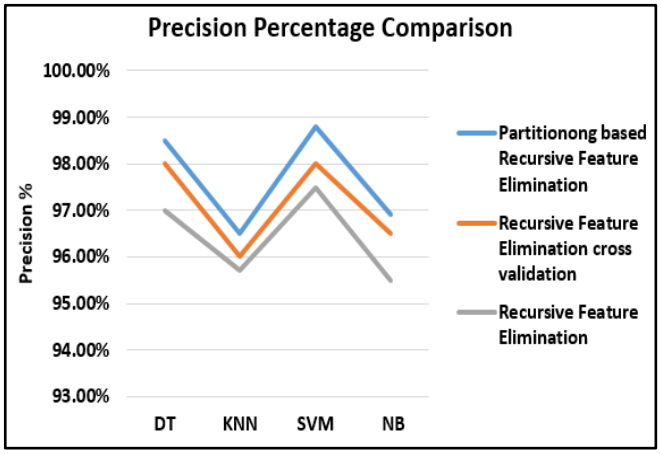


Fig. 11. Precision percentage comparison among PRFE, RFECV and RFE algorithm.

Fig. 12 shows the recall and sensitivity rates of the prediction models. Classifiers trained with PRFE-based selected features outperformed those trained with RFECV and RFE-based selected features. When compared to other classifiers, the decision tree classifier had the highest precision

percentage of 99%. In this experiment, nave Bayes exhibited the lowest precision percentage (97%). In general, the recall rate of the four machine learning classifier models got better when they used the PRFE-based feature selection method.

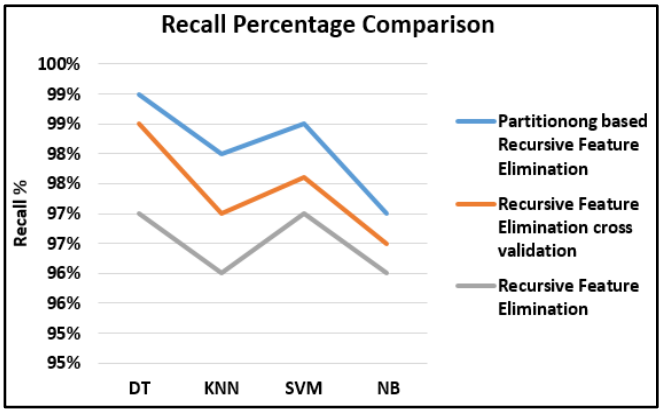


Fig. 12. Recall percentage comparison among PRFE, RFECV and RFE algorithm.

Fig. 13 shows the f-score rate of the predictive models. In general, when PRFE-based selected features were used, the classifiers' f-scores performed better than when RFECV- and RFE-based selected features were used. With PRFE's chosen features, the support vector machine (SVM) classifier achieved the greatest f-score percentage of 99%. The k-nearest neighbor (KNN) classifier performed the worst in this trial with 97.5% (Table III).

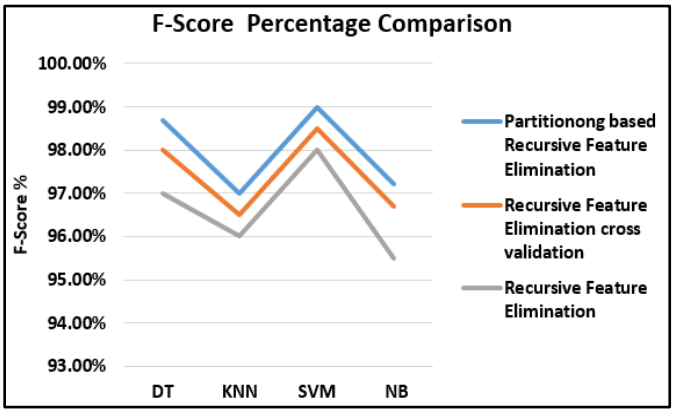


Fig. 13. F-Score percentage comparison among PRFE, RFECV and RFE algorithm.

TABLE III. OBSERVED ANALYSIS FOR DIFFERENT FEATURE SELECTION TECHNIQUES IN ISOT-CID USING ML CLASSIFIERS

Feature Selection Techniques	ML Algorithms	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	Error (%)	Time (m)
PRFE	DT	99%	98.50%	99%	98.70%	1%	2.33
	KNN	97.50%	96.50%	98.00%	97%	2.50%	1.55
	SVM	99.25%	98.80%	98.50%	99%	0.75%	1.12
	NB	97.80%	96.90%	97%	97.20%	1%	2.88
RFECV	DT	98.50%	98%	98.50%	98%	1.50%	6.88
	KNN	97%	96%	97%	96.50%	3%	5.65
	SVM	98%	98%	97.60%	98.50%	2%	4.71
	NB	97.50%	96.50%	97%	96.70%	2.50%	5.77
RFE	DT	97%	97%	97%	97%	3%	8.71
	KNN	95.50%	95.70%	96%	96.80%	4.50%	7.32
	SVM	96.50%	97.50%	97%	98%	3.50%	6.71
	NB	96.00%	96.00%	96.50%	96.00%	4.00%	7.77
Chi square	DT	97%	97.50%	96.80%	97%	3%	2.15
	KNN	96.50%	95%	94%	95.50%	3.50%	1.78
	SVM	98.50%	97%	95%	98%	1.50%	1.55
	NB	96%	95.80%	94.50%	96.40%	4%	2.45
Information Gain	DT	98%	96.50%	97.50%	96.50%	2%	1.85
	KNN	97%	96%	95%	96%	3%	1.25
	SVM	97.50%	96%	94.50%	97%	2.5%	1.03
	NB	96.50%	95%	94%	95.50%	3.50%	2.12
Backward Feature Elimination	DT	96.50%	98%	97.80%	97.50%	3.50%	5.44
	KNN	96%	95.50%	97%	96.50%	4%	4.65
	SVM	97.50%	96.50%	96%	96.70%	2.50%	4.33
	NB	96%	96.50%	95.40%	95%	4%	5.12

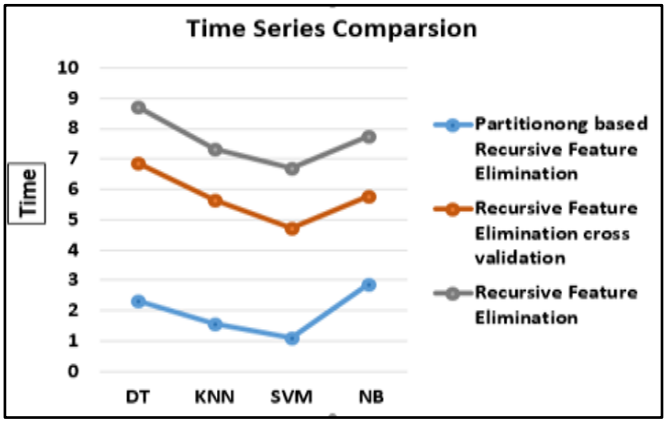


Fig. 14. Time series comparison among PRFE, RFECV and RFE algorithm.

Fig. 14 indicates a decrease in the time required to train the model using the proposed PRFE method compared to RFECV and RFE techniques.

IV. Discussion of Results

The ISOT-CID dataset was used for the evaluation experiments, which were carried out using Python programming and the Jupyter Notebook. In the evaluation experiment, the proposed PRFE was compared with RFE- and RFECV-based feature selection techniques. The comparison with the other approaches is defined in terms of accuracy, precision, recall, F-score, and time series. The evaluation trial demonstrates PRFE's advantages over competitors. Therefore, in this section, we will discuss how it stacks up against more modern research methods.

The traditional methods for selecting the most important features in the previous studies were characterized by simplicity and the low time required training machine learning models, while suffering from a low level of accuracy in identifying and detecting malicious attacks. Modern methods focus on raising the level of accuracy while neglecting the time required training machine learning models. Therefore, the proposed PFRE technique enhances the accuracy rate by combining supervised machine learning classifier models with partitioning-based recursive feature elimination techniques. This led to an increase in the level of accuracy in identifying and detecting malicious attacks to 99.25, while simultaneously reducing the time required to train automated training models to 1.2 minutes. Because this study deals with the cloud-based intrusion detection system (CIDS), the proposed PRFE technique was evaluated using the ISOT-CID dataset, which is considered one of the first public datasets of its kind collected from a production cloud environment. Most previous studies lacked datasets from a real cloud computing environment that were available to the public. This made it hard to make and test realistic detection models.

V. Conclusion and Future Works

Feature engineering techniques such as data preprocessing, feature extraction, and feature selection should be used to reduce the dimensionality of the input features, improve model performance, and shorten model computational time. Choosing the most relevant and influential characteristics has traditionally affected the power and predictability of the final classifier model. The most pertinent and effective features were chosen from large datasets using a variety of feature selection techniques, including chi-square, information gain, backward feature elimination, and recursive feature elimination. In this study, a PRFE-based feature selection method was developed to classify and select the optimal feature subset from ISOT-CID, which is considered one of the largest public intrusion detection datasets. The best feature subset was selected by partitioning the features into groups with an equal number of features in each group and eliminating the lowest-weighted group in each iteration. The PRFE method improved accuracy, precision, recall, and F-score rate while cutting training time by ignoring one group instead of removing one feature at each iteration.

In this experimental study, the proposed PRFE-based feature selection technique was first compared with recursive feature elimination (RFE) and recursive feature elimination with cross-validation (RFECV) techniques. The results showed that the proposed PRFE technique improved the accuracy, precision, recall, and F-score percentage with the four common machine learning classifier models compared to the RFE and RFECV techniques. Second, a few popular filter- and wrapper-based feature selection methods, including chi-square, information gain, and backward feature removal, are compared to the proposed PRFE-based feature selection strategy. The results of the experiments show that when the PRFE method is used with four popular machine learning classifier models, the accuracy, precision, recall, and F-score percentage are all higher than when the chi-square, information gain, and backward feature elimination strategies are used. In future work, a new unsupervised deep learning algorithm for detecting zero-day attacks will be proposed. This algorithm will use different neural network topologies, such as fully connected, recurrent, and temporal convolutional models, to reduce the number of false alarms while maintaining the accuracy of detecting and classifying malicious network attacks.

References

[1] Butt, U., Mehmood, M., Shah, S., Amin, R., Shaukat, M., & Raza, S. et al. (2020). A Review of Machine Learning Algorithms for Cloud Computing Security. Electronics, 9(9), 1379. https://doi.org/10.3390/electronics9091379.

[2] Xu, Z., Liu, Y., Yen, N., Mei, L., Luo, X., Wei, X., & Hu, C. (2020). Crowdsourcing Based Description of Urban Emergency Events Using social media Big Data. IEEE Transactions on Cloud Computing, 8(2), 387-397. https://doi.org/10.1109/tcc.2016.2517638.

[3] Sarker, I., Kayes, A., Badsha, S., Alqahtani, H., Watters, P., & Ng, A. (2020). Cybersecurity data science: an overview from machine learning perspective. Journal of Big Data, 7(1). https://doi.org/10.1186/s40537-020-00318-5.

[4] Megantara, A., & Ahmad, T. (2021). A hybrid machine learning method for increasing the performance of network intrusion detection systems. Journal of Big Data, 8(1). https://doi.org/10.1186/s40537-021-00531-w.

[5] Mebawondu, J., Alowolodu, O., Mebawondu, J., & Adetunmbi, A. (2020). Network intrusion detection system using supervised learning paradigm. Scientific African, 9, e00497. https://doi.org/10.1016/j.sciaf.2020.e00497.

[6] Jadhav, A., & Pellakuri, V. (2021). Highly accurate and efficient two phase-intrusion detection system (TP-IDS) using distributed processing of HADOOP and machine learning techniques. Journal of Big Data, 8(1). https://doi.org/10.1186/s40537-021-00521-y.

[7] Taylor, O., & Ezekiel, P. (2021). Anomaly-Based Intrusion Detection/Prevention System using Deep Reinforcement Learning Algorithm. IJARCCE, 10(1). https://doi.org/10.17148/ijarcce.2021.10114.

[8] Panda, M., Mousa, A., & Hassanien, A. (2021). Developing an Efficient Feature Engineering and Machine Learning Model for Detecting IoT-Botnet Cyber Attacks. IEEE Access, 9, 91038-91052. https://doi.org/10.1109/access.2021.3092054.

[9] Krishnaveni, S., Sivamohan, S., Sridhar, S., & Prabakaran, S. (2021). Efficient feature selection and classification through ensemble method for network intrusion detection on cloud computing. Cluster Computing, 24(3), 1761-1779. https://doi.org/10.1007/s10586-020-03222-y.

[10] Umar, M., Chen, Z., & Liu, Y. (2021). A Hybrid Intrusion Detection with Decision Tree for Feature Selection. Information &Amp; Security: An International Journal. https://doi.org/10.11610/isij.4901.

[11] Kadhum, M., Manaseer, S., & Dalhoum, A. (2021). Evaluation Feature Selection Technique on Classification by Using Evolutionary ELM Wrapper Method with Features Priorities. Journal of Advances in Information Technology, 12(1), 21-28. https://doi.org/10.12720/jait.12.1.21-28.

[12] Viharos, Z., Kis, K., Fodor, Á., & Büki, M. (2021). Adaptive, Hybrid Feature Selection (AHFS). Pattern Recognition, 116, 107932. https://doi.org/10.1016/j.patcog.2021.107932.

[13] Xiao, W., Ji, P., & Hu, J. (2021). RnkHEU: A Hybrid Feature Selection Method for Predicting Students' Performance. Scientific Programming, 2021, 1-16. https://doi.org/10.1155/2021/1670593.

[14] Das, A., -, P., & S, S. (2022). Anomaly-based Network Intrusion Detection using Ensemble Machine Learning Approach. International Journal of Advanced Computer Science and Applications, 13(2). https://doi.org/10.14569/ijacsa.2022.0130275.

[15] Dhanda, N., Datta, S., & Dhanda, M. (2019). Machine Learning Algorithms. Computational Intelligence In The Internet of Things, 210-233. https://doi.org/10.4018/978-1-5225-7955-7.ch009.

[16] Gaydamaka, K., & Belonogova, A. (2022). Applying Unsupervised Machine Learning Algorithms to Ensure Requirements Consistency. Programmnaya Ingeneria, 13(4), 187-199. https://doi.org/10.17587/prin.13.187-199.

[17] Peng, K., Leung, V., Zheng, L., Wang, S., Huang, C., & Lin, T. (2018). Intrusion Detection System Based on Decision Tree over Big Data in Fog Environment. Wireless Communications and Mobile Computing, 2018, 1-10. https://doi.org/10.1155/2018/4680867.

[18] Belouch, M., El Hadaj, S., & Idhammad, M. (2018). Performance evaluation of intrusion detection based on machine learning using Apache Spark. Procedia Computer Science, 127, 1-6. https://doi.org/10.1016/j.procs.2018.01.091.

[19] Belavagi, M., & Muniyal, B. (2016). Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection. Procedia Computer Science, 89, 117-123. https://doi.org/10.1016/j.procs.2016.06.016.

[20] Azeroual, O., & Nikiforova, A. (2022). Apache Spark and MLlib-Based Intrusion Detection System or How the Big Data Technologies Can Secure the Data. Information, 13(2), 58. https://doi.org/10.3390/info13020058.

[21] Souhail et. al., M. (2019). Network Based Intrusion Detection Using the UNSW-NB15 Dataset. International Journal of Computing and Digital Systems, 8(5), 477-487. https://doi.org/10.12785/ijcds/080505.

[22] Alshammari, A., & Aldribi, A. (2021). Apply machine learning techniques to detect malicious network traffic in cloud computing. Journal of Big Data, 8(1). https://doi.org/10.1186/s40537-021-00475-1.

[23] Aslahi-Shahri, B., Rahmani, R., Chizari, M., Maralani, A., Eslami, M., Golkar, M., & Ebrahimi, A. (2015). A hybrid method consisting of GA and SVM for intrusion detection system. Neural Computing and Applications, 27(6), 1669-1676. https://doi.org/10.1007/s00521-015-1964-2.

[24] Mohammed, B., & Gbashi, E. (2021). Intrusion Detection System for NSL-KDD Dataset Based on Deep Learning and Recursive Feature Elimination. Engineering and Technology Journal, 39(7), 1069-1079. https://doi.org/10.30684/etj.v39i7.1695.

[25] Prusty, S., Patnaik, S., & Dash, S. (2022). SKCV: Stratified K-fold cross-validation on ML classifiers for predicting cervical cancer. Frontiers in Nanotechnology, 4. https://doi.org/10.3389/fnano.2022.972421.

[26] Kasongo, S., & Sun, Y. (2020). Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset. Journal of Big Data, 7(1). https://doi.org/10.1186/s40537-020-00379-6.

[27] Sumaiya Thaseen, I., & Aswani Kumar, C. (2017). Intrusion detection model using fusion of chi-square feature selection and multi class SVM. Journal of King Saud University - Computer and Information Sciences, 29(4), 462-472. https://doi.org/10.1016/j.jksuci.2015.12.004.

[28] Aldribi, A., Traoré, I., Moa, B., & Nwamuo, O. (2020). Hypervisor-based cloud intrusion detection through online multivariate statistical change tracking. Computers &Amp; Security, 88, 101646. https://doi.org/10.1016/j.cose.2019.101646.

[29] Habibi Lashkari, A., Draper Gil, G., Mamun, M., & Ghorbani, A. (2017). Characterization of Tor Traffic using Time based Features. Proceedings of the 3Rd International Conference on Information Systems Security and Privacy. https://doi.org/10.5220/0006105602530262.

[30] Alabdulwahab, S., & Moon, B. (2020). Feature Selection Methods Simultaneously Improve the Detection Accuracy and Model Building Time of Machine Learning Classifiers. Symmetry, 12(9), 1424. https://doi.org/10.3390/sym12091424.