Enhancing Intrusion Detection System Performance through Feature Selection

Salem-Bilal Amokrane, Dimitrije Bujaković, Boban Pavlović, Milenko Andrić, Touati Adli

University of Defence in Belgrade, Military Academy, Veljka Lukića Kurjaka 33, 11000 Belgrade, Serbia; salembilal.amokrane@va.mod.gov.rs; dimitrije.bujakovic@va.mod.gov.rs; boban.pavlovic@va.mod.gov.rs; milenko.andric@va.mod.gov.rs; touati.adli@va.mod.gov.rs

Abstract: Network intrusion detection systems are critical for identifying anomalous activities and cyberthreats. The anomaly detection method for network intrusion detection systems has become substantial in detecting novel attacks in intrusion detection systems. Achieving high accuracy with the lowest false alarm rate is a significant challenge in designing an intrusion detection system. Network intrusion detection systems based on machine learning methods are effective and accurate in detecting network attacks. It also highlights the importance of using various feature selection techniques to identify the optimal subset of features. This paper investigates enhancing network intrusion detection systems performance through correlation analysis and feature selection on the part of the NF-UQ-NIDS-v2 NetFlow dataset that will be used for training and testing our models. In our experiments, binary classification configurations were considered. Two approaches are explored: applying feature selection methods directly to the initial 39 features set, and performing correlation analysis to eliminate redundant features then applying feature selection methods. Recursive feature elimination, mutual information, and One-way ANOVA methods select optimized feature subsets. An ExtraTrees ensemble classifier performs binary classification of benign and traffic under attack. Results indicate that employing Recursive feature elimination on 8 features after performing correlation analysis yields the most promising outcomes. It achieves a high detection accuracy of 98.13%, recall of 98.23%, and Area Under Curve of 99.73%. Notably, it substantially reduces the false alarm rate by 53.73% compared to using all 39 features bringing it to 0.3589%, and decreases the scoring time by 34.21%, resulting in an efficient scoring time.

Keywords: Network intrusion detection systems; Machine learning; Feature selection; Classification

1 Introduction

As communication networks rapidly evolve, there has been a significant increase in the frequency of network attacks. The attack causes a threat to the confidentiality, integrity, or availability of an information system [1]. Anomaly detection methods are crucial in enhancing network security by identifying deviations from traffic patterns, especially in cases where signature-based detection is ineffective against new attacks. This capability makes anomaly detection invaluable for communication networks, as it serves as the foundation for uncovering various types of attacks, identifying misconfigurations, and detecting network failures [2]. To protect networks against advanced cyberattacks, a combination of anomaly-based Network Intrusion Detection Systems (NIDS) and NetFlow analysis is crucial. Anomaly-based NIDS detects known and unknown attacks by analyzing network traffic patterns, while NetFlow, an industry-standard network protocol, offers features that enable accurate prediction of malicious cyberattacks by analyzing network traffic patterns and behaviors [3, 4]. As NetFlow data collection capabilities are widely available on common network infrastructure devices like routers and switches. It becomes crucial to assess the effectiveness of utilizing NetFlow features to detect potential attacks by analyzing the extracted flow records. Incorporating Machine Learning (ML) can significantly enhance the capabilities of NIDS in adapting to evolving attack patterns. Intrusion Detection Systems (IDS) use signature-based techniques to identify threats by matching known attack patterns. However, this method requires constantly updating the database of known attack signatures since intruders consistently find new ways to exploit network activities [5].

The ML applications have enabled new approaches to anomaly detection, allowing systems to identify previously unknown attacks by comparing normal user behavior patterns against events that diverge from those norms. In recent years, researchers have explored various ML techniques to enhance the performance of IDS, aiming to improve detection rates, reduce false positives, and increase overall predictive accuracy. By leveraging ML, IDS can automatically learn models of benign activity and use those models to detect suspicious anomalies that may represent malicious actions. The application of ML to anomaly detection remains an active area of research as new algorithms and methodologies emerge further to improve the capabilities of these critical security systems. Due to resource constraints for data storage, transmission, and processing, limiting input data to features highly relevant to the detection task is advantageous and easily derivable from network observations without costly operations. Enhancing detection quality for various learning-based algorithms is achieved by eliminating strongly correlated, redundant, and irrelevant features [2]. This will involve a Feature Selection (FS) method as a good technique to lead these aspects by selecting the most informative features from the network traffic data. These methods improve detection accuracy, reduce false positives, and optimize computational resources, resulting in more efficient threat detection. Eliminating uninformative features enables IDS to focus modeling efforts on the most salient aspects of network traffic. The insights from removing extraneous features allow for more robust and focused modeling of normal vs anomalous traffic patterns. When it comes to anomaly detection, the goal should be to optimize model performance not just for accuracy but also for the computational resources required to respond to threats.

The research aims to identify the most informative features from the 43 commonly used features in the NF-UQ-NIDS-v2 dataset for anomaly detection. Building upon previous work by [6] that used 39 features and after eliminating IP addresses and their ports, the goal is to remove redundant and irrelevant features to focus modeling efforts on the most pertinent anomalies. The objective is to systematically assess feature importance and select an optimal subset of features that enhance the efficiency and effectiveness of NIDS.

The remaining sections of this paper are structured as follows. Section 2 reviews related work on anomaly detection for NIDS, including research on FS techniques. Section 3 then provides a detailed description of the proposed methodology, including the NF-UQ-NIDS-v2 dataset, data preprocessing steps, Correlation Analysis (CA) and the FS methods evaluated. The experimental setup, results, and comparative analysis are presented in Section 4. Finally, Section 5 summarizes the key conclusions and contributions of this research.

2 Related Works

Several studies have explored FS techniques to improve anomaly detection in NIDS. Kumar et al. in [7] proposed using gain ratio FS with an updated Naive Bayes classifier on the NSL-KDD dataset. They compared this approach against correlation-based FS and information gain techniques using Naive Bayes, J48, and REPTree classifiers. A two-step feature selection process is applied to the NSL-KDD dataset in [8]. In the first step, correlation-based feature selection is used to identify relevant features, while, for uncorrelated features removal, symmetrical uncertainty is used in the second step. This dependency-based approach aimed to select useful features while reducing redundancy. In [9] the authors proposed an effective NIDS based on ML and FS techniques. They compared the performance of four ML techniques: Random Forest, K-Nearest Neighbors, SVM and Decision Tree for intrusion detection on the NSL-KDD dataset. The study employed feature selection using the Decision Tree technique to identify important features that impact classification results. Among the techniques evaluated, the Random Forest technique achieved the highest accuracy of 99.72%, outperforming the other techniques. Authors in [10] conducted a comprehensive analysis of the effectiveness of high-frequency features in detecting cyberattacks using machine learning algorithms. They employed various feature selection methods to identify the most relevant high-frequency features from the NSL-KDD dataset. Subsequently, they evaluated various ML algorithms to detect attacks using the selected high-frequency features. Their results demonstrated that high-frequency features significantly improved the attack detection accuracy of the employed ML algorithms, with Random Forest outperforming the other algorithms.

In [11], the authors analyzed the UNSW-NB15 dataset by first applying XGBoost for feature reduction. They tested several classifiers on the reduced feature space, including Support Vector Machine (SVM), k-nearest-neighbor, Logistic regression, Artificial neural network, and Decision tree. The Decision tree classifier improved its testing accuracy from 88.13% to 90.85% using the XGBoost-selected features, demonstrating the utility of FS. Omar Almomani [12] introduced a metaheuristic feature selection method for the UNSW-NB15 dataset using Particle swarm optimization, firefly optimization, grey wolf optimization and genetic algorithm. After iterative optimization, a 30-feature subset was identified and J48 and SVM classifiers trained on these features achieved training accuracy of 90.48% and 90.12%, respectively. In [13] hybrid Information gain and random forest are proposed based on the Recursive Feature Elimination (RFE) method. It combines Information gain and Random Forest with RFE on the UNSW-NB15 dataset. Feature reduction from 42 to 23 features improved multilayer perceptron multi-class classification accuracy from 82.25% to 84.24%.

In [14], authors evaluated multiple classifiers with Principal component analyzes and Gini impurity-based weighted forest feature selection on the ToN-IoT, UNSW-NB15 and Bot-IoT datasets. The Random Forest classifier with the proposed FS method achieved 97% accuracy on the ToN-IoT and UNSW-NB15 datasets, while the accuracy reached 99% on the Bot-IoT database with Principal component analyzes. Authors in [15] combined statistical filters like Chi-Square, Pearson’s Correlation Coefficient (PCC) and Mutual Information (MI) with a Non-dominated sorting genetic algorithm metaheuristic approach for feature optimization on the ToN-IoT dataset. With only 13 selected features, accuracy reached 99.48%, demonstrating the proposed method’s effectiveness. A genetic algorithm-based method for FS is used on the Bot-IoT dataset [16]. With only 6 selected features, botnet attack detection accuracy reached 99.98% and F1-score was 99.63%, showing the purpose of feature reduction.

Authors in [17] presented a weighted ensemble feature scoring technique tailored for the CSE-CIC-IDS2018 dataset focusing on FTP, SSH, SQL, XSS, and Web attacks. Optimal weights were determined using the Taguchi experimental design, and a Decision tree, Random Forest, and SVM classifiers were used to classify attacks. The proposed method significantly improved accuracy and F1-score for XSS, Web, and SQL injection attacks using fewer features. For XSS, the number of features is reduced from 10 to 2, while for Web attacks, the number of selected features is 13, compared to 44 originally. SQL injection attack is successfully classified with selected 7 from originally 26 features. In [18], the correlation-based FS is applied to the CIC-IDS2018 dataset for IDS. With optimized 48 features and a 60:40% testing and training data ratio, the IDS achieved 99.9995% accuracy and a true positive rate of 99.9992%, demonstrating the utility of correlation analyses as a feature selection method.

In 2022, Sarhan et al. [6] proposed a standard feature set for NIDS datasets. The research is focused on general network flow-based intrusion detection. This dataset is formed by merging and converting the four datasets (UNSW-NB15 (2015), BoT-IoT (2018), CSE-CIC-IDS-22018 (2018), and ToN-IoT (2020)) into NetFlow version 9 format. In the available literature, two sets of features can be found: one with 43 features [6] and a smaller one with 12 features only [4]. The experiments showed that the 43-feature dataset performed better than the 12-feature dataset. The NF-UQ-NIDS-v2 dataset showcases the benefits of shared dataset features by enabling the consolidation of multiple smaller datasets.

Our research examines the feature preprocessing and selection techniques for the selected portion of the NF-UQ-NIDS-v2 dataset. This dataset's choice is supported by its integration of four well-established datasets and its temporal diversity from 2015 to 2020, effectively capturing the evolution of cyberthreats over time. Correlation Analysis is applied to identify and eliminate redundant features within the 39-feature space, after eliminating IP addresses and their ports. The goal is to refine this feature set to identify informative features that improve anomaly detection performance.

3 Proposed Model Development

This section provides a detailed description of the proposed method to identify attacks. As well, Figure 1 illustrates pipeline architecture used to develop an ML-based model, including the CA and FS processes. The selected portion of the NF-UQ-NIDS-v2 NetFlow dataset is leveraged to facilitate reliable evaluation across diverse network environments.

Two approaches are pursued to compare feature sets. The first approach addresses redundancy through CA. Pearson correlation coefficients are calculated between all feature pairs as well as between each feature and the true label of benign or attack type traffic (target variable). Highly correlated features are discarded, except one feature among them. Optimized feature subsets are then selected using FS methods. The second approach acts directly without performing CA. After eliminating IP addresses and ports as in the first approach, feature subsets are immediately selected using FS methods without the initial redundancy removal step.

Optimized feature subsets are selected using three methods: Recursive Feature Elimination (RFE), Mutual Information (MI), and One-way ANOVA (ANOVA). For classification, an extremely randomized trees (ExtraTrees), ensemble model is used to perform binary classification of normal and traffic under attack.

Its performance is comprehensively evaluated on multiple metrics, including accuracy, F1-score, recall, precision, Area Under Curve (*AUC*), False Alarm Rate (*FAR*), and scoring time. This multifaceted assessment examines both detection capabilities and efficiency. Varying methods and feature sizes systematically examine the optimal approaches for maximizing detection performance.

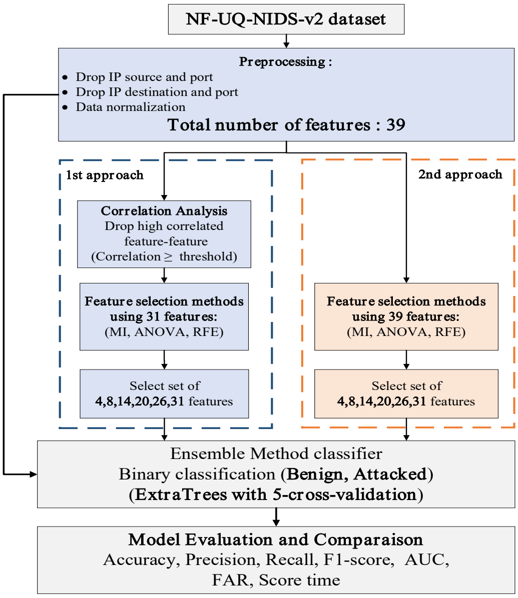
NF-UQ-NIDS-v2 dataset
Preprocessing :
Drop IP source and port
Drop IP destination and port
Data normalization
Total number of features : 39
1st approach
Correlation Analysis
Drop high correlated feature-feature
(Correlation ≥ threshold)
Feature selection methods using 31 features:
(MI, ANOVA, RFE)
Select set of 4,8,14,20,26,31 features
2nd approach
Feature selection methods using 39 features:
(MI, ANOVA, RFE)
Select set of 4,8,14,20,26,31 features
Ensemble Method classifier
Binary classification (Benign, Attacked)
(ExtraTrees with 5-cross-validation)
Model Evaluation and Comparaison
Accuracy, Precision, Recall, F1-score, AUC,
FAR, Score time

Figure 1

Proposed approaches pipeline

3.1 Dataset

The first step of the proposed classification model and methodology is to collect data on traffic flow. The dataset selected for this study is the NF-UQ-NIDS-v2 [6], a pre-labeled NetFlow packet containing benign and compromised data. The NF-UQ-NIDS-v2 dataset was originally published by Sarhan et al. [6]. It was constructed by merging and converting four existing datasets – (UNSW-NB15, BoT-IoT, CSE-CIC-IDS2018, and ToN-IoT) – into the NetFlow version 9 format. This unified dataset combines the benefits of these diverse sources, enabling ML modeling across varied network environments and attack types.

The NF-UQ-NIDS-v2 dataset has a total of 75,987,976 records, out of which 25,165,295 (33.12%) are benign flows and 50,822,681 (66.88%) are various attacks. A total of 5,034,361 records were randomly selected from the original 25,165,295 benign records. Out of the original 50,822,681 attacks, 5,850,596 were randomly chosen for the attack records.

Table 1
NetFlow features in the NF-UQ-NIDS-v2 dataset with their index

Feature	**Name**	**Description**
*f*0	**IPV4_SRC_ADDR**	IPv4 source address (dropped)
*f*0	**IPV4_DST_ADDR**	IPv4 destination address (dropped)
*f*0	**L4_SRC_PORT**	IPv4 source port number (dropped)
*f*0	**L4_DST_PORT**	IPv4 destination port number (dropped)
*f*1	**PROTOCOL**	IP protocol identifier byte
*f*2	**L7_PROTO**	Layer 7 protocol (numeric)
*f*3	**IN_BYTES**	Incoming number of bytes
*f*4	**IN_PKTS**	Incoming number of packets
*f*5	**OUT_BYTES**	Outgoing number of bytes
*f*6	**OUT_PKTS**	Outgoing number of packets
*f*7	**TCP_FLAGS**	Cumulative of all TCP flags
*f*8	**CLIENT_TCP_FLAGS**	Cumulative of all client TCP flags
*f*9	**SERVER_TCP_FLAGS**	Cumulative of all server TCP flags
*f*10	**FLOW_DURATION_MILLISECONDS**	Flow duration in milliseconds
*f*11	**DURATION_IN**	Client to Server stream duration (msec)
*f*12	**DURATION_OUT**	Client to Server stream duration (msec)
*f*13	**MIN_TTL**	Min flow TTL
*f*14	**MAX_TTL**	Max flow TTL
*f*15	**LONGEST_FLOW_PKT**	Longest packet (bytes) of the flow
*f*16	**SHORTEST_FLOW_PKT**	Shortest packet (bytes) of the flow
*f*17	**MIN_IP_PKT_LEN**	Length of the smallest flow IP packet observed
*f*18	**MAX_IP_PKT_LEN**	Length of the largest flow IP packet observed
*f*19	**SRC_TO_DST_SECOND_BYTES**	Source (src) to destination (dst) Bytes/sec
*f*20	**DST_TO_SRC_SECOND_BYTES**	Destination (dst) to source (src) Bytes/sec
*f*21	**RETRANSMITTED_IN_PKTS**	Number of retransmitted TCP flow packets (src → dst)
*f*22	**RETRANSMITTED_IN_BYTES**	Number of retransmitted TCP flow bytes (src→ dst)
*f*23	**RETRANSMITTED_OUT_PKTS**	Number of retransmitted TCP flow packets (dst→ src)
f24	**RETRANSMITTED_OUT_BYTES**	Number of retransmitted TCP flow bytes (dst→ src)
*f*25	**SRC_TO_DST_AVG_THROUGHPUT**	Src to dst average thpt (bps)
*f*26	**DST_TO_SRC_AVG_THROUGHPUT**	Dst to src average thpt (bps)
*f*27	**NUM_PKTS_UP_TO_128_BYTES**	Packets whose IP size ≤ 128
*f*28	**NUM_PKTS_128_TO_256_BYTES**	Packets whose IP size > 128 and ≤ 256
*f*29	**NUM_PKTS_256_TO_512_BYTES**	Packets whose IP size > 256 and ≤ 512
*f*30	**NUM_PKTS_512_TO_1024_BYTES**	Packets whose IP size > 512 and ≤ 1024
*f*31	**NUM_PKTS_1024_TO_1514_BYTE:**	Packets whose IP size > 1024 and ≤ 1514
*f*32	**TCP_WIN_MAX_IN**	Max TCP Window src → dst
*f*33	**TCP_WIN_MAX_OUT**	Max TCP Window dst → src
*f*34	**ICMP_TYPE**	ICMP Type * 256 + ICMP code
*f*35	**ICMP_IPV4_TYPE**	ICMP Type
*f*36	**DNS_QUERY_ID**	DNS query transaction ID
*f*37	**DNS_QUERY_TYPE**	DNS query type
*f*38	**DNS_TTL_ANSWER**	TTL of the first A record (if any)
*f*39	**FTP_COMMAND_RET_CODE**	FTP client command return code

For cyberattacks, Analysis, Backdoor, Exploits, Fuzzers, Generic, Shellcode, Theft, Worms, MitM, and Ransomware, all available records from the dataset are selected (2,299, 18,978, 31,551, 22,310, 16,560, 1,427, 2,431, 164, 7,723, 3,425 respectively). In this research, from 143,097 records for Bot attacks, 57,214 were chosen, while 123,982 available records for Brute Force attacks and 21,748,351 DDoS attacks, 49,819 and 2,175,293 records were selected, respectively. For the DoS type of attack, 1,790,782 records were picked out of 17,875,585 available, while 46,862 records out of 116,361 and 264,354 out of 2,633,778 were selected for Infiltration and Reconnaissance type of attack, respectively. Furthermore, from 684,897 records for Injection attacks, 273,272 were selected, and from 1,153,323 Password attacks, 460,737 records were chosen. It picked 379,015 and 246,380 records out of 3,781,419 and 2,455,020 for Scanning and XSS type of attack, respectively. The original dataset contained 75,987,976 records, from which 10,884,957 records were selected for training and testing purposes, maintaining a balanced portion to address class imbalance. The training set comprised 80% of the chosen records, totaling 8,707,965 records (4,027,489 benign and 4,680,476 compromised). The remaining 20%, consisting of 2,176,992 records, were designated for testing.

A total of 43 relevant features were chosen to construct this dataset in the original database. Table 1 shows the descriptions of these features, with four features omitted (source/destination IP addresses and their port number- features *f*0) as in [6]. The feature number represents the feature's position in the dataset. By referring to the feature numbers, the specific features selected through the feature selection methods can be easily identified and analyzed.

3.2 Data Pre-Processing

Data preprocessing is a crucial initial step when applying ML to real-world problems. In this research, the IP addresses and their associated ports, following the approach of [6], are not considered, resulting in a set of 39 features. The key data preprocessing challenge involves dealing with features that have different scales or distributions. Normalization facilitates feature transformation to share comparable value ranges. The NF-UQ-NIDS-v2 dataset exhibits these common data preprocessing challenges because it contains features with widely varying value ranges. To handle this challenge, the dataset has been preprocessed using feature normalization to standardize the feature value scales to a predefined range.

Normalization plays a crucial role in improving the performance and reliability of the ML model by ensuring that each feature contributes equally to the model's prediction results. In this study, the Min-Max normalization technique was applied to rescale all values linearly within the dataset to a uniform range between 0 (corresponds to minimum) and 1 (corresponds to maximum) [19]. One notable advantage of Min-Max normalization is its ability to maintain the underlying relationships among the original data values.

3.3 Correlation Analysis

The Pearson correlation coefficient (*r*) was utilized to measure the relationship between variables in the dataset. Determined through CA, *r* quantifies the degree and direction of the linear relationship between two variables. Its values range from -1 (a perfect negative correlation) to 1 (a perfect positive correlation). The formula for the Pearson correlation coefficient is:

$$r = \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum (x_i - \overline{x})^2 (y_i - \overline{y})^2}}, \tag{1}$$

where x_i and y_i denote the values of the two variables for the *i*-th observation, and $\bar{x}$ and $\bar{y}$ are the sample means of the respective variables.

Our analysis employed a correlation-based approach to assess the relationships among features and between features and the target variable y (0 for benign and 1 for compromised records). We aimed to identify and retain the most informative features while eliminating highly correlated ones to ensure the model's robustness and interpretability.

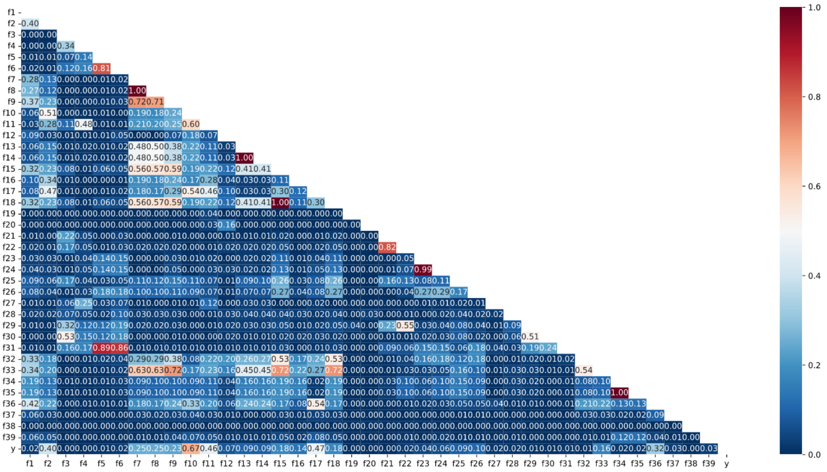

Figure 2

Correlation analysis between analyzed features and between features and target variable

Analysis of the correlation matrix (Figure 2) leads to several notable relationships between the features and with the target variable *y*. The most highly correlated feature-feature pairs were *f*15-*f*18 (1.0), *f*34-*f*35 (1.0), *f*13-*f*14 (1.0), *f*7-*f*8 (1.0), and *f*23-*f*24 (0.99), indicating potential redundancy. In contrast, feature pairs like *f*4-*f*17, *f*19-*f*20, *f*3-*f*38, *f*16-*f*20, and *f*27-*f*39 exhibited negligible correlation, suggesting distinct information. Concerning the target variable *y*, features *f*10, *f*11, *f*17, *f*2 and *f*36 demonstrated relatively high correlations of 0.673, 0.457, 0.469, 0.4020 and

0.323, respectively. On the other hand, features $f19$, $f38$, $f20$, $f3$, and $f4$ exhibited weak correlations with y, ranging from 0.0004 to 0.009. These findings informed the subsequent feature selection process, where highly correlated features were carefully evaluated for potential dimensionality reduction while retaining the most informative features for the classification task. The columns in the dataset that have an absolute value of feature-feature correlation coefficient greater than a threshold of 0.8 were dropped and one among them was retained. This decision aimed to eliminate redundancies and reduce multicollinearity, factors that could compromise the performance and interpretability of the proposed model.

3.4 Feature Selection

Feature selection refers to techniques that select a subset of the most relevant features for a dataset. Fewer features can allow ML algorithms to run more efficiently (less space or time complexity) and be more effective. Some ML algorithms can be misled by irrelevant input features, resulting in lower performance [20]. FS methods are one of the most significant pre-processing phases to succeed the anomaly detection models [21]. Using optimized subset features not only improves the accuracy and detection rate of the classifier but also reduces the execution time, which can help develop a lightweight model that can detect malicious attacks in a real-time network. In addition, avoiding the curse of dimensionality through the FS methods makes the model less prone to overfitting problem. Thus, removing significant noisy and informationless features has gained the attention of many researchers to use FS strategies in many cybersecurity intelligence solutions to achieve a high model performance using ML tasks [22]. This study used three FS approaches: RFE, MI and ANOVA. These methods represent both wrapper and filter techniques for feature selection.

RFE operates as a wrapper-type FS algorithm, using a specific ML algorithm within its core to facilitate FS [23]. It involves searching for a subset of features by initially including all features from the training dataset and then removing features until the desired number remains. This is achieved by fitting the given ML algorithm used in the model's core, ranking features by importance, discarding the least important features, and re-fitting the model. This process is repeated until a specified number of features remains [24]. RFE is popular due to its ease of configuration and usage and its effectiveness in selecting the most relevant features for predicting the target variable within a training dataset. In this research, ExtraTrees is used as an estimator for RFE.

In contrast, as multivariate and univariate filter methods, MI and ANOVA rely on statistical relationships in the data to select features without any learning model [25]. MI quantifies how much knowing one variable reduces uncertainty about another variable. It is a widely used criterion in FS, determining the relevance between features and target classes. Features with high MI amongst themselves are redundant [20, 26]. In FS, the focus is on the mutual information between candidate features and the target variable. A higher value of this information indicates that the feature is highly relevant to predicting the target. To avoid redundant features, the MI between features can be used where highly redundant features have higher MI value. In many MI-based filters, the objective is to maximize the relevance term while minimizing the redundancy term. ANOVA, a statistical technique, compares the means between two or more data groups. It tests the null hypothesis that the means of all groups are equal against the alternative hypothesis that at least one group's mean is different from the others. The F-test statistic is calculated as the ratio of the between-group variance (var_{BG}) to the within-group variance (var_{WG}) [27]:

$$F = \frac{var_{BG}}{var_{WG}} \tag{2}$$

In FS for binary classification, the F-test ranks the importance of each feature by its ability to differentiate between the two classes. A p-value obtained from the F-distribution assesses the test's significance. Features with high F-values and low p-values are selected for inclusion in the classification model [28]. The diversity of these approaches enables a thorough evaluation of FS techniques and their potential benefits for Anomaly-based NIDS.

3.5 Ensemble Method Classifier (ExtraTrees)

Machine learning algorithms can classify traffic as normal or under attack. Their ability to model unknown patterns makes ML suitable for identifying new attacks. Once trained, these models can accurately classify unknown traffic patterns and take proactive measures to mitigate anomalies [29]. This research adopts a supervised approach using the labeled NetFlow dataset. Normal traffic was assigned to class 0, anomalies were labeled as class 1, and the traffic label represents the target variable y. Ensemble-based methods are preferred for classifying unseen instances, as they demonstrate superior performance across a wide range of data sizes or types in intrusion detection [30]. To carry out this classification task, an ExtraTrees ensemble classifier was used as it belongs to the "trees" family. It has demonstrated reliable performance in NIDS datasets, allowing for a valid comparison with [6]. ExtraTrees is known for reducing overfitting and improving performance [31]. This study selected the ExtraTrees classifier, employing 50 randomized decision tree estimators.

4 Results and Discussion

The experimentation phase involved a hardware setup consisting of an 11th Gen Intel(R) Core (TM) i7-11800H processor running at a frequency of 2.30G Hz. The system had 16GB of RAM and an NVIDIA RTX 3060 GPU. The Python programming language (3.9.16) and the Scikit-learn platform (1.2.1) were used. For this study, seven metrics were made to ensure a comprehensive assessment. These metrics are based on the confusion matrix with four conditions under binary classification: *TP* – True Positive, *TN* – True Negative, *FP* – False Positive, *FN* – False Negative. The accuracy of the proposed ML algorithm can be defined as [10]:

$$Acc = (TP+TN)/(TP+FP+TN+FN), \tag{3}$$

while True Positive Rate (*TPR* or *Recall*) is [10]:

$$TPR\ (Recall) = TP/(TP+FN). \tag{4}$$

Precision (*Prec*) is defined as [10]:

$$Prec = TP/(TP+FP), \tag{5}$$

and False Alarm Rate (*FAR* or False Positive Rate) can be calculated as [10]:

$$FAR = FP/(FP+TN). \tag{6}$$

F1-Score is calculated using *Recall* and *Precision* as [10]:

$$F1\text{-}Score = 2 \times (Recall \times Prec\ /\ Recall+Prec) \tag{7}$$

The dependence between False Positive Rate and True Positive Rate can be represented as a curve showing the classifier's performance between error costs and class distributions. Area Under the Curve (*AUC*) shows the accuracy of the model estimation to be obtained as a result of the classification [10]:

$$AUC = \int_0^1 TPR(FAR)\ \mathrm{d}_{FAR} \tag{8}$$

In order to compare the complexity of the proposed method, *Score time* is used, and it refers to the duration required for predicting a single test sample. In this context, a “single test sample” is an individual network flow or record in the test dataset. The *Score time* was calculated by timing the prediction process for the entire test set and then dividing it by the number of samples, rather than selecting a specific individual sample. This method ensures a more representative measure of prediction speed across varied network records in the dataset. It should be noted that while GPU acceleration was used for model training, the *Score time* refers to CPU-based prediction performance.

The experimental design was structured into two primary approaches, with and without CA, as shown in Figure 1. These approaches were used to evaluate the performance of the FS methods MI, RFE, and ANOVA.

4.1 Experiment 1

After the preprocessing step, correlation coefficients for both feature-feature and feature-target variables were calculated, as shown in Figure 2. Features with absolute correlation coefficients (feature-feature) greater than 0.8 were identified. To ensure redundancy reduction, only one feature was retained within each group of correlated features, while the remaining features were eliminated from the analysis.

Based on the obtained results, $f31$ was retained over $f5$ and $f6$, as they were highly correlated with $f31$ and with each other. In the case of $f22$ and $f21$, $f22$ was chosen as it showed a higher correlation with y. Similarly, $f24$ was selected over $f23$ due to its higher correlation with y. In the case of $f7$, $f13$, $f15$ and $f34$ they were favored over $f8$, $f14$, $f18$ and $f35$, respectively. This decision was based on the fact that each pair of features exhibited a high correlation and shared the same correlated value with the target variable y. As a result, the following features were removed for further analysis: $f5, f6, f8, f14, f18, f21, f23, f35$, resulting in a set of 31 features. In the next step, the three methods for FS (RFE, ANOVA and MI) are implemented with various numbers of selected features (4, 8, 14, 20 and 26). The selected features are given in Table 2.

Table 2

FS features index results for 31 features set after CA

Methods	4	8	14	20	26
RFE	$f2, f10, f13, f32$	$f2, f7, f10, f13, f15, f17, f32, f33$	$f2, f7, f9, f10, f13, f15, f16, f17, f25, f26, f32, f33, f34, f36$	$f1, f2, f3, f4, f7, f9, f10, f13, f15, f16, f17, f25, f26, f27, f28, f32, f33, f34, f36, f39$	$f1, f2, f3, f4, f7, f9, f10, f11, f12, f13, f15, f16, f17, f24, f25, f26, f27, f28, f29, f32, f33, f34, f36, f37, f38, f39$
ANOVA	$f2, f10, f11, f17$	$f2, f7, f9, f10, f11, f15, f17, f36$	$f2, f7, f9, f10, f11, f12, f13, f15, f16, f17, f25, f26, f33, f36$	$f1, f2, f7, f9, f10, f11, f12, f13, f15, f16, f17, f24, f25, f26, f27, f33, f34, f36, f37, f39$	$f1, f2, f3, f7, f9, f10, f11, f12, f13, f15, f16, f17, f24, f25, f26, f27, f28, f29, f30, f31, f32, f33, f34, f36, f37, f39$
MI	$f3, f13, f15, f25$	$f2, f3, f13, f15, f16, f25, f26, f32$	$f1, f2, f3, f4, f10, f11, f13, f15, f16, f17, f25, f26, f32, f33$	$f1, f2, f3, f4, f7, f9, f10, f11, f13, f15, f16, f17, f25, f26, f27, f32, f33, f36, f37, f38$	$f1, f2, f3, f4, f7, f9, f10, f11, f12, f13, f15, f16, f17, f22, f24, f25, f26, f27, f28, f31, f32, f33, f34, f36, f37, f38$

Analyzing the selected features by all three methods in Table 2, it can be observed an increasing number of common features as the subset size grows larger. For the 8-feature subset, the intersection includes features $f2$ and $f15$. For the 14-feature subset, in addition to features $f2$ and $f15$, the intersection expands to include features $f10$, $f13$, $f16$, $f17$, $f25$, $f26$, and $f33$ while for the 20-feature subset, additional features $f1$, $f7$, $f9$, $f27$, and $f36$ are consistently selected across all three methods. Finally, for the 26-feature subset, the substantial features: $f3, f11, f12, f24, f28, f32$, $f34$ and $f37$ are added. The selected features in all analysis consist of information that distinguishes normal and traffic under attack. Binary classification experiments were conducted using the selected features for different set sizes. To reliably evaluate the datasets, five cross-validation splits were conducted and the average metrics were collected. The results of this experiment are listed in Table 3 together with all 39 features set.

Table 3
Binary classification results using features selected after correlation analysis

Number of features		**4**	**8**	**14**	**20**	**26**	**31**	**39**
Acc	RFE	0.9585	0.9813	0.9790	0.9790	0.9797	0.9797	0.9797
	ANOVA	0.8916	0.9692	0.9786	0.9788	0.9792		
	Mut inf	0.9637	0.9783	**0.9815**	0.9797	0.9797		
F1-score	RFE	0.9584	0.9812	0.9789	0.9789	0.9797	0.9797	0.9797
	ANOVA	0.8916	0.9691	0.9785	0.9787	0.9791		
	Mut inf	0.9636	0.9783	**0.9814**	0.9797	0.9797		
Prec	RFE	0.9583	0.9806	0.9783	0.9783	0.9791	0.9791	0.9791
	ANOVA	0.8970	0.9685	0.9779	0.9781	0.9785		
	Mut inf	0.9631	0.9776	**0.9808**	0.9791	0.9791		
Recall	RFE	0.9609	0.9823	0.9799	0.9798	0.9806	0.9806	0.9806
	ANOVA	0.8968	0.9698	0.9795	0.9797	0.9800		
	Mut inf	0.9655	0.9795	**0.9825**	0.9806	0.9806		
AUC	RFE	0.9866	0.9973	0.9938	0.9938	0.9948	0.9948	0.9948
	ANOVA	0.9560	0.9904	0.9936	0.9937	0.9940		
	Mut inf	0.9904	0.9971	**0.9974**	0.9948	0.9948		
FAR %	RFE	0.6098	**0.3589**	0.8282	0.8469	0.7682	0.7661	0.7756
	ANOVA	3.3782	2.1313	0.8552	0.8378	0.8261		
	Mut inf	1.0936	0.4822	0.4097	0.7645	0.7686		
Score time (µs)	RFE	55.3298	**73.7132**	101.4184	107.5398	105.7835	103.8625	112.0425
	ANOVA	59.9140	99.0251	100.8816	108.9671	109.2872		
	Mut inf	80.9242	77.9979	88.5148	103.1207	94.9661		

After applying CA and removing highly correlated features, the performance of the FS methods was compared on the reduced 31-feature set with the complete set. As shown in Table 3, the results indicate that using 31 features, obtained after CA, yields similar results to the complete 39-features set across various metrics. Furthermore, the reduced feature set exhibited a lower *FAR*, decreasing by 1.22%, and a reduced scoring time of approximately 7.30%.

Upon applying the FS methods and analyzing the results, it was observed that the MI and RFE methods yielded the best and closest results using 14 and 8 features, respectively. Moreover, these subsets outperformed both the 31-feature and the 39-features sets. For features greater than 14, the MI method provides the best results according to analyzed metrics while for 8 features, the RFE method shows the highest performance. The MI method gives the highest metrics values, except for *FAR* and *Score time* where features selected by RFE exhibit the best results.

It can be noticed that the RFE method with 8 selected features achieved a decrease of 53.15% in *FAR* and a decrease in scoring time of 29.03% compared to the 31-feature set. In contrast, MI with 14 features achieved a decrease of 46.52% in *FAR* and a gain in scoring time of 14.78% compared to the 31-feature set. However, ANOVA started from 8 features to provide closer results on all analyzed metrics to the 31-feature set, but it consistently exhibited higher *FAR* and scoring time. In conclusion, the RFE method with 8 features presents the optimal subset and achieves the best results. It exhibits a lower *FAR*, crucial in anomaly detection, and a lower scoring time. Furthermore, it closely matches the performance of MI with 14 features across other evaluation metrics.

4.2 Experiment 2

In the second approach, FS was directly applied to the 39 features, resulting in the selected features 4, 8, 14, 20, 26 and 31. The results are detailed in Table 4.

Table 4

FS features index results for all 39 features

Methods	4	8	14	20	26	31
RFE	$f2, f10, f13, f32$	$f2, f8, f10, f13, f14, f15, f18, f32$	$f2, f7, f8, f9, f10, f13, f14, f15, f16, f17, f18, f32, f33, f36$	$f1, f2, f5, f7, f8, f9, f10, f13, f14, f15, f16, f17, f18, f25, f26, f32, f33, f34, f35, f36$	$f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f13, f14, f15, f16, f17, f18, f25, f26, f27, f28, f32, f33, f34, f35, f36, f37$	$f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12, f13, f14, f15, f16, f17, f18, f25, f26, f27, f28, f29, f32, f33, f34, f35, f36, f37, f38, f39$
ANOVA	$f2, f10, f11, f17$	$f2, f7, f8, f9, f10, f11, f17, f36$	$f2, f7, f8, f9, f10, f11, f13, f15, f16, f17, f18, f26, f33, f36$	$f2, f7, f8, f9, f10, f11, f12, f13, f14, f15, f16, f17, f18, f23, f24, f25, f26, f33, f36, f37$	$f1, f2, f6, f7, f8, f9, f10, f11, f12, f13, f14, f15, f16, f17, f18, f22, f23, f24, f25, f26, f27, f33, f35, f36, f37, f39$	$f1, f2, f5, f6, f7, f8, f9, f10, f11, f12, f13, f14, f15, f16, f17, f18, f22, f23, f24, f25, f26, f27, f28, f29, f31, f33, f34, f35, f36, f37, f39$
MI	$f13, f14, f15, f18$	$f2, f3, f13, f14, f15, f16, f18, f25$	$f2, f3, f5, f10, f11, f13, f14, f15, f16, f17, f18, f25, f26, f32$	$f1, f2, f3, f4, f5, f6, f9, f10, f11, f13, f14, f15, f16, f17, f18, f25, f26, f27, f32, f33$	$f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f13, f14, f15, f16, f17, f18, f25, f26, f27, f32, f33, f34, f36, f37, f38$	$f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f13, f14, f15, f16, f17, f18, f21, f22, f24, f25, f26, f27, f28, f32, f33, f34, f35, f36, f37, f38$

For the 8-feature subset, the only common feature across all three methods is $f2$, while in the 14-feature subset, common features across all methods include $f2$, $f10$, $f13$, $f15$, $f16$, $f17$, and $f18$. The 20-feature subset includes the above features and additional ones: $f9$, $f14$, $f25$, $f26$, and $f33$. Expanding to the 26-feature subset, the common features across all three methods include $f1$, $f6$, $f7$, $f8$, $f27$, $f36$, and $f37$. Finally, in the 31-feature subset, the common features across all three methods also include $f5$, $f11$, $f28$, $f34$, and $f35$. As in experiment 1, the selected features represent important information for benign and anomalous traffic. The results obtained for the binary classification technique using the entire feature space as used in [6], including FS subsets without CA, are presented in Table 5.

Table 5
Binary classification results using features selected without correlation analysis

Number of features		**4**	**8**	**14**	**20**	**26**	**31**	**39**
Acc	RFE	0.9585	0.9795	0.9785	0.9789	0.9791	0.9797	0.9797
	ANOVA	0.8916	0.9161	0.9782	0.9788	0.9788	0.9788	
	Mut inf	0.9557	0.9739	0.9813	**0.9816**	0.9797	0.9797	
F1-score	RFE	0.9584	0.9794	0.9784	0.9789	0.9791	0.9809	0.9797
	ANOVA	0.8916	0.9160	0.9781	0.9787	0.9787	0.9800	
	Mut inf	0.9556	0.9739	0.9813	**0.9816**	0.9797	0.9797	
Prec	RFE	0.9583	0.9788	0.9778	0.9783	0.9785	0.9791	0.9791
	ANOVA	0.8970	0.9158	0.9775	0.9781	0.9781	0.9781	
	Mut inf	0.9550	0.9732	0.9806	**0.9809**	0.9791	0.9790	
Recall	RFE	0.9609	0.9806	0.9794	0.9798	0.9800	0.9806	0.9806
	ANOVA	0.8968	0.9181	0.9791	0.9796	0.9797	0.9797	
	Mut inf	0.9569	0.9752	0.9823	**0.9826**	0.9806	0.9806	
AUC	RFE	0.9866	0.9949	0.9935	0.9938	0.9939	0.9948	0.9948
	ANOVA	0.9560	0.9623	0.9933	0.9937	0.9938	0.9938	
	Mut inf	0.9879	0.9959	0.9974	**0.9975**	0.9948	0.9948	
FAR %	RFE	0.6098	0.5011	0.8436	0.8532	0.8339	0.7744	0.7756
	ANOVA	3.3782	5.3901	0.8762	0.8483	0.8527	0.8503	
	Mut inf	2.6874	0.8664	0.4263	**0.4038**	0.7691	0.7730	
Score time (µs)	RFE	59.7418	70.9827	96.2522	106.7953	107.5689	109.7750	112.0425
	ANOVA	58.7220	107.1806	100.9185	103.1657	105.6958	110.5406	
	Mut inf	**56.2335**	92.5945	89.6046	85.6696	104.7706	108.3073	

As in experiment 1, the MI method provides the highest or equally high performance as the RFE method for 14 or higher selected features. For 4 and 8 features based on accuracy, F1-score, precision, recall and *FAR*, the RFE method provides the highest performance, while MI is chosen as best according to *AUC*. MI and RFE are the optimal choices for *Score time* for 4 and 8 features, respectively

The subset of 20 features selected by MI achieved the optimal performance in accuracy, F1-score, precision, recall, *AUC* and *FAR*, while it performed the second lowest scoring time. Furthermore, MI with 14 features delivered similar results to the 20-feature subset and outperformed the use of all 39 features. RFE performed well with only 8 features, achieving results close to those obtained with the full feature set. However, MI outperformed RFE in terms of performance metrics when using 14 and 20 features. ANOVA method performed well starting from 14 features but consistently underperformed compared to other methods, including the full feature set, especially in terms of *FAR*. Given the importance of *FAR* in anomaly detection, the 20-feature subset selected by MI is the optimal choice for binary classification and anomaly detection tasks. It achieves high performance across various metrics while maintaining a low *FAR*.

4.3 Comparative Analysis

The experiments show that in most cases, RFE and MI methods provide reasonably good performances (accuracy, F1-score, recall, precision and *AUC*). In contrast, ANOVA's performance on these metrics was lower when few features were used. For *FAR*, MI and RFE maintain relatively stable and low *FAR*s as the number of features is reduced while ANOVA leads to higher *FAR*s with fewer features. As expected, there is a consistent overall increase in *Score time* as more features are added. The MI and RFE methods usually yield similar results across all evaluation metrics. Table 5 and Table 3 indicate that the MI and RFE methods achieve performance similar to the full feature set when only 8 selected features are used. In certain cases, these techniques yield marginal improvements over the models using all features, implying the successful elimination of distracting or redundant features. Conversely, the ANOVA method requires approximately 14 features to approach the performance levels of the full feature set.

When considering the scenario with CA, it was found that using the 31-feature set produced similar results to the complete 39-feature set across various metrics while also demonstrating lower *FAR* and scoring time. This suggests that CA effectively reduced the feature set without significantly compromising the overall performance. It can be argued that the performance was improved by dropping the highly correlated features. For the best subset of features, our findings suggest that RFE with 8 features, in conjunction with CA, achieved close results compared to MI with 20 features without CA. Specifically, the RFE method with 8 features, compared to all 39 features, demonstrated an increase in accuracy, F1-score, recall, precision and *AUC* by 0.16%, 0.15%, 0.17%, 0.15% and 0.25%, respectively, while for MI with 20 features provides an increase by 0.19%, 0.19%, 0.20%, 0.18% and 0.27% in these metrics, respectively. Furthermore, considering the *FAR*, the RFE method with 8 features significantly decreased it by 53.73%, while the MI method with 20 features reduced it by 47.94% compared to the 39-features set. Regarding computational efficiency, RFE with 8 features outperformed MI with 20 features, decreasing the *Score time* by 34.21%, whereas the MI method showed a decrease of 23.54%. Given the significance of minimizing *FAR* while maintaining NIDS accuracy, the results indicate a preference for RFE with 8 features along with CA.

Conclusions

This study demonstrates the efficiency of CA and FS techniques for enhancing anomaly-based NIDS. The NF-UQ-NIDS-v2 NetFlow dataset was leveraged, representing an advance over prior work since this benchmark dataset has not been previously explored using FS methods. Two approaches were pursued – one involved applying FS directly to the initial feature set of 39 features, while the second approach implemented FS after conducting CA. Our approach involved evaluating the correlation between features, analyzing the effectiveness of different FS methods, and comparing the results obtained from the selected feature subsets with the full set of features. CA effectively reduced the feature sets without significantly compromising overall performance. The reduced 31 sets after CA yielded comparable results to the full 39 sets while improving *FAR* and scoring time. This indicates that CA successfully eliminated redundant attributes to refine the feature space. RFE, MI, and ANOVA methods then selected optimized feature subsets from the reduced and full sets. An ExtraTrees ensemble classifier performed binary classification of benign and attack traffic. ANOVA did not perform as well as MI and RFE in selecting the most informative features for NIDS. Both MI and RFE exhibited superior performance in improving detection accuracy, reducing *FAR*, and optimizing computational resources. Results indicate RFE filtering on 8 of features leads to an improvement in accuracy of 0.16% compared to the 39 features, and a decrease of *FAR* rate by 53.73%, alongside a time gain of 34.21%.

While the NF-UQ-NIDS-v2 dataset provided a comprehensive evaluation, future work will explore testing on additional datasets to validate the generalizability of our findings across different network environments. Further research can extend this work by exploring the use of ensemble classifiers and emerging FS methods to improve NIDS performance further. Additionally, it can be explored how these techniques can be applied to multiclass classification.

Acknowledgement

This research has partially been funded by the Military academy, University of Defence in Belgrade under Project VA/TT/1/25-27.

References

[1] Keserwani, P. K. et al.: *An effective NIDS framework based on a comprehensive survey of feature optimization and classification techniques*. Neural Computing and Applications, **35** (7), 2023, pp. 4993-5013

[2] Iglesias, F., Zseby, T.: *Analysis of network traffic features for anomaly detection*. Machine Learning, **101**, 2015, pp. 59-84

[3] Musa, U. S. et al.: *Intrusion detection system using machine learning techniques: A review*. In: 2020 international conference on smart electronics and communication. 2020, pp. 149-155

[4] Sarhan, M. et al.: *Netflow datasets for machine learning-based network intrusion detection systems*. In: Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020, and 13th EAI International Conference on Wireless Internet, WiCON 2020, Virtual Event, December 11, 2020, Proceedings 10. 2021, pp. 117-135

[5] Perez, D. et al.: *Intrusion detection in computer networks using hybrid machine learning techniques*. In: 2017 XLIII Latin American Computer Conference. 2017, pp. 1-10

[6] Sarhan, M. et al.: *Towards a standard feature set for network intrusion detection system datasets*. Mobile networks and applications, 2022, pp. 1-14

[7] Kumar, K., Batth, J. S.: *Network intrusion detection with feature selection techniques using machine-learning algorithms*. International Journal of Computer Applications, **150** (12), 2016

[8] Shahbaz, M. B. et al.: *On efficiency enhancement of the correlation-based feature selection for intrusion detection systems*. In: 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference. 2016, pp. 1-7

[9] Avcı, İ., Koca, M.: *Cybersecurity Attack Detection Model, Using Machine Learning Techniques*. Acta Polytechnica Hungarica, **20** (7) 2023, p. 29-44

[10] Özalp, A. N., Albayrak, Z.: *Detecting Cyber Attacks with High-Frequency Features using Machine Learning Algorithms*. Acta Polytechnica Hungarica, **19** (7), 2022, pp. 213-233

[11] Kasongo, S. M., Sun, Y.: *Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset*. Journal of Big Data, **7** (1), 2020, p. 105

[12] Almomani, O.: *A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms*. Symmetry (Basel), **12** (6), 2020, p. 1046

[13] Yin, Y. et al.: *IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset*. Journal of Big Data, **10** (1) 2023, p. 15

[14] Alhanaya, M., Ateyeh Al-Shqeerat, K. H.: *Performance Analysis of Intrusion Detection System in the IoT Environment Using Feature Selection Technique*. Intelligent Automation & Soft Computing, **36** (3), 2023

[15] Dey, A. K. et al.: *Hybrid Meta-Heuristic based feature selection mechanism for cyber-attack detection in IoT-enabled networks*. Procedia Computer Science, **218**, 2023, pp. 318-327

[16] Liu, X., Du, Y.: *Towards effective feature selection for iot botnet attack detection using a genetic algorithm*. Electronics (Basel), **12** (5), 2023, p. 1260

[17] Göcs, L., Johanyák, Z. C.: *Feature Selection with Weighted Ensemble Ranking for Improved Classification Performance on the CSE-CIC-IDS2018 Dataset*. Computers, **12** (8), 2023, p. 147

[18] Heryanto, A. et al.: *Cyberattack feature selection using correlation-based feature selection method in an intrusion detection system*. In: 2022 9th International Conference on Electrical Engineering, Computer Science and Informatics. 2022, pp. 79-85

[19] Labonne, M.: *Anomaly-based network intrusion detection using machine learning*. Ph.D. dissertation, Polytechnic Institute of Paris, France, 2020

[20] Vergara, J. R., Estévez, P. A.*: A review of feature selection methods based on mutual information*. Neural computing and applications, **24**, 2014, pp. 175-186

[21] Bouzoubaa, K. et al.: *Predicting DOS-DDOS attacks: Review and evaluation study of feature selection methods based on wrapper process*. Int. J. Adv. Comput. Sci. Appl, **12** (5), 2021, pp. 132-145

[22] Bommert, A. et al.: *Benchmark for filter methods for feature selection in high-dimensional classification data*. Computational Statistics & Data Analysis, **143**, 2020, p. 106839

[23] Guarino, I. et al.: *On the use of machine learning approaches for the early classification in network intrusion detection*. In: 2022 IEEE International Symposium on Measurements & Networking (M&N) 2022, pp. 1-6

[24] Awad, M., Fraihat, S.: *Recursive feature elimination with cross-validation with decision tree: Feature selection method for machine learning-based intrusion detection systems*. Journal of Sensor and Actuator Networks, **12** (5), 2023, p. 67

[25] Pudjihartono, N. et al.: *A review of feature selection methods for machine learning-based disease risk prediction*. Frontiers in Bioinformatics, **2**, 2022, p. 927312

[26] Qian, W., Shu, W.: *Mutual information criterion for feature selection from incomplete data*. Neurocomputing, **168**, 2015, pp. 210-220

[27] Ostertagová, E., Ostertag, O.: *Methodology and Application of One-way ANOVA*. American Journal of Mechanical Engineering, **1** (7) 2013, pp. 256-261

[28] Elssied, N. O. F. et al.: *A novel feature selection based on one-way anova f-test for e-mail spam classification*. Research Journal of Applied Sciences, Engineering and Technology, **7** (3), 2014, pp. 625-638

[29] Fosić, I. et al.: *Anomaly detection in NetFlow network traffic using supervised machine learning algorithms*. Journal of Industrial Information Integration, 2023, p. 100466

[30] Kharwar, A. R., Thakor, D. V: *An ensemble approach for feature selection and classification in intrusion detection using extra-tree algorithm*. International Journal of Information Security and Privacy, **16** (1), 2022, pp. 1-21

[31] Sharma, J. et al.: *Multi-layer intrusion detection system with ExtraTrees feature selection, extreme learning machine ensemble, and softmax aggregation*. EURASIP Journal on Information Security, **2019** (1) 2019, pp. 1-16