Enhancing Cybersecurity: Innovative Hybrid Feature Selection for Intrusion Detection

Guntoro Guntoro[1], Lisnawita Lisnawita[2], Loneli Costaner[2]

{guntoro@unilak.ac.id[1], lisnawita@unilak.ac.id[2], lonelicostaner@unilak.ac.id[2]}

Informatics of Engineering, Universitas Lancang Kuning, Pekanbaru, Indonesia

Abstract. Network security has evolved into a major issue that necessitates careful consideration in the present digital world. Intrusion Detection Systems (IDS) are critical in detecting and preventing network intrusions. In this work, we propose a feature selection and majority vote based intrusion detection method. Using the Majority Voting approach, an intrusion detection system with many models is created. The proposed method was tested using the NSL-KDD benchmark dataset. To improve the model's performance, we combined the BestFirst search strategy with the Correlation Feature Selection (CFS) technique. This strategy successfully reduced the number of available features from 41 to 12 while maintaining detection accuracy. The experiment findings reveal that the suggested model has an accuracy rate of 96.49%, indicating the method's worth in selecting the most relevant and instructive features for the classification operation. This research contributes significantly to the development of more efficient and effective intrusion detection systems by emphasizing the role of feature selection in improving classification model performance in detecting network security threats.

Keywords: Intrusion Detection System, Correlation Feature Selection (CFS), BestFirst, Majority Voting

1 Introduction

In the ever-changing environment of cyber threats, the function of Intrusion Detection Systems (IDS) in network security has grown critical. IDS are critical for monitoring network traffic, detecting strange patterns, and notifying administrators of potential security breaches. However, increasing network traffic complexity and volume pose substantial problems, necessitating developments in IDS capabilities [1].

Machine Learning (ML) has emerged as a powerful method for enhancing the performance of intrusion detection systems (IDS). Large datasets may be automatically analyzed using ML algorithms, facilitating the early identification of irregularities and possible intrusions. IDS prefers Random Forest, an ensemble learning technique that is well-known for its excellent accuracy and capacity to manage large feature spaces in classification problems [2].

Despite the strengths of ML-based IDS, the curse of [3] due to high feature dimensions can impede performance, leading to prolonged processing times and potential misclassifications. Among the crucial pre-processing steps to address these problems is feature selection, which aims to separate and preserve the most useful characteristics from the redundant and unnecessary ones [4].

ICEEES 2023, October 30-31, Pekanbaru, Indonesia
Copyright © 2024 EAI
DOI 10.4108/eai.30-10-2023.2343092

Correlation Feature Selection (CFS) has become popular as a useful technique for feature selection [5]. It determines the value of a feature subset by taking into account each feature's unique predictive capacity as well as the degree of duplication among them. The integration of CFS with search algorithms [6] like BestFirst further enhances its ability to navigate the feature space efficiently, leading to optimal feature subsets [7].

Recent studies have underscored the significance of feature selection in IDS, highlighting its impact on reducing dimensionality, accelerating processing times, and improving classification accuracy [3]. This promising direction in IDS research aligns with the broader trend of adopting hybrid approaches, combining the strengths of ML algorithms and intelligent feature selection methods [8]. Such approaches not only bolster the security posture of networks but also contribute to the development of scalable and efficient IDS solutions capable of adapting to the dynamic nature of cyber threats [9] [1] .

In conclusion, the combination of Machine Learning, and in particular Random Forest and Support Vector Machine (SVM), with sophisticated feature selection approaches like CFS and BestFirst, represents a critical development in the field of network security. This synergy offers the potential to greatly boost the effectiveness of IDS, guaranteeing robust network security in the face of ever-evolving cyber threats.

2 Related Work

The application of Machine Learning techniques in Intrusion Detection Systems (IDS) has been the focus of substantial study in recent years, with the purpose of boosting the accuracy and efficiency of threat detection in network security. Various machine learning algorithms, including as Decision Trees, Support Vector Machines, Neural Networks, Random Forest, and others, have been researched for use in IDS [2] [4]. To solve the problem of high-dimensional datasets in IDS, feature selection methodologies like as Principal Component Analysis (PCA) and Recursive Feature Elimination (RFE) have been deployed [10]. Random Forest, noted for its excellent accuracy and capacity to handle huge feature areas, has been frequently employed in IDS [11]. For feature selection in IDS, Correlation Feature Selection (CFS) and BestFirst search have been merged, resulting in reduced feature dimensions while maintaining excellent detection accuracy [4]. To increase classification performance and reduce false positive rates, hybrid approaches combining machine learning algorithms with heuristic methods for feature selection have also been developed [12].

Finally, machine learning and feature selection methodologies have been widely and diversely used in IDS. A constant theme in these studies is the pursuit of increased detection capabilities, reduced processing overhead, and the capacity to adapt to changing network parameters. Combining Random Forest with advanced feature selection methods like CFS and BestFirst has emerged as a viable path for developing more efficient and robust IDS solutions.

3 Proposed Method

The objective of our proposed method is to enhance the capability of Intrusion Detection Systems (IDS) in identifying potential network threats with high accuracy and efficiency. To achieve this, we integrate Machine Learning techniques with intelligent feature selection methods. Figure 1 illustrates the overall workflow of our proposed method.

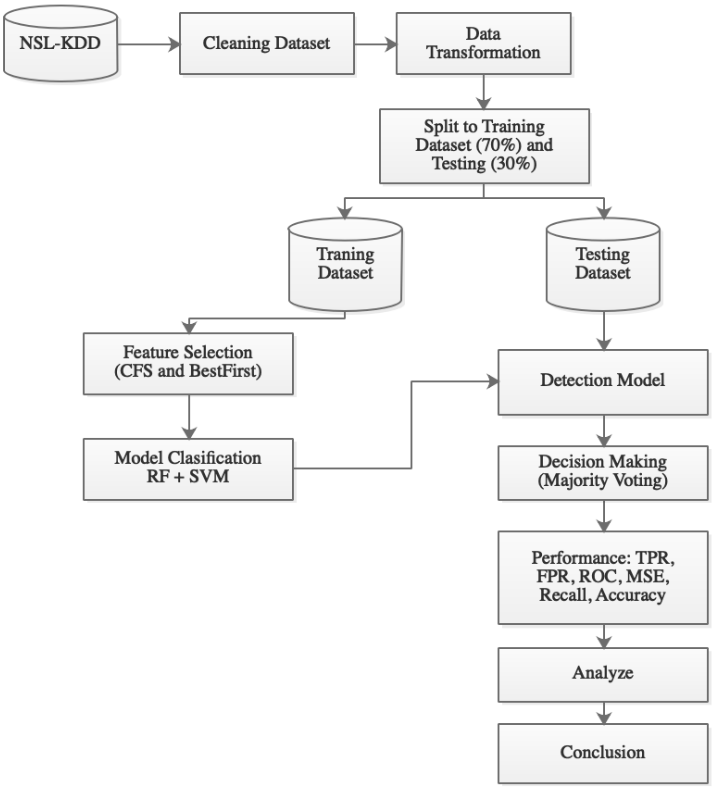
NSL-KDD
Cleaning Dataset
Data Transformation
Split to Training Dataset (70%) and Testing (30%)
Traning Dataset
Testing Dataset
Feature Selection (CFS and BestFirst)
Model Clasification RF + SVM
Detection Model
Decision Making (Majority Voting)
Performance: TPR, FPR, ROC, MSE, Recall, Accuracy
Analyze
Conclusion

Figure 1. Proposed Method

3.1 Data Preprocessing

The first stage involves preparing the dataset for training and evaluation. We utilize the NSL-KDD dataset, a well-known benchmark in network security research. The dataset is preprocessed to handle any missing values, and we encode categorical variables to ensure compatibility with ML algorithms.

3.2 Feature Selection

Network datasets typically comprise a plethora of features, among which some might be irrelevant or carry minimal information. These particular features tend to have a negligible impact on the outcomes of classification tasks. The primary function of a feature selection algorithm is to identify and remove these superfluous features, subsequently diminishing their influence on the performance of the classification algorithm.

3.2.1 Correlation Feature Selection Algorithm (CFS)

The Correlation Feature Selection (CFS) algorithm, denoted as [13] in your text, primarily operates by identifying subsets of features, taking into account the redundancy present among these features. The aim is to discover subsets of features that exhibit high internal correlation while maintaining a low correlation with each other. This approach surpasses the limitations of single-variable screening, as it thoroughly evaluates the interdependencies among features. By doing so, CFS is able to efficiently discard features that are irrelevant or do not contribute meaningfully to the model. The evaluation function of CFS for feature subsets is defined as follows:

$$Merit_s \frac{k r_{\bar{c}\,f}}{\sqrt{k+(k-1)r_{\bar{f}\,f}}} \tag{1}$$

The term $Merit_s$ represents a heuristic 'merit‘ score for evaluating the quality of a selected feature subset S containing k features. This merit score is crucial in determining how well the selected features contribute to the performance of the classification model.

The component $r_{\bar{f}\,f}$ stands for the average correlation between the features and the class, essentially measuring the relevance of each feature in the subset to the target class. A higher value of $r_{\bar{f}\,f}$ indicates that the features in the subset have a strong relationship with the class, implying that they are important for making accurate predictions.

On the other hand, $r_{\bar{f}\,f}$ represents the average correlation among the features themselves, denoting the redundancy within the feature subset. The goal here is to minimize $r_{\bar{f}\,f}$, as high redundancy among features can lead to overfitting and can negatively affect the model’s performance. The Pearson correlation coefficient $r_{\bar{f}\,f}$ is used to calculate both $r_{\bar{f}\,f}$ and $r_{\bar{f}\,f}$, providing a measure of linear correlation between variables. In order to ensure consistency and reliability in the calculations, all variables involved need to be normalized. In summary, the CFS algorithm utilizes the $Merit_s$ score to find a feature subset that maximizes feature-class relevance while minimizing feature-feature redundancy, with the aim of enhancing the classification model’s performance. The use of normalized variables and the Pearson correlation coefficient ensures a standardized approach to evaluating the feature subsets.

3.2.2 Best First Algorithm

One well-known method in the fields of computer science and artificial intelligence is best first search, which is used for graph traversal and pathfinding problems.. It operates on the principle of making the most informed decision at each step of the traversal, employing a heuristic function to estimate the cost or distance from the current node to the target. Unlike uninformed search algorithms that may traverse the graph blindly, Best First Search leverages additional knowledge in the form of the heuristic to guide its search, aiming to reach the goal in a more efficient manner.

3.3 Classification Methods

In this step, the intrusion detection system classification models are developed using Random Forest (RF) and Support Vector Machine (SVM). The purpose of this stage is to test the correctness of the suggested categorization models.

3.3.1 Random Forest

A random forest classifier stands out as a highly adaptable and straightforward supervised machine learning algorithm, delivering outstanding performance even in the absence of extensive parameter adjustment. Its ease of use and widespread acceptance are amplified by its capability to handle a variety of tasks, predominantly classification and regression, which represent the bulk of contemporary machine learning applications. Random forests operate as an ensemble learning technique, generating numerous decision trees and amalgamating them to yield a more precise and consistent prediction of the data's category. In doing so, random forests successfully address the tendency of decision trees to overfit the training data.

3.3.2 Support Vector Machine (SVM)

The Support Vector Machine (SVM) is recognized as a highly versatile machine learning model capable of handling both classification and regression tasks, positioning it as one of the predominant models within the field of machine learning research. The primary aim of SVM is to segregate a provided dataset into distinct categories, striving to identify the most optimal hyperplanes for this purpose. One of the notable advantages of SVM is its proficiency in dealing with high-dimensional input spaces, delivering effective performance. Additionally, SVM offers the flexibility to choose from an array of Kernel functions during the decision-making process, enhancing its adaptability. However, a potential drawback of SVM is its demand for meticulous parameter tuning, a necessity that becomes particularly pronounced when dealing with scenarios where the input dimension surpasses the number of samples available.

3.3.3 Majority Voting

Voting is a fundamental group approach that frequently shows to be quite successful. It can be applied to problems involving both regression and classification, making it flexible. Using this method, a model is broken down into two or more sub-models—in this case, five. Following that, predictions from every sub-model are combined using a majority voting method. Figure 2 depicts the process of majority voting. This method functions as a meta-classifier by using a majority vote to determine if two machine learning classifiers are similar or different from one another. The majority vote approach is applied to select the final class label, pinpointing the label most commonly predicted by the classification models. The class label *y* is ascertained using equation (7), taking into account the majority vote from each classifier *Cj*

3.4 Evaluation Measures

Following the classification model stage, the next step involves testing the performance of the proposed anomaly detection methods. Using accuracy metrics including (i) sensitivity and specificity, (ii) misclassification rate, (iii) confusion matrix entries, (iv) precision-recall, and F-measures, a number of intrusion detection studies assess performance. For accuracy evaluation, a confusion matrix is typically used, as shown in Table 2.

Table 1. Confusion Matrix

Actual	**Prediction**	
	Class 1	**Class 2**
Class 1	True Positive (TP)	False Positive (FP)
Class 2	False Negative (FN)	True Negative (TN)

In Table 2, the following explanations are provided:

a. False Positive (FP): The number of actual normal instances that were incorrectly detected as attacks.
b. False Negative (FN): Prediction errors where actual attacks were incorrectly classified as normal.
c. True Positive (TP): Correct predictions where actual normal instances were accurately classified as normal.
d. True Negative (TN): Instances of actual attacks that were accurately classified as attacks.

To test the accuracy of the intrusion detection system in this research, the following equation is employed:

a. Accuracy (Ac): The degree of closeness between the classification results and the actual values.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2}$$

b. Detection Rate (DR) or Recall: The proportion of actual positives correctly categorized as the positive class.

$$DR = \frac{TP}{TP+FN} \tag{3}$$

c. False Positive Rate: The proportion of actual negatives incorrectly categorized as the positive class.

$$FPR = \frac{FP}{FP+TN} \tag{5}$$

For assessing the performance of the intrusion detection system classification model in the research, the following metrics are used: TPR (True Positive Rate), FPR (False Positive Rate), Accuracy, and processing or computation time. The processing or computation time is calculated from the moment the classification model starts running until the process is completed.

4 Result and Discussion

In order to minimize the dimensionality of the data utilized for the model's training and testing, features from the NSL-KDD dataset were picked using the CFS (Correlation Feature Selection) and BestFirst search techniques. Effective techniques to shorten training times and simplify models are CFS and BestFirst. Based on the analysis of the data, this method worked well in the current model, albeit it might not be useful in other situations.

Here, 111,386 and 37,129 samples from the NSL-KDD dataset were used to train and evaluate a range of classifiers, including Random Forest (RF) and Support Vector Machine (SVM). The model's performance was examined using a range of parameters. The figures below demonstrate the findings graphically in terms of accuracy, precision, recall, and F1-score across multiple feature subsets.

It is important to note that while random feature selection can reduce training time and model complexity, it may not always yield the optimal model. This is because some features may be more important than others for making accurate predictions, and random selection could potentially omit these crucial features. Therefore, it is crucial to perform a comprehensive performance evaluation to ensure that the model can still make accurate predictions even with a reduced number of features.

Following the pre-processing processes, the dataset was split into two halves and 12 important characteristics were identified. testing and training data. The performance was tested based on numerous metrics utilizing both all characteristics and the chosen features, as shown in Table 2

Table 2. Feature of Selected

Number	Number of Feature	Feature	Accuracy
1	2	protocol_type	100 %
2	3	service	100 %
3	4	flag	100 %
4	5	src_bytes_binarized	100 %
5	6	dst_bytes_binarized	100 %
6	7	land_binarized	100 %
7	8	wrong_fragment_binarized	100 %
8	10	hot_binarized	100%
9	18	num_shells_binarized	70%
10	30	diff_srv_rate_binarized	100%
11	34	dst_host_same_srv_rate_binarized	100%
12	36	dst_host_same_src_port_rate_binarized	100%

From the data shown in Table 3, it can be noticed that the RF (Random Forest) classifier surpassed the other approaches in terms of accuracy (96.87%), TPR (True Positive Rate) (0.969), and MSE (Mean Squared Error) (0.0458). On the contrary, the SVM (Support Vector Machine) classifier demonstrated the highest MSE (0.1961) and an accuracy of 96.66% among the selected set of classifiers.

Table 3. Performance Comparison of Selected Classifiers Using NSL-KDD Dataset With the Feature Selection Method

Classifier	Accuracy	Recall	ROC	MSE	TPR	FPR
RF	96,87%	0.967	0.988	0.1961	0.967	0.016
SVM	96,66	0.969	0.998	0.0458	0.969	0.013

4.1 Evaluating Our Majority Voting Classifier Method Against Earlier Research on the NSL-KDD Dataset

Table 4 and Figure 3 compare the performance of our proposed technique, which employs the Majority Voting classifier, to other similar efforts on the NSL-KDD dataset. Our methodology outperforms alternative approaches in terms of attack detection accuracy, obtaining a phenomenal rate of 96.49%. Notably, this great degree of accuracy was achieved by employing only 12 of the 41 features available in the dataset. This demonstrates the efficacy and usefulness of our technique, emphasizing its potential to give greater performance in attack detection while leveraging a smaller number of features.

Table 4. Evaluating Our Majority Voting Classifier Method Against Earlier Research on the NSL-KDD Dataset

Approach	Method	Accuracy (%)
[14]	Tree Algorithm	89.24
[15]	SVM	78.89
[16]	RNN+Random Forest+CNN	87.28
Proposed Method	CFS+BestFirst RF + SVM	96.49

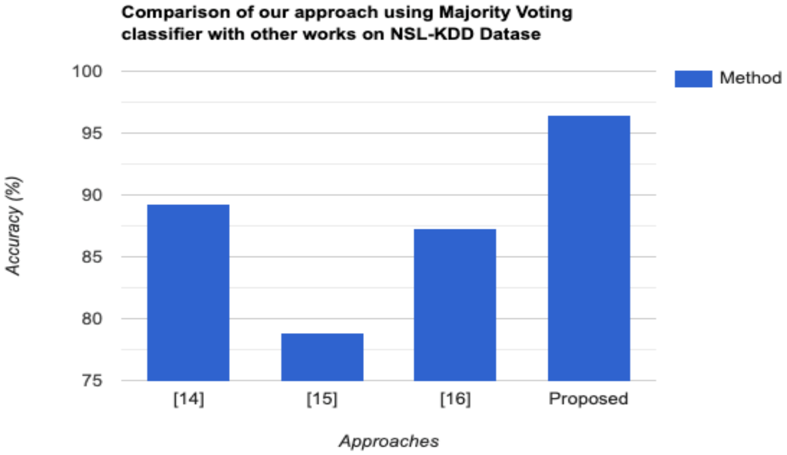

Figure 2. Evaluating Our Majority Voting Classifier Method Against Earlier Research on the NSL-KDD Dataset

5 Conclusion

Using the NSL-KDD dataset, we used feature selection and a majority voting-based classification strategy to detect assaults. We employed the Correlation-based Feature Selection (CFS) technique with the BestFirst search strategy to select features. These technique enabled us in identifying 12 important features. We utilized three machine learning classifiers to test the performance of different feature selection procedures and the selected features: Random Forest (RF), Support Vector Machine (SVM), and Majority Voting. We constructed a multi-model classification model using the Majority Voting classifier. Based on the outcomes of our trials, we proposed a strategy that combines the Majority Voting classifier with feature selection techniques like CFS and BestFirst. This approach successfully detected assaults with an astounding 96.49% accuracy while reducing system overhead. Furthermore, our strategy showed better attack detection accuracy when compared to similar approaches.

References

[1] L. Yang and A. Shami, "IDS-ML: An open source code for Intrusion Detection System development using Machine Learning[Formula presented]," *Softw. Impacts*, vol. 14, 2022, doi: 10.1016/j.simpa.2022.100446.

[2] R. A. Disha and S. Waheed, "Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique," *Cybersecurity*, vol. 5, no. 1, 2022, doi: 10.1186/s42400-021-00103-8.

[3] H. H. Htun, M. B. And, and N. Petkov, *Survey of feature selection and extraction techniques for stock market prediction*, vol. 9, no. 26. Springer Berlin Heidelberg, 2023. doi: 10.1186/s40854-022-00441-7.

[4] P. Dini, A. Elhanashi, A. Begni, S. Saponara, Q. Zheng, and K. Gasmi, "Overview on Intrusion Detection Systems Design Exploiting Machine Learning for Networking Cybersecurity," *Appl. Sci.*, vol. 13, no. 13, 2023, doi: 10.3390/app13137507.

[5] R. Zhao, Y. Mu, L. Zou, and X. Wen, "A Hybrid Intrusion Detection System Based on Feature Selection and Weighted Stacking Classifier," *IEEE Access*, vol. 10, pp. 71414–71426, 2022, doi: 10.1109/ACCESS.2022.3186975.

[6] G. Farahani, "Feature Selection Based on Cross-Correlation for the Intrusion Detection System," *Secur. Commun. Networks*, vol. 2020, 2020, doi: 10.1155/2020/8875404.

[7] P. Krishnamurthy, F. Khorrami, S. Schmidt, and K. Wright, "Machine Learning for NetFlow Anomaly Detection With Human-Readable Annotations," *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 2, pp. 1885–1898, 2021, doi: 10.1109/TNSM.2021.3075656.

[8] M. Samadi Bonab, A. Ghaffari, F. Soleimanian Gharehchopogh, and P. Alemi, "A wrapper-based feature selection for improving performance of intrusion detection systems," *Int. J. Commun. Syst.*, vol. 33, no. 12, 2020, doi: 10.1002/dac.4434.

[9] I. A. Saeed, A. Selamat, M. F. Rohani, O. Krejcar, and J. A. Chaudhry, "A Systematic State-of-the-Art Analysis of Multi-Agent Intrusion Detection," *IEEE Access*, vol. 8, pp. 180184–180209, 2020, doi: 10.1109/ACCESS.2020.3027463.

[10] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. A. A. Khan, "Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review," *Procedia Comput. Sci.*, vol. 171, pp. 1251–1260, 2020, doi:

https://doi.org/10.1016/j.procs.2020.04.133.

[11] E. E. Abdallah, W. Eleisah, and A. F. Otoom, “Intrusion Detection Systems using Supervised Machine Learning Techniques: A survey,” *Procedia Comput. Sci.*, vol. 201, no. C, pp. 205–212, 2022, doi: 10.1016/j.procs.2022.03.029.

[12] S. V. Amanoul, A. M. Abdulazeez, D. Q. Zeebare, and F. Y. H. Ahmed, “Intrusion Detection Systems Based on Machine Learning Algorithms,” *2021 IEEE Int. Conf. Autom. Control Intell. Syst. I2CACIS 2021 - Proc.*, no. August, pp. 282–287, 2021, doi: 10.1109/I2CACIS52118.2021.9495897.

[13] D. Preethi and N. Khare, “An intelligent network intrusion detection system using particle swarm optimization (PSO) and deep network networks (DNN),” *Int. J. Swarm Intell. Res.*, vol. 12, no. 2, pp. 57–73, 2021, doi: 10.4018/IJSIR.2021040104.

[14] A. Ahmim, L. Maglaras, M. A. Ferrag, M. Derdour, and H. Janicke, “A novel hierarchical intrusion detection system based on decision tree and rules-based models,” *Proc. - 15th Annu. Int. Conf. Distrib. Comput. Sens. Syst. DCOSS 2019*, pp. 228–233, 2019, doi: 10.1109/DCOSS.2019.00059.

[15] W. L. Al-Yaseen, “Improving intrusion detection system by developing feature selection model based on firefly algorithm and support vector machine,” *IAENG Int. J. Comput. Sci.*, vol. 46, no. 4, pp. 1–7, 2019, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85077145406&partnerID=40&md5=ccf0f8fb4257444332c608c72ca10255

[16] A. Andalib and V. T. Vakili, “An autonomous intrusion detection system using an ensemble of advanced learners,” *2020 28th Iran. Conf. Electr. Eng. ICEE 2020*, 2020, doi: 10.1109/ICEE50131.2020.9260808.