ISSN: 2182-2069 / E-ISSN: 2182-2077

Towards Robust IDSs: An Integrated Approach of Hybrid Feature Selection and Machine Learning

Mohammad Al-Omari[1], and Qasem Abu Al-Haija[2*]

[1]Assistant Professor, Department of Business Information Technology, Princess Sumaya University for Technology, Jordan. m.alomari@psut.edu.jo, https://orcid.org/0000-0002-5770-6579

[2*]Assistant Professor, Department of Cybersecurity, Jordan University of Science and Technology, Jordan. qsabuahaija@just.edu.jo, https://orcid.org/0000-0003-2422-0297

Received: December 14, 2023; Revised: February 05, 2024; Accepted: March 05, 2024; Published: May 30, 2024

Abstract

Due to the rapid growth of technology, the urgency for effective cybersecurity systems has become increasingly critical, notably within the paradigm of the Internet of Things (IoT) and Cloud services. With an increasing number of security features in communication protocols, there is a heightened need for systems to manage this complexity and protect assets efficiently. Machine learning (ML) techniques are increasingly indispensable in identifying and mitigating cyber threats. However, the vast number of security features can affect the performance of these techniques. This research presents a hybrid feature selection approach integrating Correlation Analysis (CA) and Mutual Information (MI) as filter methods. Moreover, Recursive Feature Elimination with Cross-Validation (RFECV) is also integrated as a wrapper method to identify highly ranked security features efficiently. These selected features are then deployed in tree-based classifiers, namely, Decision Tree (DT) and Random Forest (RF) classifiers, for predicting cyber-attacks. The proposed system is validated using a real-world dataset specific to a Network Intrusion Detection System (NIDS). The empirical results demonstrate that it can detect attacks effectively and significantly reduce the computational complexity compared to existing approaches. Therefore, the proposed system can enhance cybersecurity measures in complex network environments.

Keywords: Feature Selection, Tree-based Classifiers, Cybersecurity, Hybrid Methods, Machine Learning.

1 Introduction

Network environments are more complex than ever, thanks to the development of new communication technologies and advancements in artificial intelligence. The increasing use of cloud services and IoT devices adds to this complexity by creating more varied data flows in cyberspace (Awad & Fraihat, 2023; Yin et al., 2023; Steephen et al., 2022). As a result, safeguarding both tangible and digital assets has emerged as a crucial issue for efficiently using cloud services. Traditional security techniques, such as user authentication, encryption, and firewalls, are helpful but must be more effective in preventing new cyberattacks (Buczak & Guven, 2016; Aqlan et al., 2023).

Journal of Internet Services and Information Security (JISIS), volume: 14, number: 2 (May), pp. 47-67.
DOI: *10.58346/JISIS.2024.I2.004*

*Corresponding author: Assistant Professor, Department of Cybersecurity, Jordan University of Science and Technology, Jordan.

Intrusion Detection Systems (IDS) are vital because they can handle various cyber threats, such as malware, phishing, and denial of service (DoS) attacks. Current trends indicate a growing focus on using ML techniques to develop IDS to efficiently identify and categorize unusual network activities (Thakkar & Lohiya, 2022; Salim et al., 2023; Park et al., 2019; Muralidharan et al., 2020). Well-known ML algorithms like k-nearest Neighbors (k-NN), Support Vector Machines (SVM), and Logistic Regression have been effective in detecting network intrusions (Mishra et al., 2019; Nisioti et al., 2018; Camgozlu et al., 2023). Among these, Decision Trees and their more advanced form, RFs, have become popular for intrusion detection in the supervised learning category.

Tree-based classifiers like Decision Trees and RFs work by making decisions at each node, guided by how much each feature improves the information. These decisions lead to a final leaf node that indicates the class label, such as *'Normal'* or *'Attack' (*Zhao et al., 2020). With the rise of new technologies such as IoT and Cloud services, there has been a significant increase in network traffic data. This change has introduced many new security features, which can negatively impact the accuracy of cyber-attack predictions and lead to increased computational complexity for machine learning-based predictive models (Kasongo & Sun, 2020; Marangunic et al., 2022; Thang et al., 2020). Specifically, these factors can cause overfitting and longer processing times. Given these challenges and the growing need for reliable IDSs, it is essential to improve the design of these models to increase accuracy and reduce false predictions.

To this end, in Machine Learning, especially for predicting cyber attacks, selecting the right features from vast datasets is pivotal (Buczak & Guven, 2016; Priyanka et al., 2023). Efficient feature selection simplifies the model by reducing dimensionality and enhances prediction accuracy. It is essential to develop robust feature selection methods to obtain optimal features. Researchers can significantly improve the model's performance in detecting and mitigating cyber threats by focusing on the optimal security features. Thus, the principal drive of this investigation is to delve deeper into innovative feature selection methods that can significantly impact cyber attack predictions using ML.

1.1 Research Objectives

The key objective of this research is to explore various aspects of creating an efficient and effective IDS for today's complex network settings. The key objectives of this research are outlined below:

- To propose a hybrid feature selection approach that combines CA and MI as filter methods and RFECV as a wrapper method for selecting the top features that aid in effective intrusion detection.
- Employ ML techniques such as RF and DT classifiers to build a prediction model that takes advantage of these selected features.
- To validate the proposed system empirically using a real-world dataset designed specifically for network intrusion detection.

1.2 Research Questions

The study aims to address the following main questions:

1. How does the proposed hybrid feature selection approach reduce the number of features and identify the optimal features?
2. To what extent does the hybrid approach enhance detection rates and reduce the computational complexity of machine learning-based IDS compared to existing methods in this domain?

The rest of this paper is organized as follows. Section 2 reviews the most related work on cyber-attack prediction models, specifically focusing on feature selection. Section 3 explains the methodology

used in this work, including the proposed cyber-attack prediction system. Section 4 details the implementation of experiments and discusses the achieved results, emphasizing the research limitations. The last section concludes the paper, summarizing the primary findings and suggesting directions for future research.

2 Literature Review

In recent times, there has been a growing focus among scholars and security experts on improving the efficiency of IDSs (Asif et al., 2021; Jelena et al., 2023). The urgent need for efficient IDS models incorporating ML has risen, given their capacity to handle large datasets and yield precise predictions. Numerous studies explore the application of ML to identify irregularities in network environments. This review examines existing models and techniques, emphasizing contemporary feature selection and ranking approaches.

Yin et al., (2023) used the UNSW-NB15 dataset to develop a two-step feature selection procedure for an anomaly-based network IDS. At first, they used RF and Information Gain algorithms to weed out irrelevant features. They then used a Multilayer Perceptron classifier in combination with Recursive Feature Elimination to reduce the feature set to 23. The suggested multiclassification model produced an F1-Score of 82.85% and an accuracy of 84.24%.

Using the NSL-KDD dataset, an empirical analysis of attack classification is carried out in Thakkar and Lohiya's study (Thakkar & Lohiya, 2021). For intrusion detection, seven ML classifiers are used: Artificial Neural Networks, RF, Naive Bayes, k-nearest Neighbors, Decision Trees, Support Vector Machines, and Logistic Regression. Feature selection techniques like chi-square, information gain, and recursive feature elimination are used for feature engineering. The findings show that the combination of Recursive Feature Elimination and Support Vector Machines performs better compared to other tested feature selection methods and classifiers.

In their study, Al-Omari et al., (2021) introduced an intelligent tree-based model for intrusion detection aimed at predicting and identifying cyber attacks. The model employed a single-feature ranking method for optimal feature selection before implementing the predictive algorithm. When evaluated on the UNSW-NB 15 dataset, the model demonstrated superior accuracy and computational simplicity relative to conventional ML classifiers.

In the study Patgiri et al., (2019), Recursive Feature Elimination was employed for feature selection in conjunction with Support Vector Machines and RF classifiers. Using the NSL-KDD dataset, the experiments reduced the feature set to 13 out of 41 to facilitate attack categorization. The results showed that Support Vector Machines performed better than RF for the designated attack types.

In a study Maniriho et al., (2020), two ML techniques, a single classifier called K-Nearest Neighbor and an ensemble technique called Random Committee, were used to evaluate the efficacy of an IDS. Two distinct datasets, NSL-KDD and UNSW NB-15, are used to evaluate these techniques. The study employed feature selection to find and use only the most pertinent feature subsets for the selected datasets. For the NSL-KDD and UNSW NB-15 datasets, the misclassification rate differential of 1.19% and 1.62%, respectively, showed that the ensemble-based approach performed better than the single-classifier approach. The study also emphasized the need for more investigation into high dimensionality, big data sets, and IDS performance optimization.

Bhavani et al., (2020) created an IDS utilizing single ML classifiers, RF and DT methods, in a study on the KDD-NSL dataset. The accuracy rate of the RF classifier was 95.323%. However, the study did not address problems with low detection rates and FP.

To identify network intrusion, Raviteja et al., (2020) used a variety of single ML algorithms, such as Decision Tree, RF, Logistic Regression, and Support Vector Machine. The experiments in the study made use of the KDD-NSL dataset. According to the results, the RF classifier outperformed the other algorithms regarding accuracy and execution time. One challenge was that the study's efficacy might have been increased with a single dataset.

Khraisat et al. (2020) proposed a hybrid IDS model that combines one-class Support Vector Machine classifiers with DT classifiers. In particular, an anomaly-based IDS is developed using one-class support vector machines, whereas a signature-based IDS is built using a DT classifier. The model seeks to identify known as well as novel forms of attacks. The NSL-KDD and AFDA datasets are used for the experimental evaluation, with accuracy-focused performance metrics. The model performs well on the AFDA and NSL-KDD datasets.

In their study, Awad & Fraihat (2023) introduced an enhanced feature selection technique named RFECV using a DT estimator (RFECV-DT). They also shed light on the shortcomings of current methods. The features selected via this method were employed to train advanced ML models like Naive Bayes, Logistic Regression, AdaBoost, RF, and Multilayer Perceptron for IDSs. The well-known UNSW-NB15 dataset was used for this purpose.

In their paper, Alissa et al., (2022) employed 34 features from a subset of the UNSW-NB15 dataset to carry out binary classification. Various classifiers, such as Decision Tree, XGB, and Logistic Regression, were evaluated for their model. The DT classifier yielded the best results in terms of accuracy. The F-1 score, recall, and precision were also notably high and consistent with the accuracy.

In their study, Barkah et al., (2023) used the UNSW-NB15 dataset and ran multiple experiments to identify the most effective detection model. They employed Recursive Feature Elimination to select the top 13 features, which were inputted into four different classifiers. Both RF and DT were found to deliver the best results in multiclassification. In terms of accuracy and F1-Score, the two classifiers performed comparably. Similar or lower performance was observed in other test cases where the authors tackled imbalanced data through oversampling and adaptive synthetic methods.

To this end, numerous studies have employed ML techniques emphasizing feature reduction and model simplification to address the complexities and challenges in developing IDS. However, the performance of existing models can vary significantly based on the dataset and ML algorithms employed. This study proposes a hybrid feature selection approach that integrates CA, MI, and RFECV. This approach is applied to the UNSW-NB15 dataset to identify the most relevant security features. Subsequently, DT and RF classifiers are utilized to predict cyber attacks. The contributions of this research shed light on the efficacy of deploying hybrid feature selection methods in conjunction with ML techniques like DT and RF and evaluate their impact on prediction and computational complexity in terms of processing time. To the authors' knowledge, this is the first work that has utilized the proposed hybrid approach. Thus, the proposed hybrid approach can be added to existing methods in the literature so that researchers can further investigate and refine feature selection methods, thereby expanding the field's domain boundaries. The following sections elaborate on the methodology and the proposed IDS in detail. To sum up, we provide a summary table below, Table 1, that recaps the reviewed studies in this paper.

Table 1: Summary of Reviewed Papers in this Research

Reference	Key Attributes	Advantages	Limitations
(Yin et al., 2023)	- Two-stage feature selection - Information Gain, RF, Recursive Feature Elimination - Multilayer Perceptron classifier - UNSW-NB15 dataset	- Achieved 84.24% accuracy, 82.85% F1-Score - Efficient feature selection process	- Limited evaluation of a specific dataset
(Thakkar & Lohiya, 2021)	- Attack classification using the NSL-KDD dataset - Seven classifiers - Recursive Feature Elimination, Support Vector Machines - Various feature selection methods	- Recursive Feature Elimination + SVM outperforms - Comprehensive analysis with multiple classifiers	- Dataset-specific findings
(Al-Omari et al., 2021)	- Tree-based model for intrusion detection - Single-feature ranking method - UNSW-NB15 dataset	- Superior performance and computational simplicity - Optimized feature selection	- Limited dataset evaluation
(Patgiri et al., 2019)	- Recursive Feature Elimination + SVM, RF - NSL-KDD dataset	- Support Vector Machines outperformed RF	- Focus on specific attack types
(Maniriho et al., 2020)	- K-Nearest Neighbor, Random Committee ensemble - NSL-KDD, UNSW NB-15 datasets - Feature selection	- Ensemble-based approach outperformed single classifier - Addressed challenges in intrusion detection	- Challenges like data size and dimensionality are not fully addressed
(Tulasi Bhavani et al., 2020)	- RF, DT on KDD-NSL dataset	- RF achieved 95.323% accuracy	- Lack of focus on detection rates and FP
(Raviteja et al., 2020)	- Decision Tree, Logistic Regression, RF, SVM - KDD-NSL dataset	- RF outperformed in accuracy and execution time	- Limited to a single dataset
(Khraisat et al., 2020)	- Hybrid IDS model with DT and one-class SVM - NSL-KDD, AFDA datasets	- High accuracy on both datasets	- Limited information on other performance metrics

(Awad & Fraihat, 2023)	- RFECV-DT feature selection - Naive Bayes, Logistic Regression, AdaBoost, RF, ML P - UNSW-NB15 dataset	- Introduced RFECV-DT feature selection - Used well-known dataset	- Limited exploration of other feature selection methods
(Alissa et al., 2022)	- Binary classification with Decision Tree, XGB, and Logistic Regression - UNSW-NB15 dataset	- The DT yielded the best results in accuracy	- Limited information on model evaluation metrics
(Barkah et al., 2023)	- Recursive Feature Elimination + UNSW-NB15 dataset - RF, DT classifiers	- Comparative Performance of RF and Decision Tree - Addressed imbalanced data	Limited evaluation of classifiers in other scenarios
Proposed Hybrid Approach	- CA, MI, RFECV feature selection - Decision Tree, RF - UNSW-NB15 dataset	- Integrates multiple feature selection methods - Evaluate the impact on prediction and computational complexity	Yet to be tested and compared with existing methods.

3 Methodology

In this section, the architecture of the IDS is presented, and all parts of the proposed system are discussed in the following subsections.

3.1. System Architecture

The architecture of the proposed system is illustrated in Figure 1 and consists of three primary components: Data Collection and Preprocessing, Hybrid Feature Selection, and Classification. The initial phase involves selecting the research dataset and conducting crucial tasks such as data inspection and elimination of irrelevant data. This phase also focuses on data encoding and normalization to facilitate reliable feature selection and ranking. The second component implements the proposed Hybrid Feature selection approach. The rationale for employing this hybrid approach is to amalgamate top features identified by multiple techniques for use in the classification stage. The final component utilizes DT and RF classifiers to evaluate the performance of each classifier against both the complete feature set and the top-selected feature set. This assessment uses validation metrics, including Accuracy, Precision, Recall, F1-Score, and Fit time.

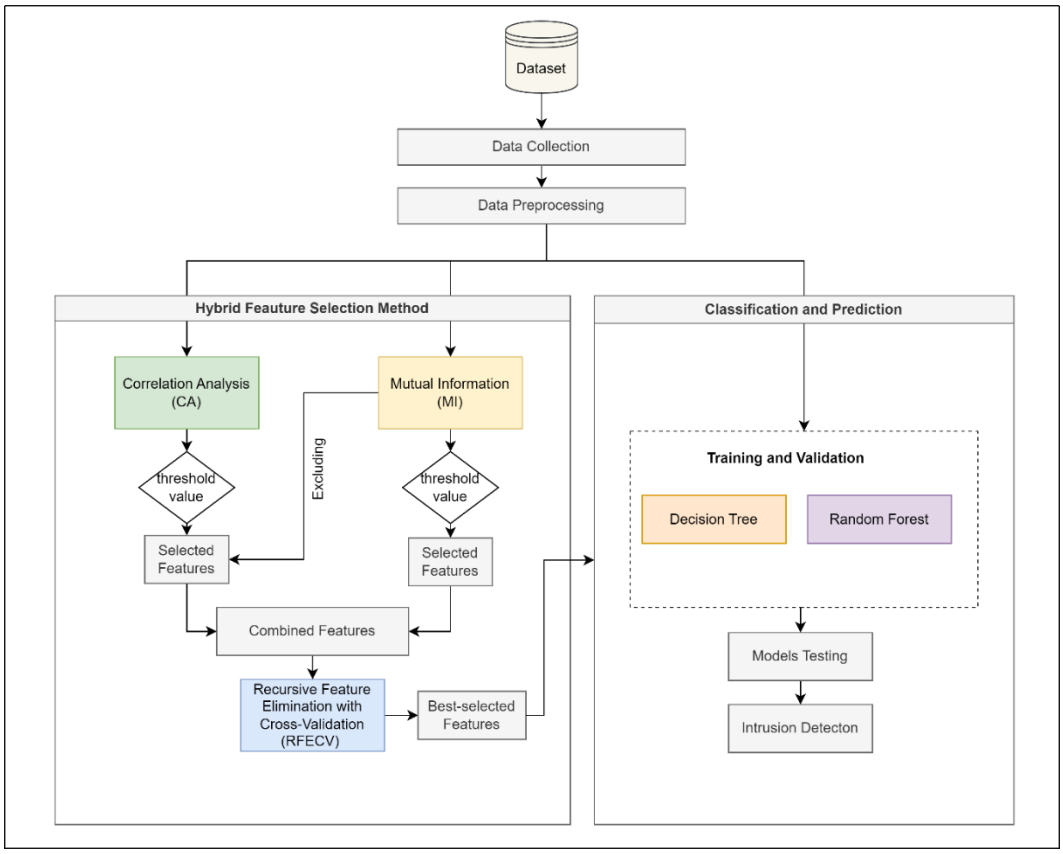

Figure 1: The Proposed System Architecture

3.2. Data Collection and Preprocessing

The current study's subset of the 'UNSW_NB15' dataset comprises 175,341 records specifically for network IDS. This dataset is publicly accessible via the Kaggle platform (Kaggle, 2023). Originating from the Cyber Range Lab of the Australian Centre for Cyber Security, the dataset contains 42 features, not including the *'label'* class that indicates whether a record represents normal activity or an attack. For this research, the feature specifying the attack type was excluded, as it falls outside the research scope.

Data engineering, a crucial component for the effectiveness of the learning process, encompasses several key tasks. These tasks include cleaning rows and columns, encoding features, and normalizing data. All data were examined to find any missing values and unnecessary columns. For instance, the column *'id'* was deleted because it provided no meaningful information. Furthermore, it is worth mentioning that some features found in the dataset require categorical transformations, like the case of *'proto'*, *'service'*, and *'state'* features. These features store categorical data that requires encoding. In this work, the LabelEncoder is used; the use of LabelEncoder is compatible with the UNSW_NB15 dataset, as it assigns distinct numerical labels to each unique categorical value. This encoding enables the classifier to understand and learn from these labels effectively. Unlike OneHotEncoder, which generates individual binary columns, LabelEncoder preserves the categorical essence of the feature without expanding the feature space (Jackson & Agrawal, 2019; Zheng & Casari, 2018). Consequently, the values of the above features have been numerically encoded based on each feature's range.

The subsequent stage focuses on normalizing features that exhibit varying value distributions or scales. Training datasets with high dimensionality demands substantial computational power. Different

normalization techniques, such as Min-Max Normalization, Z-score Normalization, or Decimal Scaling, can mitigate this challenge. The Min-Max Normalization method is chosen in this work and calculated using Formula (1). It transforms the initial values to a range of 0 to 1 while maintaining the relative relationships between the data points (Raju et al., 2020). This step is crucial in data preprocessing and should be executed before deploying the proposed IDS to ensure accurate and reliable performance. In the dataset, any features with significantly disparate scales are appropriately rescaled.

$$X_{Scaled} = \frac{X - Xmin}{Xmax - Xmin} \quad (1)$$

The dataset's features and format are shown in Table 2. The description of each feature can be retrieved online from the Kaggle platform (Kaggle, 2023).

Table 2: Dataset Security Features

Security feature	Format	No. of Features
dur, rate, sload, dload, sinpkt, dinpkt, sjit, djit, tcprtt, synack, ackdat.	Float	11
proto, service, state.	Categorical	3
spkts, dpkts, sbytes, dbytes, sttl, dttl, sloss, dloss, swin, stcpb, dtcpb, dwin, smean, dmean, trans_depth, response_body_len, ct_srv_src, ct_state_ttl, ct_dst_ltm, ct_src_dport_ltm, ct_dst_sport_ltm, ct_dst_src_ltm, ct_ftp_cmd, ct_flw_http_mthd, ct_src_ltm, ct_srv_dst.	Integer	26
is_ftp_login, is_sm_ips_ports.	Binary	2

It is worth mentioning here that Sarhan et al., along with other studies (Awad & Fraihat, 2023; Fraihat et al., 2023; Sarhan et al., 2021), pointed out that using the TTL-based features *'sttl'*, *'dttl'*, and *'ct_state_ttl'* in the UNSW-NB15 dataset might cause biases in classification. These features strongly correlate with the *'Label'* class, which can skew the results. These features were eliminated from the dataset to ensure a reliable and accurate analysis. Thus, the total feature count is reduced from 42 to 39.

3.3. Hybrid Feature Selection

Feature selection plays a pivotal role in constructing effective ML models, particularly in the domain of IDS. This section explores three feature selection techniques, CA, MI, and RFECV, and their efficacy in selecting the best features. The proposed hybrid feature selection approach is also introduced to leverage the strengths of the individual methods.

Correlation Analysis (CA)

CA is commonly used to assess the relationship between two variables. Although it efficiently and accurately identifies connections between linearly related variables, it cannot capture nonlinear relationships (Ambusaidi et al., 2016; Press et al., 1986). The Pearson Correlation Coefficient is the CA technique used in this study. It measures the linear connection between the *'Label'* class and dataset features. The coefficient is in the range of -1 to 1. A strong positive correlation is denoted by a 1, a strong negative correlation by a -1, and no correlation is indicated by a 0. This method decreases the number of features the MI method needs to process and enables quick identification of the most pertinent features. The computation of the correlation coefficient Corr(X; Y) between X and Y is demonstrated in Formula 2.

$$Corr(X;Y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (2)$$

Mutual Information (MI)

MI is a well-known statistical technique widely applied in ML to select features. It can identify nonlinear dependencies between variables (Roulston, 1999). Generating a non-negative value quantifies the information that two variables have in common. The two observed variables may be statistically independent if the value is zero (Cover & Thomas, 2012). When choosing features, a feature is deemed significant if it offers insightful data about the class. If not, it is thought to be redundant. As a result, MI is frequently employed to determine a feature's relevance to a label class. This technique is applied as a backup strategy to find any nonlinear features in the dataset that may still exist. Formula 3 provides a mathematical definition of the MI I(X; Y) between two variables, X and Y.

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \quad (3)$$

Here, $p(x,y)$ is the joint probability distribution function of X and Y, and $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y, respectively.

RFECV

The RFECV method is an efficient and sophisticated feature selection technique. At its core, RFECV combines the strengths of recursive feature elimination (RFE) and cross-validation to determine the optimal number of features for a given model. The process involves recursively removing features, evaluating model performance using cross-validation, and determining which feature subset leads to the highest performance metrics (Kuhn & Johnson, 2013).

In the proposed hybrid approach context, RFECV does not start from scratch. Instead, it leverages the features already identified by previous methods. This pre-selection step significantly reduces the dimensionality of the data and enhances the speed of the RFECV process. As RFECV operates on a reduced set of pre-filtered features, it can more swiftly and effectively select the most relevant features. This synergy ensures that computational efficiency is achieved and that the final feature set chosen is highly relevant to the dataset, promoting improved model performance. This study utilized RFECV with the RF Classifier as an estimator. Besides, the 10-fold Cross-Validation is employed to validate the robustness of the classifier. The dataset of the combined features of the previous methods is divided into 10-fold subsets, and the method is repeated ten times, with each of the ten subsets used exactly once as a validation set. The total results are then averaged to produce a single estimation for each metric. This provides further insight into the proficiency of the selected features in differentiating between the label classes.

Proposed Hybrid Approach

In the proposed hybrid feature selection algorithm, the strengths of both CA and MI-based filter methods are integrated, followed by a wrapper method using RFECV to select the best features. The CA and MI methods are used as initial steps to reduce the dimensionality of data and expedite the RFECV process. This allows RFECV to select the most relevant features more efficiently by working with a smaller set of pre-filtered features. Our experiments have shown that applying RFECV to the complete set of security features is significantly slower compared to its application on a smaller, pre-selected group of features. The proposed hybrid approach is detailed in Algorithm 1.

As Algorithm 1 shows, it initially evaluates the linear relationships between each feature in dataset *d* and the class label *c* using Pearson correlation. Features demonstrating a correlation magnitude exceeding the set threshold (i.e., *correlation_threshold*) are captured in the *high_correlation_features*

set. After this, by computing MI scores, the algorithm evaluates the remaining features not already identified as high correlations to discern any nonlinear relationships with the label class c. Those features with scores surpassing the threshold value (i.e., *mi_threshold*) are incorporated into the *mi_features* set. The results of the two filter methods, *high_correlation_features* and *mi_features*, are combined to form a consolidated feature set, *combined_features*. This set represents the culmination of linear and nonlinear relationships deemed significant by the two filter methods. A wrapper method, RFECV, is employed with an RF model to refine this feature set further. The algorithm fits this model to the *combined features* set and the label class *c*. Features deemed most critical by the RFECV process, indicated by a rank of 1, are selected as the final *selected features*. Finally, the algorithm concludes by returning this optimally selected feature subset, *selected_features*, which encapsulates the best features for the given dataset concerning the target label class.

Algorithm 1: the proposed hybrid feature selection

Input:
***d*:** Dataset with n samples and m features
correlation_threshold: Correlation-based feature selection threshold.
mi_threshold: MI-based feature selection threshold.
Estimator: RF model
***c*:** is the class label in the dataset

Output:
***selected_features*:** Top Features subset

1: Correlation-based Feature Selection:
- Compute the ***correlation_matrix*** between each column of ***d*** and ***c***.
- Identify columns where the absolute value of the correlation exceeds ***correlation_threshold*** and store them in ***high_correlation_features***

2: Prepare for MI-based Feature Selection:
- Create ***remaining_features*** by excluding ***high_correlation_features*** from all features in ***d***.
- Filter ***d*** to create ***d_remaining*** using ***remaining_features***.

MI -based Feature Selection:
- Compute ***mutual_info_scores*** between each column of ***d_remaining*** and ***c***.
- Identify columns in ***d_remaining*** where the MI score exceeds ***mi_threshold*** and store them in ***mi_features***.

3: Combine Features from Both Filter Methods:
- Union ***high_correlation_features*** and ***mi_features*** to get ***combined_features***.
- Filter ***d*** using ***combined_features*** to get ***d_filtered***.

4: RFECV-based Feature Selection:
- Initialize the wrapper with ***estimator***, 10-fold cross-validation, and accuracy scoring.
- Fit ***estimator*** using ***d_filtered*** and ***c***.
- Identify columns in ***d_filtered*** ranked one by the ***estimator*** and store them in ***selected features.***

5: Return *selected_features.*

3.4. Training and Validation

In this step, training and validation of the proposed system takes place. Specifically, two distinct classification algorithms, DT and RF, are employed to develop the proposed IDS. Two training and validation scenarios are considered:

1. Using all features in the dataset.

2. Utilize only the top-ranked features identified through the previously introduced hybrid feature selection approach, comprising CA, MI, and RFECV.

The performance of the classifiers is evaluated based on the metrics: Accuracy, Precision, Recall, and F1-Score. These metrics are derived from the definitions: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) (Thomas et al., 2019). The confusion matrix in Table 3 describes these definitions associated with the *'Normal'* and *'Attack'* classes.

Table 3: Confusion Matrix

		Predicted Class	
		Normal	**Attack**
Actual Class	**Normal**	TN	FP
	Attack	FN	TP

- **Accuracy:** This metric provides the ratio of correctly predicted instances to the total number of instances in the dataset. It is a general indicator of a model's performance. Formula 4 shows how the Accuracy metric is calculated.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Instances} \quad (4)$$

- **Precision:** Precision focuses on the correctly predicted positive observations relative to the total predicted positives. It is also known as the Positive Predictive Value. Formula 5 shows how the Precision metric is calculated.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (5)$$

- **Recall:** Recall indicates the proportion of accurately predicted attack records relative to the overall count of records in the attack class. Formula 6 shows how the Recall metric is calculated.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (6)$$

- **F1-Score:** This is the weighted average of Precision and Recall, considering both FP and negatives. It ranges between 0 and 1, with a value closer to 1 indicating better classification performance. Formula 7 shows how the F1-Score metric is calculated.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (7)$$

In addition to the metrics above, the Fit time metric is used and determined by assessing the time needed for the classifier to complete the training process using the dataset, specifically through the fit() method available in the sci-kit-learn library in Python. A timer function is initiated before invoking the fit() method, which is then executed with the training data as its argument. The timer function ends once the fit() function is completed successfully. The time difference between the timer's start and stop is then used to determine how long the fit() method will take to finish.

4 Experiments and Results

This section delves into the different facets of the study, encompassing the experimental design, setup, and methodology. After that, a thorough summary of the experimental results is given, and the outcomes are thoughtfully analyzed. Furthermore, the study's limitations are emphasized.

4.1. Experimental Setup

Python 3.9.7 was used to implement the proposed IDS, which included the hybrid feature selection method, in a Jupyter Notebook environment. The feature selection, data processing, and visualization processes were made more accessible by libraries like Scikit-learn, Matplotlib, Pandas, and Numpy. The

computational work was performed on a desktop computer with an Intel(R) Core(TM) i7-1065G7 processor running at 1.5 GHz, 16 GB of RAM, and a Windows 11 Enterprise 64-bit operating system.

4.2. Experiments Design and Procedure

Preparing the data, preprocessing it, applying our proposed hybrid feature selection method, and training and validating the model on the UNSW-NB15 dataset are all included in the experimental process. Two different scenarios, DT and RF, are used to train and evaluate the classification models to answer the research questions presented in Section 1. Initially, they are trained using all security features available in the dataset. Subsequently, they are trained by employing the hybrid feature selection approach delineated in Section 3. The evaluation metrics and tools specified in Section 3 are applied in both cases. A comparative analysis of the results is conducted to ascertain the efficacy and performance enhancements the proposed hybrid approach provides. A thorough discussion of the findings is presented, contrasting them with previous work.

4.3. Experiments Results

In this study, the UNSW-NB15 dataset was employed, excluding the features *'state'*, *'dttl'*, and *'ct_state_ttl'*. This adjustment reduced the total number of features to 39. Experiments commenced with data preparation and preprocessing as outlined in Section 3.2, executed using Python. Subsequent sections provide a detailed exploration of the specific implementations of the proposed system.

Hybrid Feature Selection Implementation

The proposed hybrid approach, detailed in Algorithm 1 in Section 3.3, begins by assessing the linear relationships, as the first filter method, between each dataset feature and the label class through the correlation coefficient. Focusing on linear relationships, this method swiftly identifies features highly correlated with the label class. Figure 2 presents the correlation scores for all features.

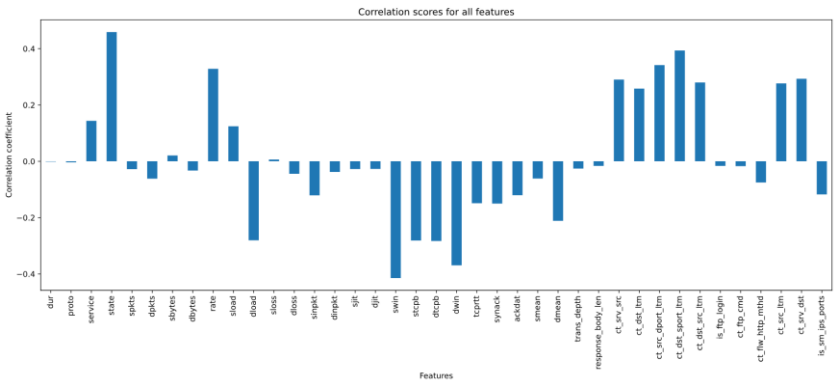

Figure 2: Correlation Scores for all Features

Notably, upon analyzing the correlation scores for all features, a threshold of 0.1 was set for this method. This value may vary based on the specific dataset under consideration. Features with correlation scores, either positive or negative, exceeding this threshold are selected. Figure 3 shows the result of this method. The features selected are 22 as follows: *'service'*, *'state'*, *'rate'*, *'sload'*, *'dload'*, *'sinpkt'*, *'swin'*, *'stcpb'*, *'dtcpb'*, *'dwin'*, *'tcprtt'*, *'synack'*, *'ackdat'*, *'dmean'*, *'ct_srv_src'*, *'ct_dst_ltm'*, *'ct_src_dport_ltm'*, *'ct_dst_sport_ltm'*, *'ct_dst_src_ltm'*, *'ct_src_ltm'*, *'ct_srv_dst'*, *'is_sm_ips_ports'*.

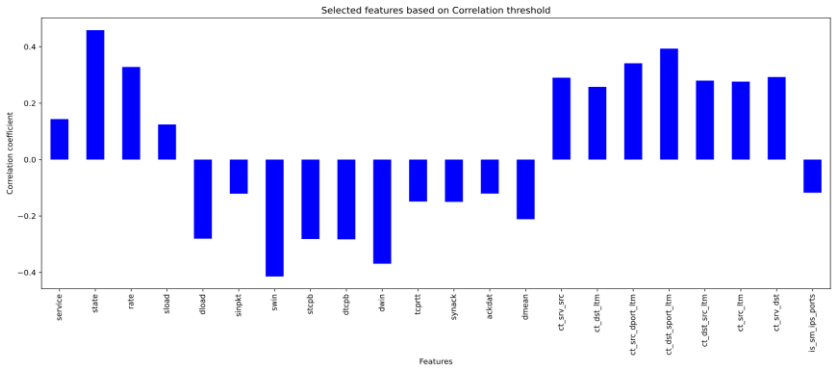

Figure 3: Selected Features with Correlation Threshold

Following the initial filter method, the filtering begins by computing the MI scores for the remaining features, as detailed in Algorithm 1. Features already selected by the first method are not considered in this phase, optimizing the speed of MI computations. A threshold of 0.3 was set for this method, which could differ based on the dataset in use. Figure 4 shows the MI scores for the remaining features, while Figure 5 illustrates the selected features based on the MI threshold.

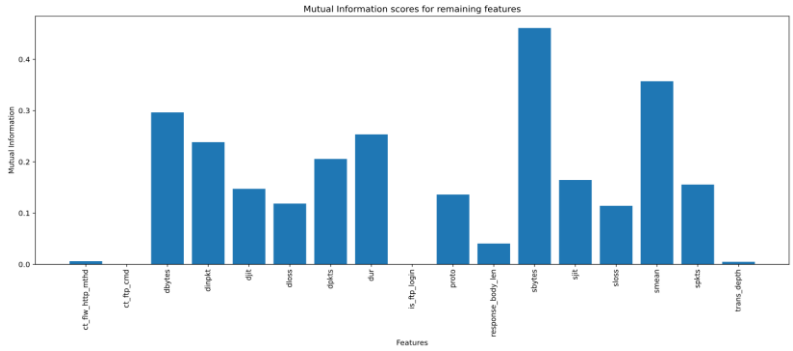

Figure 4: MI Scores for the Remaining Features

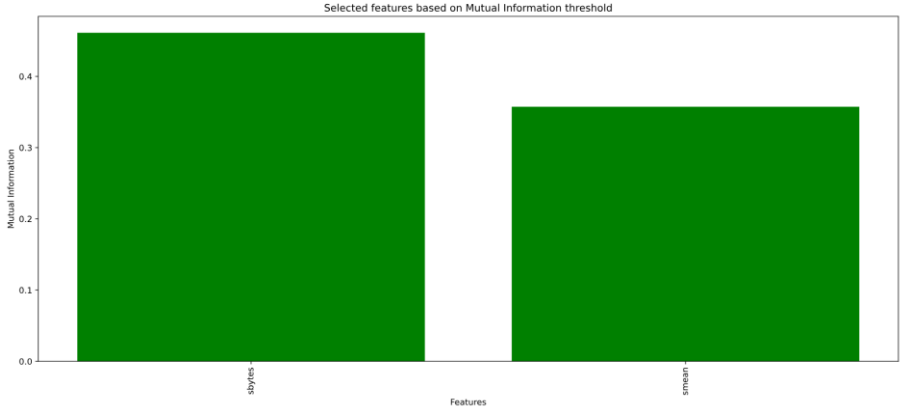

Figure 5: Selected Features based on the MI Threshold

As illustrated in Figure 5, only two features, *'sbytes' and 'smean'*, are selected. These features are combined with those obtained through the correlation method, resulting in 24 combined features. Such obtained features are further processed by the wrapper method explained in Section 3.3. As stated before, the RFECV was applied using the RF Classifier as the estimator. Furthermore, 10-fold Cross-Validation was conducted to ascertain the reliability of the classifier. The combined feature dataset from the filter

methods was partitioned into ten subsets. The process was iterated ten times, using each subset precisely once as a validation set. Subsequently, the results were averaged to yield a singular estimate for every metric. This offers a deeper understanding of the efficacy of the chosen features in distinguishing between the *'Normal'* and *'Attack'* classes. It is crucial to highlight that the execution of the selected wrapper method on the dataset's entire feature set is notably slower than when applied to the combined features derived from the filter mentioned above methods. Interestingly, the resultant set of features from the proposed hybrid approach comprises a concise list of 22 distinct features. These are: *'smean', 'ct_dst_ltm', 'sinpkt', 'tcprtt', 'ackdat', 'sload', 'ct_dst_src_ltm', 'sbytes', 'ct_srv_src', 'service', 'stcpb', 'dmean', 'ct_src_dport_ltm', 'ct_src_ltm', 'ct_dst_sport_ltm', 'swin', 'dload', 'synack', 'rate', 'state', 'ct_srv_dst', 'sbytes'*.

Classification and Prediction Implementation

To this end, the features derived from the proposed hybrid approach are now ready to be fed into the classification models chosen for this study. In this study, two distinct experiments were undertaken:

1. **Experiment 1:** The DT classifier was applied to the entire set of features. Subsequently, its performance was compared when applied only to the selected features.
2. **Experiment 2:** The RF classifier was similarly applied to all the features. Its results were then compared when the classifier was applied solely to the selected features.

The evaluation metrics outlined in Section 3.4 were employed across both experiments to evaluate comprehensively the feature selection approach proposed in this paper.

Experiment 1

In this experiment, the DT classifier was first trained and tested using all 39 security features in the dataset. In the subsequent phase, the same classifier was trained and tested, but only with the features selected by the proposed hybrid selection approach. The results of this experiment are shown in Table 4.

Table 4: Results of Experiment 1

Evaluation metric	**Prediction using the DT classifier**	
	All features (39)	**Selected features (22)**
Accuracy	96.42%	96.54%
Precision	96.42%	96.54%
F1-Score	96.42%	96.54%
Recall	96.42%	96.54%
Fit Time	0.94 seconds	0.5 seconds

As shown in Table 4 above, it was observed that all evaluation metrics showed slight improvements when utilizing the selected features from the proposed hybrid approach. Notably, the DT classifier exhibited a reduced processing time when operating with the selected features for the *'Fit Time'* metric. This reduction in the number of features and computational time mitigates the Decision Tree's propensity for overfitting, lessens computational complexity, and enhances the accuracy of intrusion prediction. Figure 6 and Figure 7 illustrate the Confusion Matrix of Experiment 1.

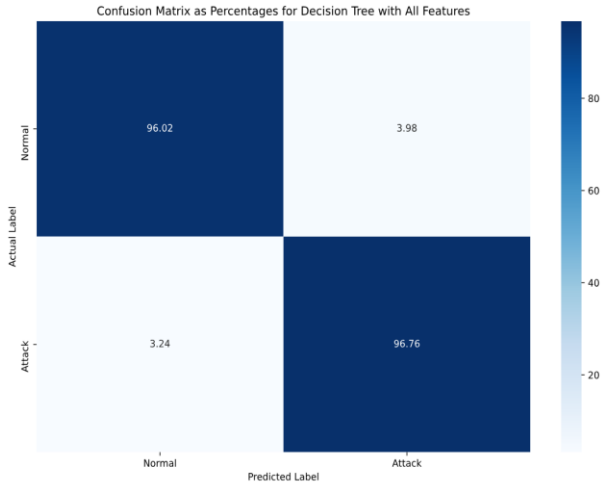

Figure 6: Confusion Matrix for DT with all Features

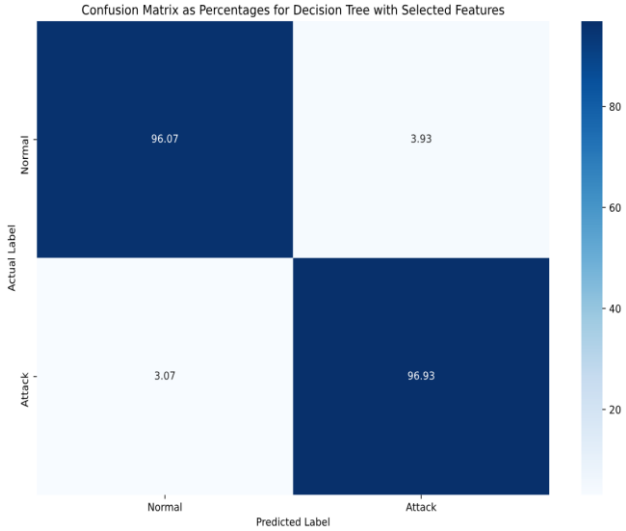

Figure 7: Confusion Matrix for DT with Selected Features

Comparing the results illustrated in Figure 6 with those in Figure 7, it becomes evident that the proposed hybrid approach improves the values of TP, TN, FP, and FN. Notably, the FN (where the actual category is *'Attack'* but predicted as *'Normal'*) dropped from 3.24 with all features to 3.07 with the selected ones. Simultaneously, the TP (where both actual and predicted categories are *'Attack'*) enhanced from 96.76 using all features to 96.93 when employing the selected features. Such results underscore the capability of the proposed approach to enhance attack prediction accuracy with DT classification.

Experiment 2

In this experiment, the RF classifier was initially trained and tested using all 39 available security features from the dataset. Subsequently, this classifier was trained and tested again, but only with the features identified by the proposed hybrid selection approach. The outcomes of this experiment are detailed in Table 5.

Table 5: Results of Experiment 2

Evaluation metric	Prediction using the RF classifier	
	All features (39)	Selected features (22)
Accuracy	97.76%	97.81%
Precision	97.76%	97.81%
F1-Score	97.76%	97.81%
Recall	97.76%	97.81%
Fit Time	9.90 seconds	7.00 seconds

Table 5 above highlights that using the selected features from the hybrid approach led to marginal enhancements across all evaluation metrics for the RF classifier. Notably, the RF classifier exhibited a reduced processing time when operating with the selected features for the *'Fit Time'* metric. The Confusion Matrix for this experiment is depicted in Figure 8 and Figure 9.

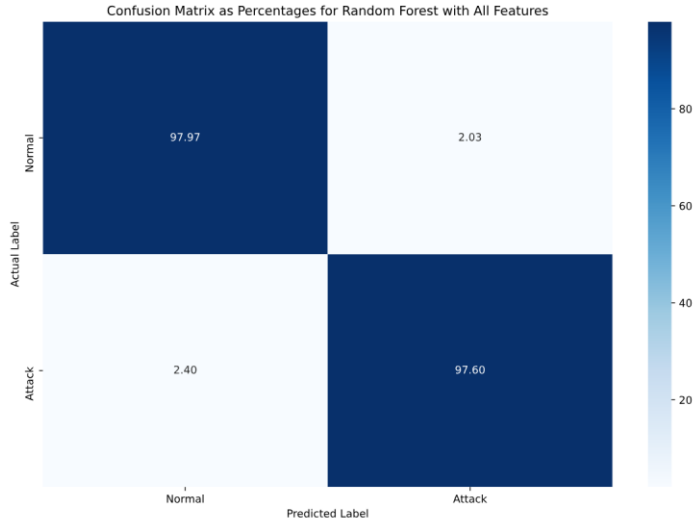

Figure 8: Confusion Matrix for RF with all Features

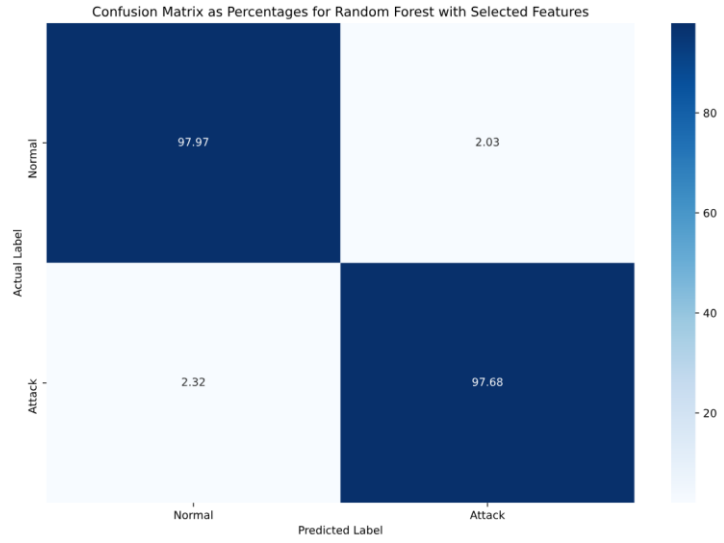

Figure 9: Confusion Matrix for RF with Selected Features

Analyzing the data in Figure 8 against that in Figure 9 reveals that the hybrid approach has boosted the performance of TP and reduced FN values. Specifically, FN decreased from 2.4 (using all features) to 2.3 (using selected features). Meanwhile, TP improved from 97.60 using the complete feature set to 97.68 with the selected features. This highlights the potential of this method to refine the accuracy of attack prediction when integrated with RF classification.

In light of the results, the proposed hybrid approach demonstrably enhanced the performance of both classifiers, specifically in terms of Accuracy, Precision, Recall, F1-Score, and Fit Time. This strategy can be easily adapted with tree-based classifiers to strengthen cyber-attack predictions and reinforce IDSs. Contextualizing the findings of this study, it is instructive to compare them with related works, particularly those leveraging the UNSW-NB15 dataset. As referenced in Section 2, Awad and Fraihat (2023) introduced a feature selection technique termed RFECV using a DT estimator. Their methodology identified 15 pivotal features integrated into various prediction models. Yet, they applied their selection technique to all 39 features, potentially increasing computational time. This could compromise the efficiency of an IDS when handling vast dimensions of features. By contrast, the current study applied only 22 out of 39 features to the RFECV within the hybrid approach.

Interestingly, even with a larger set of 22 features, the RF classifier's performance in this research surpassed that in their study. Al-Omari et al. (2021) proposed a sophisticated tree-based model for cyber-attack predictions. They used a single feature selection method based on the Gini score and validated it with a DT classifier. However, their analysis encompassed all dataset features, including the potentially biasing TTL-based attributes *'sttl'*, *'dttl'* and *'ct_state_ttl'* from the UNSW-NB15 dataset. Among similar works using tree-based models, the DT and RF classifiers with the proposed hybrid feature selection approach outperform the methods proposed in previous work. The proposed approach identifies 22 optimal features, enhances the performance of tree-based models in terms of accuracy and F1-Score, and reduces the fit-time metric. Table 6 summarizes the performance of previous methods compared to our proposed hybrid approach.

Table 6: Performance Comparison with Some Existing Works on the Same Dataset

Work	classifier	Feature selection method	No. of Features obtained	F1-Score	Accuracy
(Barkah et al., 2023)	DT RF	RFE	13	86.87% 85.68%	85.64% 85.07%
(Al-Omari et al., 2021)	DT	Gini score	19	97%	96.72%
(Awad & Fraihat, 2023)	RF	DT-RFECV	15	95.29%	95.30%
(Alissa et al., 2022)	DT	-	34	94%	94%
Proposed Hybrid Approach	DT RF	CA, MI, RFECV	22	**96.54%** **97.81%**	**96.54%** **97.81%**

4.4. Limitations

The proposed IDS has a hybrid feature selection approach which has shown promise in improving prediction accuracy and reducing computational complexity. However, its performance across a wider range of datasets is yet to be thoroughly examined. Validating the system on various datasets is crucial to understand its adaptability and reliability in diverse network environments. Additionally, integrating a broader spectrum of classification models, including advanced machine learning and deep learning

algorithms, would provide valuable insights into the versatility and effectiveness of the proposed hybrid approach in different contexts.

Furthermore, there are practical implementation challenges that need to be addressed, such as scalability, adaptability to evolving threats, and integration with existing network security infrastructures. To further validate the utility of the proposed IDS, it is necessary to explore its real-world applicability in various industry sectors, each with distinct cybersecurity requirements. Understanding how the system can be implemented and function effectively alongside other security components is essential for assessing its potential for practical deployment and widespread adoption in complex network environments.

5 Conclusion and Future Work

This study applied a hybrid feature selection approach for tree-based IDS to the UNSW-NB15 intrusion detection dataset. The hybrid approach comprises three distinct feature selection techniques: CA employing the Pearson coefficient, MI as a filter method, and RFECV as a wrapper method. The filtering methods identified 24 features from the initial set of 39 based on predetermined threshold values. The wrapper method subsequently reduced these selected features to 22 optimal features. Experimental results show that the proposed hybrid feature selection approach can effectively and efficiently identify the most relevant features, thereby enhancing the performance of tree-based IDSs.

Furthermore, the proposed system demonstrated better performance compared to existing similar work. The developed intrusion detection framework can be integrated into existing prediction models and offer additional insights for researchers concerning applying hybrid feature selection techniques to improve cyberattack prediction in complex network environments. Future work will extend the application of this hybrid feature selection approach to other intrusion detection datasets and incorporate additional ML algorithms.

Declarations

- Funding: Not applicable
- Conflicts of interest: The authors declare that they have no conflict of interest.
- Data Availability Statement: Not applicable
- Code availability All experiments in this research were implemented in Jupyter Notebook with Python using predefined ML packages and libraries.

References

[1] Al-Omari, M., Rawashdeh, M., Qutaishat, F., Alshira'H, M., & Ababneh, N. (2021). An Intelligent Tree-Based Intrusion Detection Model for Cyber Security. *Journal of Network and Systems Management*, *29*(2), 1–18. https://doi.org/10.1007/S10922-021-09591-Y

[2] Alissa, K., Alyas, T., Zafar, K., Abbas, Q., Tabassum, N., & Sakib, S. (2022). Botnet Attack Detection in IoT Using Machine Learning. *Computational Intelligence and Neuroscience*, *2022*. https://doi.org/10.1155/2022/4515642

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

[4] Aqlan, A., Saif, A., & Salh, A. (2023). Role of IoT in Urban development. *International Journal of Communication and Computer Technologies (IJCCTS), 11*(2), 13-18.

[5] Asif, M., Barnaba, M., Babu, K.R., Prakash, P.O., & Khamuruddeen, D.S. (2021). Detection And Tracking of Theft Vehicle. *International Journal of Communication and Computer Technologies (IJCCTS)*, *9*(2), 6-11.

[6] Awad, M., & Fraihat, S. (2023). 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), 1-23. https://doi.org/10.3390/JSAN12050067

[7] Barkah, A.S., Selamat, S.R., Abidin, Z.Z., & Wahyudi, R. (2023). Impact of Data Balancing and Feature Selection on Machine Learning-based Network Intrusion Detection. *JOIV : International Journal on Informatics Visualization*, *7*(1), 241–248. https://doi.org/10.30630/JOIV.7.1.1041

[8] Buczak, A.L., & Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys and Tutorials*, *18*(2), 1153–1176. https://doi.org/10.1109/COMST.2015.2494502

[9] Camgozlu, Y., & Kutlu, Y. (2023). Leaf Image Classification Based on Pre-trained Convolutional Neural Network Models. *Natural and Engineering Sciences, 8*(3), 214-232.

[10] Cover, T.M., & Thomas, J.A. (2012). *Elements of Information Theory*. Wiley.

[11] Fraihat, S., Makhadmeh, S., Awad, M., Al-Betar, M.A., & Al-Redhaei, A. (2023). Intrusion detection system for large-scale IoT NetFlow networks using machine learning with modified Arithmetic Optimization Algorithm. *Internet of Things*, *22*, 100819. https://doi.org/10.1016/J.IOT.2023.100819

[12] Jackson, E., & Agrawal, R. (2019). Performance Evaluation of Different Feature Encoding Schemes on Cybersecurity Logs. *Conference Proceedings - IEEE SOUTHEASTCON*, *2019-April*. https://doi.org/10.1109/SOUTHEASTCON42311.2019.9020560

[13] Kaggle. (2023). https://www.kaggle.com/

[14] Jelena, T., & Srđan, K. (2023). Smart Mining: Joint Model for Parametrization of Coal Excavation Process Based on Artificial Neural Networks. *Arhiv za tehničke nauke, 2*(29), 11-22.

[15] Kasongo, S.M., & 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), 1–20. https://doi.org/10.1186/S40537-020-00379-6/TABLES/8

[16] Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., & Alazab, A. (2020). Hybrid Intrusion Detection System Based on the Stacking Ensemble of C5 Decision Tree Classifier and One Class Support Vector Machine. *Electronics*, *9*(1), 1-18. https://doi.org/10.3390/ELECTRONICS9010173

[17] Kuhn, M., & Johnson, K. (2013). Applied predictive modeling. *In Applied Predictive Modeling* (1st ed.). Springer New York, NY. https://doi.org/10.1007/978-1-4614-6849-3/COVER

[18] Maniriho, P., Mahoro, L.J., Niyigaba, E., Bizimana, Z., & Ahmad, T. (2020). Detecting Intrusions in Computer Network Traffic with Machine Learning Approaches. *International Journal of Intelligent Engineering and Systems*, *13*(3), 433-445. https://doi.org/10.22266/ijies2020.0630.39

[19] Marangunic, C., Cid, F., Rivera, A., & Uribe, J. (2022). Machine Learning Dependent Arithmetic Module Realization for High-Speed Computing. *Journal of VLSI Circuits and Systems*, *4*(1), 42-51.

[20] Mishra, P., Varadharajan, V., Tupakula, U., & Pilli, E.S. (2019). A detailed investigation and analysis of using machine learning techniques for intrusion detection. *IEEE Communications Surveys and Tutorials*, *21*(1), 686–728. https://doi.org/10.1109/COMST.2018.2847722

[21] Muralidharan, J. (2020). Wideband Patch Antenna for Military Applications. *National Journal of Antennas and Propagation (NJAP)*, *2*(1), 25-30.

[22] Nisioti, A., Mylonas, A., Yoo, P.D., & Katos, V. (2018). From intrusion detection to attacker attribution: A comprehensive survey of unsupervised methods. *IEEE Communications Surveys*

and Tutorials, *20*(4), 3369–3388. https://doi.org/10.1109/COMST.2018.2854724

[23] Patgiri, R., Varshney, U., Akutota, T., & Kunde, R. (2019). An Investigation on Intrusion Detection System Using Machine Learning. *Proceedings of the IEEE Symposium Series on Computational Intelligence, SSCI 2018*, 1684–1691. https://doi.org/10.1109/SSCI.2018.8628676

[24] Park, M., You, G., Cho, S.J., Park, M., & Han, S. (2019). A Framework for Identifying Obfuscation Techniques applied to Android Apps using Machine Learning. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 10*(4), 22-30.

[25] Press, W.H., Teukolsky, S.A., Vetterling, W.T., & Flannery, B.P. (1986). *Numerical Recipes: The Art of Scientific Computing*. Cambridge UP Cambridge etc.

[26] Priyanka, J., Ramya, M., & Alagappan, M. (2023). IoT Integrated Accelerometer Design and Simulation for Smart Helmets. *Indian Journal of Information Sources and Services, 13*(2), 64–67.

[27] Raju, V.N.G., Lakshmi, K.P., Jain, V.M., Kalidindi, A., & Padma, V. (2020). Study the Influence of Normalization/Transformation process on the Accuracy of Supervised Classification. *Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020*, 729–735. https://doi.org/10.1109/ICSSIT48917.2020.9214160

[28] Raviteja, P., Satya Venkata, M., Devi, S., Gowri, M., Vamsi, M., Krishna, S., & Prabhakar, V.S. (2020). Implementation Of Machine Learning Algorithms For Detection Of Network Intrusion. *International Journal of Computer Science Trends and Technology (IJCST), 8*(2), 163-169. www.ijcstjournal.org

[29] Roulston, M.S. (1999). Estimating the errors on measured entropy and mutual information. *Physica D: Nonlinear Phenomena*, *125*(3–4), 285–294. https://doi.org/10.1016/S0167-2789(98)00269-3

[30] Salim, Q.M., and Mohammed, A.E.H. (2023). Reducing False Negative Intrusions Rates of Ensemble Machine Learning Model based on Imbalanced Multiclass Datasets. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 14*(2), 12-30.

[31] Sarhan, M., Layeghy, S., & Portmann, M. (2021). *Feature Analysis for Machine Learning-based IoT Intrusion Detection*. https://arxiv.org/abs/2108.12732v2

[32] Steephen, K., Abinaya, S., Aruna, S., Deepika, P., & Gowthami, G. (2022). An intelligent car parking using IoT with node MCU module. *International Journal of Communication and Computer Technologies (IJCCTS), 10*(2), 88-92.

[33] Thakkar, A., & Lohiya, R. (2021). Attack classification using feature selection techniques: a comparative study. *Journal of Ambient Intelligence and Humanized Computing*, *12*(1), 1249–1266. https://doi.org/10.1007/S12652-020-02167-9/METRICS

[34] Thakkar, A., & Lohiya, R. (2022). A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions. *Artificial Intelligence Review*, *55*(1), 453–563. https://doi.org/10.1007/S10462-021-10037-9/METRICS

[35] Thang, N. C., & Park, M. (2020). Detecting Malicious Middleboxes In Service Function Chaining. *Journal of Internet Services and Information Security, 10*(2), 82-90.

[36] Thomas, T., Vijayaraghavan, A.P., & Emmanuel, S. (2019). *Machine Learning Approaches in Cyber Security Analytics*. Springer.

[37] Tulasi Bhavani, T., Rao, M.K., & Reddy, A.M. (2020). Network intrusion detection system using random forest and decision tree machine learning techniques. *Advances in Intelligent Systems and Computing*, *1045*, 637–643. https://doi.org/10.1007/978-981-15-0029-9_50/COVER

[38] Yin, Y., Jang-Jaccard, J., Xu, W., Singh, A., Zhu, J., Sabrina, F., & Kwak, J. (2023). IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset. *Journal of Big Data*, *10*(1), 1–26. https://doi.org/10.1186/S40537-023-00694-8/TABLES/9

[39] Zhao, Y., Hu, N., Zhang, C., & Cheng, X. (2020). DCG: A Client-side Protection Method for DNS Cache. *Journal of Internet Services and Information Security*, *10*(2), 103-121.
[40] Zheng, A., & Casari, A. (2018). *Feature Engineering for Machine Learning*. O'Reilly Media.

Authors Biography

Mohammad Al-Omari is an assistant professor at the Department of Business Information Technology at Princess Sumaya University for Technology, Jordan. He received his BSc. in Computer Information Systems from Jordan University of Science and Technology (JUST) in 2006. He received his PhD in Computer Science from De Montfort University, United Kingdom, in 2017. He has a variety of academic and professional qualifications and experience. His research interests include, but are not limited to, Machine Learning, Cyber Security, Business Analytics, Data Mining, Big Data, and Cloud Computing.

Qasem Abu Al-Haija received his Ph.D. from Tennessee State University (TSU), USA, in 2020. He is an Assistant Professor at the Department of Cybersecurity, Faculty of Computer & Information Technology, Jordan University of Science and Technology, Irbid, Jordan. He authorizes more than 200 scientific research papers and book chapters. His research interests include Artificial Intelligence (AI), Cybersecurity and Cryptography, the Internet of Things (IoT), Cyber-Physical Systems (CPS), Time Series Analysis (TSA), and Computer Arithmetic. Recently, he was listed as one of the world's top 2% of scientists list released publicly by Stanford University and Elsevier Publisher.