applied sciences

MDPI

Article

A Novel Feature Selection Approach to Classify Intrusion Attacks in Network Communications

Merve Ozkan-Okay [1,*], Refik Samet [1], Ömer Aslan [2], Selahattin Kosunalp [3], Teodor Iliev [4,*] and Ivaylo Stoyanov [5]

[1] Department of Computer Engineering, Ankara University, Ankara 06830, Turkey; samet@eng.ankara.edu.tr
[2] Department of Software Engineering, Bandırma Onyedi Eylül University, Bandırma, Balıkesir 10200, Turkey; oaslan@bandirma.edu.tr
[3] Department of Computer Technologies, Gönen Vocational School, Bandırma Onyedi Eylül University, Bandırma 10200, Turkey; skosunalp@bandirma.edu.tr
[4] Department of Telecommunication, University of Ruse, 7017 Ruse, Bulgaria
[5] Department of Electrical and Power Engineering, University of Ruse, 7017 Ruse, Bulgaria; stoyanov@uni-ruse.bg
* Correspondence: merveozkan@ankara.edu.tr (M.O.-O.); tiliev@uni-ruse.bg (T.I.)

Abstract: The fast development of communication technologies and computer systems brings several challenges from a security point of view. The increasing number of IoT devices as well as other computing devices make network communications more challenging. The number, sophistication, and severity of network-related attacks are growing rapidly. There are a variety of different attacks including remote-to-user (R2L), user-to-remote (U2R), denial of service (DoS), distributed DDoS, and probing. Firewalls, antivirus scanners, intrusion detection systems (IDSs), and intrusion prevention systems (IPSs) are widely used to prevent and stop cyber-related attacks. Especially, IDPSs are used to stop and prevent intrusions on communication networks. However, traditional IDSs are no longer effective in detecting complicated cyber attacks from normal network traffic. Because of this, new promising techniques, which specifically utilize data mining, machine learning, and deep learning, need to be proposed in order to distinguish intrusions from normal network traffic. To effectively recognize intrusions, the feature generation, feature selection, and learning processes must be performed delicately before the classification stage. In this study, a new feature selection method called FSAP (Feature Selection Approach) is proposed. In addition, a hybrid attack detection model called SABADT (Signature- and Anomaly-Based Attack Detection Technique) is suggested, which utilizes different classification metrics to recognize attacks. The proposed general method FSACM (Feature Selection and Attack Classification Method) is tested on KDD '99, UNSW-NB15, and CIC-IDS2017 datasets. According to the experiment results, the proposed method outperformed the state-of-the-art methods in the literature in terms of detection, accuracy, and false-alarm rates.

Keywords: cyberattacks; intrusion detection system; feature selection; classification; machine learning

Citation: Ozkan-Okay, M.; Samet, R.; Aslan, Ö.; Kosunalp, S.; Iliev, T.; Stoyanov, I. A Novel Feature Selection Approach to Classify Intrusion Attacks in Network Communications. *Appl. Sci.* **2023**, *13*, 11067. https://doi.org/10.3390/app131911067

Academic Editor: Agostino Forestiero

Received: 13 September 2023
Revised: 4 October 2023
Accepted: 6 October 2023
Published: 8 October 2023

1. Introduction

Cybercrime emerged several years ago. During that era, safeguarding the digital realm was simpler due to the limited number of machines in the digital domain, and the assaults were less intricate compared to the present scenario [1]. However, in recent years, regular world transactions have been transferred into the digital environment due to its ease of use and low cost, and due to new technological advancements. Moreover, the progression of technology has empowered cybercriminals to construct automated tools for executing advanced cyber attacks, particularly targeting wireless networks over time [2]. The usage of social media, blockchain, online banking, cloud environment, Internet of Things (IoT) devices, and wireless communication is increasing rapidly as well. Users and companies started to use wireless communication more than normal wire communication

on the computer network. The increased usage of wireless communication technology also raised the amount of research in this domain. Wireless local area networks are the most widely used wireless communication technologies in this field. In general, the use of wireless local area networks has been growing quickly recently and the continuation of this increase is expected in the coming years [3]. On the other hand, the existing condition of wireless local area network technologies renders the utilized network susceptible to a wide range of cyberattacks from passive to active. It is difficult for similar attacks to occur physically because there are situations where the cyberattacker must physically access network cables or traverse several lines, whereas attacks on a wireless local area network can come from any direction and target any node. Examples of these attacks are leaking confidential information, message contamination, and legal impersonation.

With the evolution of novel attack mechanisms, it is now feasible to breach someone's financial system, purloin sensitive data from major corporations, encrypt computer data on hard drives, and impede access to substantial corporate resources through DDoS attacks. The financial cost of these incidents sums up to trillions of US dollars globally each year [4]. Furthermore, the rise of new gadgets like smartphones and IoT devices has expanded the potential points of attack in the wireless arena. Cybercriminals persistently refine existing intrusion tactics by crafting diverse iterations and employing novel attack variations [5] tailored for smartphones and IoT devices. To protect communication networks, and especially wireless networks, from attackers, IDSs (intrusion detection systems) have been used for a long time. However, alongside the rate of development of IDSs, the types and severity of attacks against those networks are increasing at a similar rate as well [6].

An intrusion detection system is software that monitors the computer network's traffic flows for indication of malicious activities including breaking network protocols, censoring, stealing information, and preventing network services [7]. An IDS monitors inbound and outbound traffic [8] from all devices on the network. IDSs work behind a firewall as another filter to catch suspicious packets. In short, they look for two clues: The first is the capture of signatures of known attacks. The second is to detect occurring declinations from ordered activities. An IDS generally depends on pattern correlation to determine attacks. Thanks to this technique, an intrusion detection system can compare packets on the network with a database containing signatures of known attack types. The attacks that an IDS can identify by pattern correlation are commonly known attacks such as worms, ransomware, trojans, viruses, and bots [1]. When an IDS detects an abnormal situation, the system marks the suspicious activity and triggers the alarm. The type of this alert can change from a basic node in an audit log to an emergency alert mail sent to an IT administrator. Finally, the contact or team fixes the problem and tries to identify the main reason for the problem.

The dynamic and complex nature of cyberattacks on computer networks causes existing intrusion detection systems to be insufficient in this regard [9]. This is because new and obfuscated malicious attacks make the detection process challenging and can escape from the network security systems. These reasons raise the need in order to develop new methods in the IDS world. This paper aims to contribute to the solution of wireless local area network security problems by using the latest technological improvements. The paper has three main contributions:

- In order to detect attacks efficiently, the first contribution was made with the feature selection approach (FSAP).
- Afterwards, a hybrid classification technique (SABADT) was presented to detect attacks with high accuracy.
- Finally, an application was made on the KDD '99 datasets in the literature for the performance evaluation of the suggested approaches and techniques.

The KDD '99 as well as UNSW-NB15 training datasets were analyzed while extracting the signatures and rules to train the model. The created model was tested by applying to the KDD '99 and UNSW-NB15 testing datasets. According to the test results, the attacks as well as the attacks' categories were obtained with 99.89% and 98.84% accuracy rates, respectively. Compared to other state-of-the-art studies, there are not many studies proposing new

methods for both feature selection as well as classification stages at the same time. By integrating signature-based and anomaly-based methods, the detection time was reduced while the accuracy rate was increased. In addition, the proposed model was tested on the updated CIC-IDS2017 dataset. Attacks were detected with a 99.51–99.91% accuracy rate. More detailed analysis will be conducted on this data set in future research. In summary, within the scope of this study, intrusion attacks were recognized by examining the traffic behaviors in wireless local area networks, and the attacks' types were determined.

The remaining sections of the paper are structured as follows: Section 2 provides an overview of the rationale for feature selection and explores various approaches that have been proposed in the literature for selecting relevant features in intrusion detection systems. Section 3 details the proposed feature selection and classification methods. Section 4 describes the implementation of the suggested approach. Section 5 presents the results and discussion of the findings. Section 6 discusses the limitations of the study and outlines directions for future research. Finally, Section 7 provides the conclusion of the paper.

2. Related Work

In this section, we have given background information about attack detection and classification processes as well as reviewed needs for feature selection and feature selection approaches, provided a literature review of feature selection methods for intrusion detection systems, and evaluated feature selection methods in the literature.

2.1. Needs for Feature Selection

In the real world, data can be noisy, incomplete, and inconsistent [10] because they are collected from various sources with different techniques. Data preprocessing on collected raw data will increase the quality of the data for further analysis. Data cleaning, reduction, integration, and transformation are commonly used data preprocessing techniques. Dimension reduction, numerosity reduction, and data compression are the methods used during the data reduction process [11,12]. Feature selection and principal component analysis are two commonly used data reduction and dimension reduction techniques. This study aims to make the dataset more suitable for machine learning by examining the data reduction techniques and suggesting new feature selection techniques.

Feature selection is the choosing of the optimal subsets of features that can summarize the data according to the determined criteria. Statistical tests [13], probability tests, and distance calculations, methods, and metrics are widely used in the feature selection process. Examples of these are the Fisher score, Hellinger distance [14], correlation coefficient, Cosine coefficient, Chi-square, and Tversky index [15]. When feature creation processes take place, automatic network monitoring tools are mostly used, which skip the manual work and expert domain knowledge. This causes the collected data to be noisy, incomplete, and inconsistent. Data must be clean, less noisy, and consistent before the learning process takes place [16,17]. Selecting optimal subsets of features in current intrusion detection datasets will increase model performances. Feature selection is needed specifically for the following reasons:

- To remove unrelated and noisy data;
- To make the data more understandable and visible;
- To prevent excessive learning and increase the performance of the model that is used;
- To reduce data cost;
- To reduce the complexity of the model that is used;
- To reduce storage requirements and computational cost.

2.2. Feature Selection Approaches

Relations between feature selection and classification processes in intrusion detection systems can be seen in Figure 1. First, raw data are collected by using relevant packet analyzer tools. Second, the preprocessing stage and feature selection phase take place. Finally, model learning and testing phases take place when the learning process is per-

formed (Figure 1(1.1)). During the learning process, the feature selection process plays an important role. Feature selection methods are divided into three main categories including the filter approach, wrapper approach, and embedded approach. In the filter approach, the more significant features are selected by using a heuristics search [18]. The selection is performed only once before the learning process takes place (Figure 1(1.2)). This approach is fast and scalable, and selected features can be used for various classifiers. However, if the selection is made once and independently of the classifier, it may lead to poor classification performance. On the other hand, in the wrapper approach, the feature selection and classification processes work together [19]. Subsets of properties are generated to represent the features, and classification is performed by using these subfeatures (Figure 1(1.3)). The most ideal subfeature group is selected according to the model performance. Due to the interaction between the feature subset search and model selection, generally higher accuracy and a greater consideration of dependencies between features can be counted among the advantages of the wrapper approach. On the other hand, the disadvantages of these approaches are that the learning process is slower, computationally more expensive, and has the risk of over-learning. In the embedded approach, the feature selection method and the classifier work together [20], but the features are selected during the learning process (Figure 1(1.4)). The embedded approach interacts with classifiers and requires less computation time when compared to the wrapper approach.

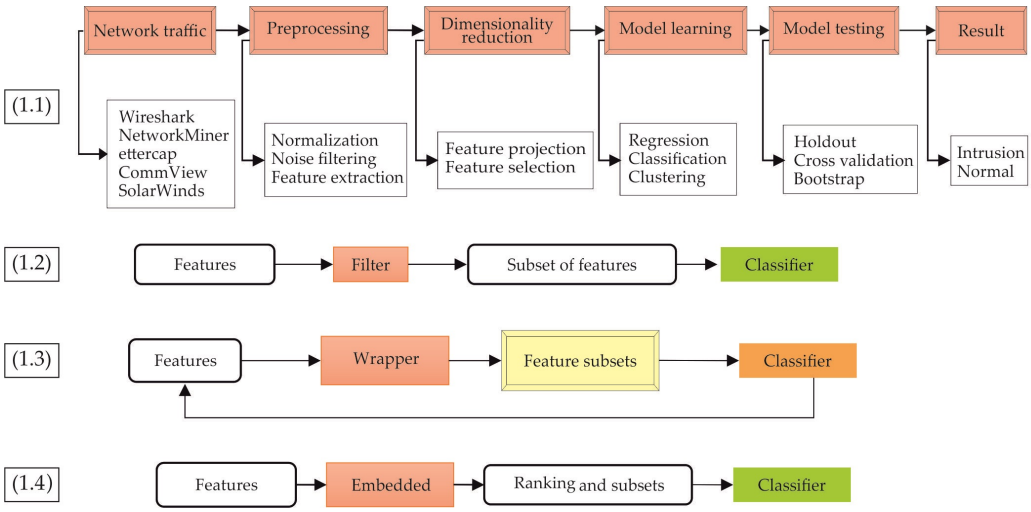

Figure 1. Relations between feature selection and classification processes on intrusion detection system.

2.3. Literature Review of Feature Selection Methods for Intrusion Detection Systems

In the studies in the literature, each detection method used different feature selection techniques for intrusion detection systems. Some studies performed a filter approach to select the best subsets of the features, while others used wrapper or embedded approaches. A few studies utilized both filter and wrapper approaches together to specify the optimum subset of features. In this study, we reviewed the current state-of-the-art methods for intrusion detection systems in the literature for feature selection. Each study is evaluated based on the main idea, used techniques, and performances on applied datasets. The deficiencies of each study are also discussed, and how to improve each method is also given as a suggestion.

Olusola et al. [21] proposed a new feature reduction technique for the KDD '99 dataset. The paper emphasized that selecting more relevant features leads to faster and more appropriate results for network attack detection. First, they used a rough set to remove the redundant properties in the dataset. Then, to obtain the most discriminant properties, the dependency ratio was determined for each class. The information gain was performed to split attributes. The test results confirmed that the performance was increased with fewer features.

Amiri et al. [11] proposed a feature selection method based on a mutual information measure. For feature selection, they used both a linear correlation coefficient and a modified mutual information measure. The modified mutual information selection method was performed with minimum redundancy and maximum appropriateness. Feature goodness and the evaluation function were improved by forward feature selection, linear correlation, and the mutual information measure. They also used a Least Squares Support Vector Machine (LSSVM) for classification. The Least Squares Support Vector Machine was a modification of the standard Support Vector Machine. A linear equation must be unroped in the optimization phase, which is important when avoiding local minima in SVM problems. The experiment was carried out on the KDD cup 99 datasets with high accuracy, specifically for user-to-remote (U2R) and remote-to-login (R2L) attacks. The suggested feature selection method does not include the preprocessing stage and has not been compared with other feature selection methods; thus, these deficiencies have to be avoided for better classification.

Feature selection for wireless IDS, which utilizes filter and wrapper techniques, is proposed in [22]. In the feature selection phase, the information gain ratio measure and K-means algorithm were used. A neural network, which used a backpropagation algorithm, was performed for the learning and testing phases. The suggested system was designed for Wireless LAN environments. According to the paper, the number of features was reduced from 38 to 8. In addition, the accuracy of the system was improved while the learning time was reduced. The proposed approach had not been tested on new datasets or novel attacks. Furthermore, the effectiveness of the suggested system was not compared to leading techniques in the literature. To show the proposed system is suitable for current IDSs, these deficiencies need to be eliminated.

Bostani and Sheikhan [23] suggested a hybrid feature selection method that uses binary gravitational search and mutual information techniques. The proposed method utilized the wrapper feature selection approach, which performed global search. To select the optimal feature sets and decrease the number of features further, the mutual information was integrated with binary gravitational search, which was used as a filter feature selection approach. The proposed hybrid method was tested on the NSL-KDD dataset. According to the paper, the suggested technique increased both the detection as well as accuracy rates while decreasing the false-positive rate as well as feature space drastically. The paper also emphasized that using the wrapper and filter feature selection approaches together allows the best subset of features to be selected. The proposed method was only tested on the NSL-KDD dataset, which is not enough to correctly evaluate the proposed method's performance.

Aminanto et al. [24] proposed a Wi-Fi IDS that utilizes a weighted feature selection technique and neural network algorithm. The feature selection method used an ANN (Artificial Neural Network) and C4.5 to calculate each feature weight. The most significant features were chosen based on the corresponding weights, which represent the importance of each feature. A neural network classifier was performed for classification. The proposed approach was performed on the AWID (Aegean Wi-Fi Intrusion Dataset) dataset. According to the experimental results, a 99.9% detection rate, 99.7% F1 score, and 0.4 false-alarm rate were obtained, which outperformed the state-of-the-art methods for IDSs that use the filter approach. Additionally, the suggested system could handle both known and unknown attacks. The suggested method was performed only on the AWID dataset, and the authors did not mention how they handled unknown attacks or specify the most important features that represent the indication of the intrusion. To make the suggested method more clear, these deficiencies must be eliminated.

Mishra et al. [25] investigated various machine learning techniques for intrusion detection systems. For each attack, associated features were given. Problems with low-frequency attacks were examined and necessary contributions were applied for benchmark datasets. For different attack categories, various machine learning techniques as well as algorithms were employed, and results were compared to identify efficient ML techniques

for attack categories. According to the paper, the performance of standard ML algorithms such as decision trees, C4.5, SVM, neural networks, and fuzzy association rules was fairly high. However, they found that deep learning and reinforcement learning face several difficulties when detecting intrusion in network attacks.

Mohammadi et al. [26] presented a new intrusion detection system. The model presented works by employing a combination of feature selection as well as clustering algorithms, utilizing both wrapper and filter approaches. The proposed model conducted feature grouping by using linear correlation coefficient and cuttlefish algorithms. After relevant features were selected, classification was performed. A decision tree classifier was employed on the KDD Cup dataset for classification. The experimental test results showed that the proposed method produced higher detection and accuracy rates than leading methods in the literature.

Liu et al. [27] presented feature grouping and selection techniques that used deep learning auto-encoder with random forest. The suggested method selected the most significant features from the datasets. An AP (Affinity Propagation) algorithm was performed in order to identify the features that had a strong correlation. To reduce the time complexity and accelerate the learning phase, a traditional unsupervised clustering algorithm and a three-layer shallow auto-encoder neural network were combined together. According to the authors, the presented method outperformed the state-of-the-art methods in terms of detection accuracy, ease of training, and adaptability. These days, normal and attack traffic can be mixed, and the attack traffic can be encrypted, which results in misclassification. Because of that, feature selection alone is not enough to correctly separate intrusions from normal network traffic. Moreover, using only the self-learning three-layer auto-encoder is not guaranteed to assist in increasing the accuracy of unsupervised Affinity Propagation clustering algorithms.

The intrusion detection framework, which utilizes a feature selection process and ensemble learning techniques, was proposed by Zhou et al. [28]. Initially, for dimensionality reduction, the CFS-BA algorithm was performed to choose the best subset by using the correlation between properties. After that, an ensemble method, which combines random forest, C4.5, and Forest by Penalizing Attributes algorithms, was carried out. At last, voting was employed to combine the probability distributions of the base learners for intrusion detection systems. The proposed framework was tested on the NSL-KDD, CIC-IDS2017, and AWID datasets. According to the authors, the test results revealed that the presented CFS-BA feature selection algorithm outperformed state-of-the-art approaches based on various metrics. The proposed framework also needs to be tested on zero-day attacks, which are occasionally seen in communication networks.

Nancy et al. [29] explained intrusion detection systems for wireless sensor networks using dynamic feature selection and fuzzy temporal decision tree classification. The featured selection method identified the ideal subset of features within the dataset. After feasible features were selected, fuzzy temporal decision tree and convolution neural networks were combined to separate intrusions from the normal network traffic. The proposed method was tested on the KDD Cup dataset as well as the network trace dataset. According to the paper, the proposed method effectively detected known and unknown intrusions with less energy consumption. Moreover, it increased the packet delivery ratio while decreasing false-positive and -negative rates. The proposed approach was tested on the KDD Cup dataset, which consists of redundant features and does not consist of new intrusions. Thus, the proposed approach must be tested on new datasets to measure the effectiveness of the suggested method more correctly.

Nazir and Khan [30] suggested TS-RF (Tabu search–random forest) which uses a wrapper-based feature selection method. In the proposed approach, Tabu search was performed as a search strategy whereas random forest was used as a learning algorithm. Tabu search is a kind of local neighborhood search in which each possible solution has an inter-related set of neighbors. On the other hand, random forest is a combination of tree estimators that consists of many trees and classifies using features from which each tree is

sampled independently. To show the effectiveness of the proposed method, the presented technique was tested on the UNSW-NB15 dataset. According to the paper, the proposed feature selection method improved the accuracy rate while decreasing the false-positive rate when compared to other feature selection approaches. The UNSW-NB15 dataset has a class imbalance problem, which increases the misclassification rate while decreasing the detection rate. Before applying the feature selection method, the class imbalance problem needs to be solved. In addition, other new machine learning algorithms such as deep learning can be used instead of random forest. Tabu search can be modified to find more relevant features.

Al-Safi et al. [31] proposed a hybrid approach to select features and detect anomalies in a computer network. They used an information-gain-based algorithm on the NSL-KDD dataset for feature selection. For the classification stage, the Support Vector Machine as well as the Optimization Cuckoo Search algorithm were used. The proposed algorithm was tested on the NSL-KDD dataset. According to the test results, the proposed method generated high accuracy and outperformed other methods that had applied to the NSL-KDD dataset. The proposed method was only tested on the NSL-KDD dataset, which consists of old attack types. Thus, the proposed method needs to be tested on the new intrusion detection datasets. Moreover, the obtained accuracy result, which was 94.21%, is not very high for current IDSs, so feature selection and optimization algorithms must be improved further.

Krishnaveni et al. [32] presented an intrusion detection system that uses ensemble feature selection and classification in the cloud environment. They used a univariate ensemble filter feature selection method to separate the most significant features from the datasets. During the feature selection, five filter feature selection methods were obtain to get the best subsets. To obtain the final subset, rule combinations were used too. After features were selected, voting techniques were used for classification. Classifiers including decision tree, logistic regression, Naive Bayes, and Support Vector Machine were used for classification. The presented method was tested on NSL-KDD, Kyoto, and real-time datasets. According to the test results, model performance was increased compared with other existing methods. The obtained results are statistically significant based on the pairwise t-test. Furthermore, the model achieved a higher accuracy rate while decreasing false alarms. In the paper, the authors did not mention how they handled unknown attacks or how their method was different from existing techniques. Also, deep learning can be used as a classifier to distinguish intrusions from normal network traffic.

Cyber–physical systems (CPSs) constitute intricate multi-layered structures that underpin critical global infrastructure, exerting substantial influence on human lives [33]. With the growing emphasis on connectivity within CPSs, the importance of cyber security has surged. Silvio et al. [33] introduced a metaheuristics-based feature selection model for IDSs in CPSs. They tested how effective feature selection improves the performance of different machine learning approaches. They introduced the utilization of F1-Score within the adapted Greedy Randomized Adaptive Search Procedure (GRASP) metaheuristic to elevate intrusion detection performance across binary, multi-class, and expert classifiers. The findings from numerical analyses unveil that distinct feature subsets align more favorably with specific combinations of IDS strategies, classifier algorithms, and attack classes. Notably, the GRASP metaheuristic identifies features that effectively discern four classes of Denial of Service (DoS) attacks and numerous iterations of injection attacks in the realm of cyber–physical systems.

Prasad et al. [34] proposed a feature selection method grounded in multi-level correlations that was tailored for network intrusion detection data. The suggested technique chooses significant features while reducing the size of the analyzed data. First, they normalized the data. Then, they used correlation-based feature selection in a multi-level manner. Finally, they applied classification. The method's effectiveness was analyzed on the UNSW-NB'15 dataset, for both binary and multi-class cases, showing its prowess. Experimental results demonstrated its superiority over existing techniques.

Albulayhi et al. [35] suggested an innovative methodology for feature selection and extraction in the context of anomaly-based intrusion detection systems (IDSs). The proposed approach utilized two entropy-based techniques, namely information gain (IG) and gain ratio (GR), to select and extract pertinent features at varying ratios. It involved leveraging mathematical set theory, specifically union and intersection operations, to extract the optimal feature set. The method was evaluated on two datasets, the NSL-KDD dataset and the IoT Intrusion Dataset 2020 (IoTID20), which employed four machine learning algorithms: Bagging, IBk, J48, and Multi-Layer Perception. The proposed approach successfully identified 11 and 28 relevant features out of 86 using intersection and union approaches, respectively, on IoTID20. Similarly, the NSL-KDD dataset identified 15 and 25 relevant features out of 41 using the intersection and union strategies. Through a comprehensive comparison with other state-of-the-art investigations, the proposed model demonstrated its superiority and competence, achieving an impressive classification accuracy of 99.98%.

The assessment of intrusion detection systems (IDSs) encompasses various parameters, with a primary focus on the chosen feature selection technique for distinguishing between malicious and legitimate activities [36]. The study was structured to identify an efficient feature selection method that enhances classifier accuracy for intrusion detection systems. First, they introduced a Hybrid Ant–Bee Colony Optimization (HABCO) approach, which transforms the feature selection challenge into an optimization task. Then, the performance of HABCO in conjunction with BHSVM, IDSML, DLIDS, HCRNNIDS, ANNIDS, SVMTHIDS, and GAPSAIDS was evaluated. The findings demonstrated that HABCO outperformed the mentioned methods, yielding notably higher accuracy.

Subramani and Selvi presented an intelligent intrusion detection system tailored for identifying intruders within IoT-based wireless sensor networks [37]. In developing this advanced IDS, the feature selection algorithm rooted in rules and Multi-Objective Particle Swarm Optimization (PSO) was introduced. Additionally, they put forth an intelligent rule-based approach that enhances the performance of Multi-Class Support Vector Machines for more precise intruder identification. An experimental test was performed on the KDD '99 Cup and CIDD datasets. The test results demonstrated that the suggested IDS considerably improves intruder detection accuracy while reducing false-positive rates.

Evaluation of Feature Selection Methods in the Literature

From the literature, different feature selection techniques have been summarized based on the main ideas, proposed methods, and obtained performances for intrusion detection systems. Filter as well as wrapper approaches are used in various studies to select the best subsets of features. In some studies, an embedded approach is performed as well. Even though most studies use the filter or wrapper approaches, the techniques that are used to specify the subset are varied. A summary of each feature selection method in the literature which uses different techniques can be seen in Table 1. Generally, each study tested their approach on only one or two datasets, which cannot be generalized and used for current IDSs, or the proposed methods were not good enough to detect and classify new attack types. During the feature selection process, some irrelevant features are chosen that increase the detection and accuracy rate while decreasing the false-positive and -negative rates (Table 1). However, most of the approaches were tested on old datasets that do not contain unknown attacks and they have not been performed on real network intrusions.

It can be said that selecting the most significant subset of features before performing the learning process increases the model performance when distinguishing attack traffic from the normal one. This is because during the feature generation process, irrelevant, noisy, and unrelated features are being created. Furthermore, certain properties may be seen in intrusions as well as normal network traffic. Moreover, old intrusion detection datasets contain redundant features that decrease the detection and accuracy rates. Eliminating the redundant, irrelevant, or less-significant features enhances the learning process as well as the detection and accuracy rates while decreasing the false-negative rates. Our presented method, which consists of various techniques and algorithms, is designed to overcome

the deficiencies that are mentioned in the literature and makes necessary contributions by selecting the best features for the classification process.

Table 1. Current feature selection methods that are used in the literature for intrusion detection systems.

Paper	Year	Proposed Method	Goal/Success
Olusola et al. [21]	2010	A novel feature selection method for KDD '99 dataset	The performance was increased with less features
Amiri et al. [11]	2011	A technique for selecting features that used the mutual information measure	A higher accuracy specifically for U2R and R2L attacks
Mohanabharathi et al. [22]	2012	A wireless IDS that used filter and wrapper approaches	The number of features was reduced from 38 to 8 with higher accuracy
Bostani and Sheikhan [23]	2017	A feature selection approach that combined binary gravitational search and mutual information techniques in a hybrid manner	The proposed method increased the detection and accuracy rates while decreasing the false-positive rates
Aminanto et al. [24]	2017	Wi-fi IDS that used a weighted feature selection technique and neural network algorithm	The proposed method could handle unknown attacks and outperformed the state-of-the-art methods
Mishra et al. [25]	2018	Evaluated different machine learning techniques for IDSs	The performance was increased when C4.5, SVM, neural network, and fuzzy association rules were used
Mohammadi et al. [26]	2019	A feature selection and clustering algorithm which used wrapper and filter approaches	The proposed method produced higher detection and accuracy rates than leading methods in the literature
Li et al. [27]	2020	A feature grouping and selection technique which used deep learning auto-encoder IDS	The presented method outperformed the state-of-the-art methods in terms of detection accuracy, ease of training, and adaptability
Zhou et al. [28]	2020	A feature selection process and ensemble learning techniques	The presented CFS-BA feature selection algorithm outperformed state-of-the-art approaches based on various metrics
Nancy et al. [29]	2020	A wireless sensor networks used dynamic feature selection and fuzzy temporal decision tree classification	The proposed method effectively detected known and unknown intrusions with less energy consumption
Nazir and Khan [30]	2021	A TS-RF that used a wrapper-based feature selection method	The proposed feature selection method improved the accuracy while decreasing the false-positive rates
Al-Safi et al. [31]	2021	A hybrid approach selected the best subset of features	The proposed method generated high accuracy and outperformed other state-of-the-art methods
Krishnaveni et al. [32]	2021	An ensemble feature selection and classification on the cloud environment	A model performance was increased when it was compared with existing methods
Silvio et al. [33]	2022	Metaheuristics-based feature selection model for IDSs	It identified attacks with fewer features
Prasad et al. [34]	2022	Multi-level correlation-based feature selection method analyzed on the UNSW-NB'15 dataset	Its superiority over existing techniques
Albulayhi et al. [35]	2022	Two entropy-based techniques IG and GR	Achieved 99.98% accuracy with fewer features
Sangaiah et al. [36]	2023	A hybrid heuristics artificial feature selection method	It outperformed existing methods with higher accuracy
Subramani and Selvi [37]	2023	Multi-objective feature selection in IoT networks	The performance was increased significantly

3. Methodology of FSACM

In this study, the FSACM (Feature Selection and Attack Classification Method) is proposed in order to determine attacks dynamically in WLANs quickly and with high accuracy. In the first step of the proposed method, the feature selection process is performed. In this context, an approach called FSAP (Feature Selection Approach) has been proposed for feature selection. In the second stage, a technique called SABADT is proposed for the detection of attacks. The block diagram of this proposed method, consisting of 5 different blocks, is given in Figure 2.

- Choosing and analyzing the dataset.
- Identifying important features and organizing them into groups.
- Detecting anomaly-based and signature-based attacks.
- Using the training and classification for identifying attacks.
- Classification of attacks according to their behavior.

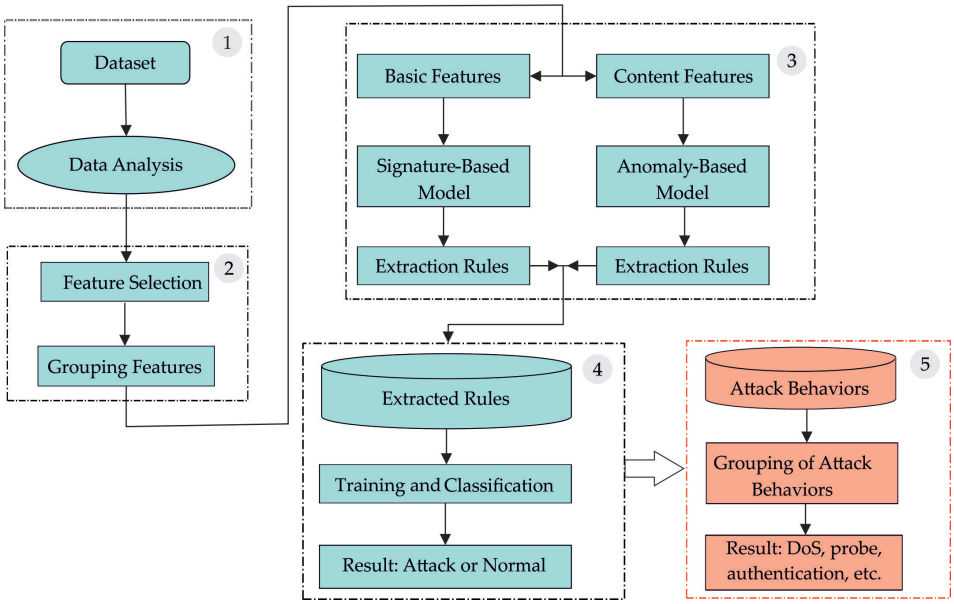

Figure 2. Architecture of the proposed methodology.

The basis of this architecture was proposed in our previous study (Ozkan-Okay et al.). Block 5 has been added within the scope of this study, and the methods in other blocks have been updated according to our previous study. In the architecture given in Figure 2 and block numbers are given on the figure. Datasets and analysis methods from the literature are used in Block 1. In Block 2, a contribution is made within the scope of the study, and the FSAP approach is proposed for feature selection, which is the basis of the model. For detecting attacks in Block 3, a hybrid technique, which is a conjunction of signature-based and anomaly-based techniques, is presented for training and classification stages. Training as well as classification processes are performed in Block 4, using the proposed methodology and known machine learning algorithms from the literature. In Block 5, if an attack is detected in the network traffic, the behaviors of the attack are determined and the behaviors are grouped according to attack types.

3.1. Feature Selection Approach—FSAP

The first version of the proposed FSAP approach was developed in our previous study. The first version of the FSAP approach consists of two stages: With the formula developed in the first stage, some of the features were eliminated. In the second stage, an algorithm was developed to determine which feature is more important by grouping with the remaining features [38].

In this new version of the FSAP approach, a new algorithm has been added that can further reduce the number of features to be used in the learning process. With this algorithm, the distribution of the normal/attack states within the groups formed from the features is examined. That is, this algorithm is based on the homogeneity condition. If there is a homogeneous distribution within a group, this feature is not selective; there must be a heterogeneous distribution. For example, if there is a DoS attack, it should be within a certain value range in the feature. If the feature can correspond to every value it contains, a homogeneous distribution has occurred and this feature cannot be an indicator in due diligence. Based on this situation, an algorithm has been developed. The developed algorithm is given in Algorithm 1.

Algorithm 1: Feature Selection

```
Input  :  g : Feature Groups List
Output :  f : Selected Features List
Definitions
  β : state corresponding to data in the group
  α : state corresponding to data in the group
  c : counter of traffic situation distribution
g_1 ← group1, g_2 ← group2, ....., g_N ← groupB, s ← u(g(j))
for j ← 1 to n do
    for j ← 1 to s − 1 do
        if β(g(j)) == normal then
            cNormal++
        else if β(g(j)) == attackType_1 then
            cType_1++
        else if β(g(j)) == attackType_2 then
            cType_2++
        ...
        else
            cType_k++
    end
    if c_normal/s_i <= α_normal then
        p++
    end
    if c_Type_1/s_i <= α_type_1 then
        p++
    end
    if c_Type_2/s_i <= α_type_2 then
        p++
    end
    ...
    if c_Type_n/s_i <= α_type_n then
        p++
    end
    if p > k/2 then
        add.f()
    else
    eliminatefeature
end
```

By utilizing feature selection algorithms with a predetermined threshold value, we were able to identify significant features. The proper selection of features is crucial for creating a successful machine learning model. Therefore, we utilized the FSAP approach to choose the best subset of features for the given datasets.

3.2. Hybrid Signature- and Anomaly-Based Attack Detection Technique—SABADT

While determining the attack status and type, initially the signature-based model and then the anomaly-based model are applied. The purpose of such a hybrid system is to increase the intrusion detection speed with the signature-based model and to increase the accuracy with the anomaly-based model. As is known, the signature-based model is successful in detecting known attacks quickly and accurately. The anomaly-based model

is slower than the signature-based model, but it has a higher accuracy rate in detecting unknown attack types.

3.2.1. Signature Based Model

The signature-based model, which is the first step of the proposed intrusion detection technique, was developed in our previous study and updated within the scope of this study. This new algorithm can detect attacks as well as attack types [Algorithm 2]. The updated algorithm was created as a result of the arrangement given in Figure 2 (due to the high number of determined rules and parameters, each of them is presented as basic functions rather than algorithmic details).

In this context, the features commonly used within many datasets in the literature are set out and determined as basic features. Parameters for normal situations such as network connection time, the number of network logins, and minimum and maximum values were determined. Based on these parameters, rules were issued. Attacks and their types were determined by comparing possible attack situations with these normal situations. For instance, the maximum/minimum usage times of a network in a normal state and attack state were specified. By comparing these times in both cases, a rule was derived.

Algorithm 2: Signature Based Model

Input : f : Selected Based Features
Output : s : Determined Signatures
Definitions
β : corresponding state in the network
NDP : Normal state parameters
ADP_1 : Attack type1 state parameters
ADP_2 : Attack type2 state parameters
ADP_n : Attack type3 state parameters
r : extracted rules

$f_1 \leftarrow feature1, f_2 \leftarrow feature2, ... f_n \leftarrow featureN, n \leftarrow u(f(i))$
for $i \leftarrow 1$ **to** n **do**
  if $\beta(f(i)) == normal$ **then**
    $write.NDP()$
  else
    if $\beta(f(i)) == type_1$ **then**
      $write.ADP_1()$
    else if $\beta(f(i)) == type_2$ **then**
      $write.ADP_2()$
    else if $\beta(f(i)) == type_3$ **then**
      $write.ADP_3()$
    else
      $write.ADP_n()$
  end
end
$compare(NDP, ADP_1, ADP_2, ..., ADP_n)$
  determine.r()
  write.s(r)
analyze.f()
  determine.r()
  write.s(r)

In general, we recorded both known attack signatures from the literature as well as signatures identified based on common fundamental features found in well-known datasets. Our aim with this model was to enhance accuracy rates through detailed feature analysis and improve the speed of attack detection via rules that lead to direct results.

3.2.2. Anomaly-Based Model

In the second step of the model, traffic behaviors are analyzed. During the analysis process, the traffic flow features of the data sets, content features, and additional features other than these were listed and rules were extracted. With the help of the anomaly-based model, it is possible to detect attacks that are not part of known attacks [39]. During the

creation of the anomaly-based model, each feature group was first analyzed internally, followed by a comprehensive analysis of all features to extract rules for detecting attacks and their types. The algorithm for the anomaly-based model is presented in Algorithm 3, and the calculation details for each parameter and rule have been omitted due to their high number, and have instead been presented as basic functions.

The anomaly-based model generates a large number of complex rules, which require a significant time investment during implementation. However, this model yields a high accuracy rate in attack detection. Furthermore, by decreasing the number of features used, the model is able to achieve improved efficiency in terms of time.

Algorithm 3: Anomaly Based Model

Input : f_c : Selected content features
f_t : Selected time features
f_a : Selected additional features
f : Selected features
Output : s : Determined Signatures
Definitions
β : corresponding state in the network
NDP : Normal state parameters
ADP : Attack state parameters
r : extracted rules

$f_c1 \leftarrow feature1$, $f_c2 \leftarrow feature2$, $f_cn \leftarrow featureN$, $n \leftarrow u(f_c(i))$
$f_t1 \leftarrow feature1$, $f_t2 \leftarrow feature2$, $f_tn \leftarrow featureN$, $n \leftarrow u(f_t(i))$
$f_a1 \leftarrow feature1$, $f_a2 \leftarrow feature2$, $f_an \leftarrow featureN$, $n \leftarrow u(f_a(i))$
$f_1 \leftarrow feature1$, $f_2 \leftarrow feature2$, $f_n \leftarrow featureN$, $n \leftarrow u(f(i))$
for $i \leftarrow 1$ **to** n **do**
  if $\beta(f_c(i)) == normal$ **then**
    $write.NDP()$
  else
    if $\beta(f_c(i)) == type_1$ **then**
      $write.ADP_1()$
    else if $\beta(f_c(i)) == type_2$ **then**
      $write.ADP_2()$
    else if $\beta(f_c(i)) == type_3$ **then**
      $write.ADP_3()$
    else
      $write.ADP_n()$
  end
end
for $i \leftarrow 1$ **to** n **do**
  if $\beta(f_t(i)) == normal$ **then**
    $write.NDP()$
  else
    if $\beta(f_t(i)) == type_1$ **then**
      $write.ADP_1()$
    else if $\beta(f_t(i)) == type_2$ **then**
      $write.ADP_2()$
    else if $\beta(f_t(i)) == type_3$ **then**
      $write.ADP_3()$
    else
      $write.ADP_n()$
  end
end
for $i \leftarrow 1$ **to** n **do**
  if $\beta(f_a(i)) == normal$ **then**
    $write.NDP()$
  else
    if $\beta(f_a(i)) == type_1$ **then**
      $write.ADP_1()$
    else if $\beta(f_a(i)) == type_2$ **then**
      $write.ADP_2()$
    else if $\beta(f_a(i)) == type_3$ **then**
      $write.ADP_3()$
    else
      $write.ADP_n()$
  end
end
$compare(NDP, ADP_1, ADP_2, ..., ADP_n)$
  determine.r()
  write.s(r)
analyze.f()
  determine.r()
  write.s(r)

3.3. Evaluating of Model Performance

The performance and efficiency of the proposed method were evaluated on the KDD '99 and UNSW-NB15 datasets. In addition to the proposed method, machine learning techniques used in previous studies in this field were also applied to these datasets. During the training phase, the KDD '99 and UNSW-NB15 training datasets were used, while the KDD '99 and UNSW-NB15 test datasets were utilized during the testing phase. Machine learning algorithms have been widely applied in various fields, including network attack analysis and detection, and were therefore assessed based on the datasets used in this study. While no algorithm is universally superior, each algorithm has its own advantages and disadvantages, and the optimal algorithm will depend on several factors such as the data distribution, number of features, and inter feature relationships [40–42].

In order to evaluate the efficiency of the proposed model and to compare the performance of various machine learning algorithms, a range of metrics were utilized, including precision, recall, f-measure, accuracy, false positives, and false negatives. These metrics were computed using the confusion matrix. To reduce the overfitting, the datasets were divided into training, validation, and test sets. When performance started to degrade, we stopped the training. Before the classification, the data preprocessing stage was performed efficiently, which degraded the level of overfitting, and the most appropriate features were selected. We also used a large amount of training data to reduce overfitting. Most of the time, using a more diverse dataset can help a model learn the underlying patterns of the data efficiently, which decreases the level of memorizing.

4. Application of Methodology

The implementation of the presented method was coded using Python language. The proposed methodology was tested on the KDD '99 and UNSW-NB15 datasets. The step-by-step implementation of the proposed methodology is explained in detail for KDD '99, which is the most widely used dataset in the literature. Only the steps of the KDD '99 dataset are given in detail, as the processing was performed using the same methods and its application to the UNSW-NB15 dataset is also briefly explained. The results obtained are presented in tables for both datasets. Test procedures on other datasets are also briefly explained in order to avoid repetition. The details of the applications made on the dataset are given in the following subsections.

4.1. KDD '99 Dataset

The rationale for utilizing the KDD '99 dataset in this study is due to its large size and widespread use within the field. The dataset contains 41 features in total, including 9 basic features and 32 derived features. These derived features are divided into three different categories including content features, host-based traffic features, and time-based traffic features.

The KDD '99 dataset is composed of 38 states and 5 groups, including normal, DoS, probe, u2r, and r2l. In the training dataset, 22 states are included, but the test dataset has 16 additional attack types that are not included in the training dataset. The KDD '99 training dataset was used to develop the proposed model for the training stage, and the attack types were labeled as "dos", "probe", "u2r", or "r2l", and normal situations were labeled as "normal" to identify whether the result was an attack or not. The proposed model utilized feature selection methods to choose important features and extracted rules from both signature-based situations, where the result could be directly determined, and anomaly-based behavior of the traffic. The KDD '99 testing dataset was utilized to evaluate the model's effectiveness during the testing phase.

Each feature in the dataset is grouped with the traffic situation result (normal/dos/probe/u2r/r2l) (service-result, src_bytes-result, dst_bytes-result, etc.). To select important features for the result, the number of groups formed by each feature and their impact were compared to a threshold value that was previously calculated. Any feature above this threshold value was considered important and thus selected for use in the final analysis.

In Figure 3b, when the "service" feature is separated according to the values and the attack status, 140 different groups have been formed. There is usually a heterogeneous distribution when looking at the connected service types and attack types. There are two different states for the "IRC" service, and the "normal state" is quite obvious. There are three different cases in the "ecr_i" service type. The type of attack in this service type also increased, but there is still a clear difference between the numerical distributions. As a result of the grouping of the "service" feature, "probe" attacks represented 48, "dos" attacks represented 56, "r2l" attacks represented 7, "u2r" attacks represented 4, and "normal" situations represented 25 different service types. The "service" feature provides the algorithm with four cases in the dataset, but contains five different cases as a result of the developed algorithm. The proposed feature selection II method included specific parameters that were used to identify important features. Based on this method, the "service" feature was found to be significant and remained above the threshold value, thus meeting the criteria for inclusion in the final feature list.

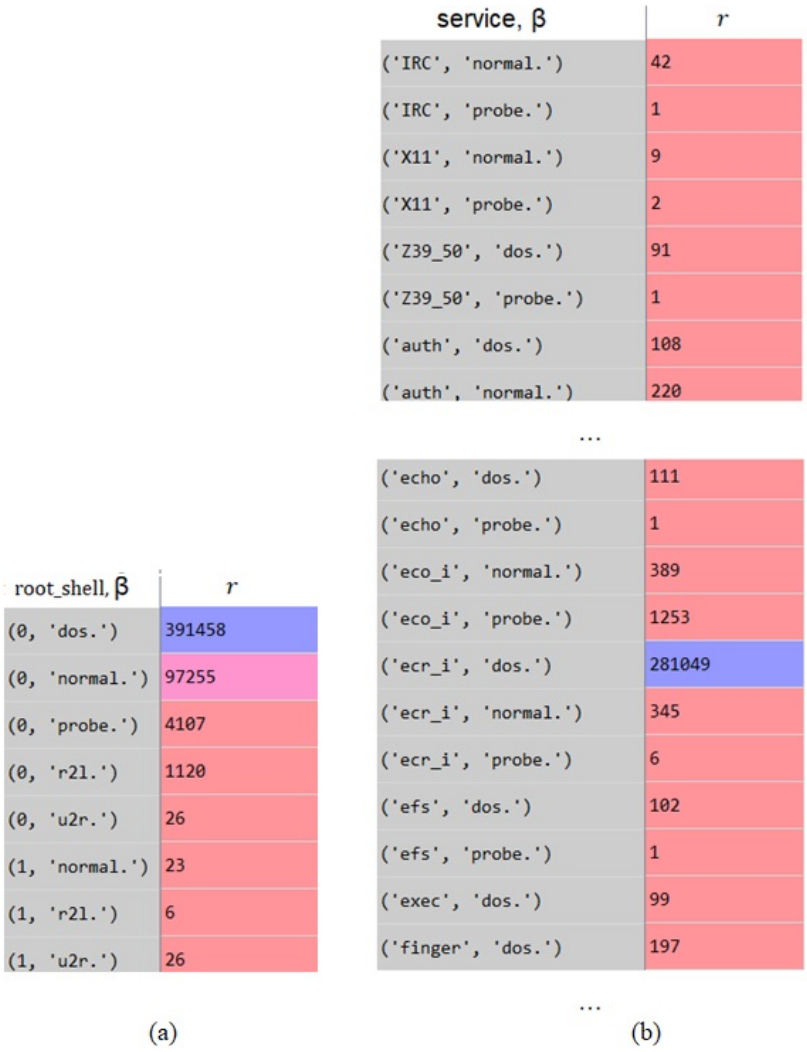

Figure 3. (**a**) Grouping result of "*root_shell*" feature; (**b**) grouping result of "*service*" feature.

In summary, Table 2 displays the selected features that will be used in our model, as determined via the proposed feature selection approach. These features were identified using a combination of our previous study's equation and feature selection Algorithms 1 and 2. With the proposed feature selection algorithms, 10 features were selected among 41 features. In our previous study [38], 22 features were selected from 41 features when intrusions were

distinguished from the normal network traffic flows. With the updated algorithms, we decreased the number of features from 22 to 10 features.

Table 2. Selected features.

duration
service
src_bytes
dst_bytes
count
srv_count
same_srv_rate
dst_host_same_srv_rate
dst_host_rerror_rate
dst_host_srv_rerror_rate

By using the features that were selected from the basic features of the KDD '99 dataset specified in Table 2, rules for the signature-based model that can directly obtain an attack result were extracted. These rules were stored by labeling them as "normal, dos, probe, u2r, r2l". Within this step, 30% of the KDD '99 dataset could be classified for different attack types. This situation allows us to save time and memory. Some of these signature states are illustrated in Table 3. For instance, if the connection time is between the minimum and maximum traffic time, the connection can be detected as normal. If the service is "auth" and the number of connections is larger than the average connection, the connection can be detected as an attack, and the attack can be classified as DoS.

Table 3. Extracted signatures from basic features (abbreviated list).

min("normal_duration")<duration<max("normal_duration")	normal	normal
service="auth", count>avg(normal)	attack	dos
min("normal_src_bytes")<src_bytes<max("normal_src_bytes")	normal	normal
service="efs", same_srv_rate=1	attack	probe
service="ftp_data", dst_bytes>max(other_situations)	attack	r2l
service="telnet",max("normal_duration")<duration<max(other_situations_duration)	attack	u2r
service="ecr_i", count<=1	normal	normal

To develop the anomaly-based model, selected content and host-based and time-based traffic features were utilized to analyze traffic behavior. Rules were extracted using these features, with content features analyzed first, followed by host-based features, and, finally, time-based features. Ultimately, these features were analyzed together to identify patterns and rules in traffic behavior. Table 4 provides some examples of the extracted rules.

Table 4. Extracted rules from content features (abbreviated list).

service=private, duration=0, src_btyes=0, dst_btes=0	attack	dos
service=private, duration>max("normal"), src_btyes=0, dst_btes=0	attack	probe
service=eco_i, srv_diff_host_rate=1	attack	probe
service=http, src_bytes>max("normal_src_bytes")	attack	dos
service=finger, count>= ort("normal_count"), dst_host_srv_serror_rate >=0.1	attack	probe
service=ssh, src_btyes=0, dst_btes=0	attack	dos
service=ftp_data, duration>max("normal_dur"), dst_host_same_srv_rate=1	attack	r2l
service=telnet, count >= avg("normal_count"), srv_count >= avg("normal_srv_count")	attack	u2r
other situations	normal	normal

Two rules are presented as follows to illustrate the approach: (1) When connecting via a private service type, if there is no data flow and the count of faulty connections is non-zero, the connection is considered anomalous. (2) If there is no data flow despite normal traffic conditions, and the connection request to the same host is unusually high, the connection is stored as suspicious. At the final stage, the rules derived from the KDD '99 training dataset, along with selected features, are applied to the KDD '99 test dataset.

4.2. UNSW-NB15 Dataset

The raw network packets of the UNSW-NB15 dataset were created at the Australian Cyber Security Centre's (ACCS) Cyber Range Laboratory using the IXIA PerfectStorm tool to obtain a mixture of real normal activities and artificial attack behaviors. The Tcpdump tool was employed to capture 100 GB of raw traffic (e.g., Pcap files). This dataset includes nine types of attacks, namely Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. It comprises a total of 49 features.

As applied in the KDD'99 dataset, the UNSW-NB15 dataset was also grouped according to attack types and results (Figure 4). Feature selection was performed with the proposed approach.

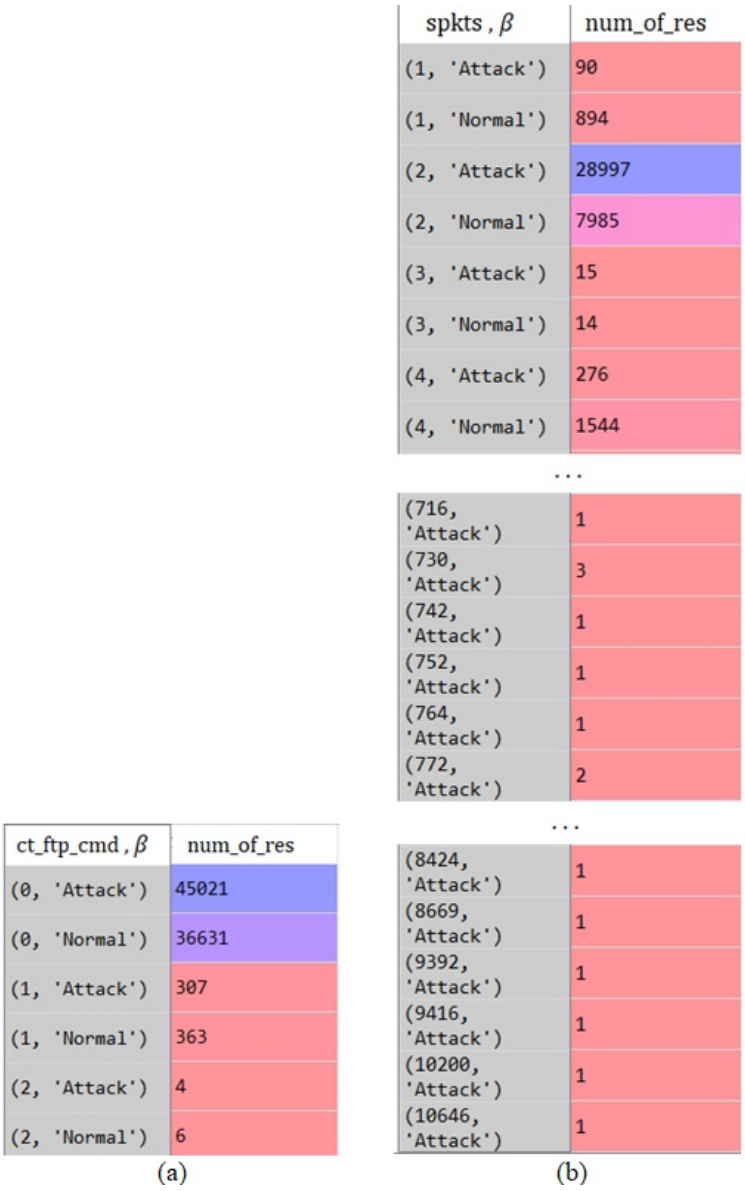

Figure 4. (**a**) Grouping result of "ct_ftp_cmd" feature; (**b**) grouping result of "$spkts$" feature.

In summary, Table 5 displays the selected features that will be used in our model, as determined via the proposed feature selection approach. With the proposed feature selection algorithms, 13 features were selected from among 49 features.

Table 5. Selected features.

service
proto
duration
sttl
spkts
dpkts
sbytes
dbytes
ct_dst_src_ltm
ct_dst_sport_ltm
ct_dst_dport_ltm
sloss
response_body_len

By using the remaining features after the feature selection algorithm from the basic features of the UNSW-NB15 dataset, rules were derived for the signature-based model that could directly obtain an attack result. During the creation of the anomaly-based model, traffic behavior analysis was performed using content features, flow features, time-dependent features, and selected additional features. After that, rules were derived. The list of extracted signatures and rules is given in Table 6.

Table 6. Extracted signatures and rules (abbreviated list).

sttl>=max(normal_dest_time) & response_body_len>0	attack
sttl>=max(normal_dest_time) &ct_dst_src_ltm<min(normal) &sbytes<avg(normal)	attack
sttl>=max(normal_dest_time)& min(normal)<ct_dst_src_ltm<max(normal) & min(normal)<sbytes	normal
ct_dst_src_ltm>max(normal)∣response_body_len >max(normal) ∣ ct_src_dport_ltm>max(normal)	attack
service=”dns”, protocol=”udp”, dur > max(“normal_duration”)	attack
duration=0, spkts>max(normal_spkts)	attack
service=”-”, protocol=”tcp”, dur< max(“normal_duration”)	normal
Dur>min(normal) & response_body_len==0 & tcprtt<min(normal)	attack

5. Results and Discussion

The results of the proposed method with the extracted rules are given in Tables 7–11. The test results are compared with similar methods that were used in the literature, which used machine learning algorithms, based on the performance metrics.

Table 7. Performances of the proposed method on KDD ’99 dataset versus other machine learning algorithms when distinguishing attacks from the normal network traffic.

Technique	FalsePositive	Precision	F-Measure	Accuracy
FSACM	0.011	99.87	99.93	99.89
NaiveBayes	0.028	97.5	90.6	86.13
DecisionTree	0.02	99.8	99.8	99.75
DecisionTable	0.08	98.7	94.65	99.50
SMO	0.056	91.6	61.95	96.83
AdaBoost	0.023	95.25	96.95	97.61

After testing the proposed method, a 99.89% accuracy rate was found when detecting attacks, as shown in Table 7. Although the Decision Table and Decision Tree techniques yielded similar results, their performance was inferior to the proposed method in terms of running time and memory usage. The AdaBoost algorithm could not detect “u2r” and “r2l” attacks. In addition, when the proposed performance values were compared with state-of-the-art studies, the detection rates of the proposed method were higher when detecting

the U2R attack type in the dataset. These attacks are much fewer in the dataset compared to other types of attacks. Therefore, in cases where the distribution is not homogeneous, these algorithms cannot perform effectively. In order to reduce the overfitting, the datasets were split into training, validation, and test sets. Before the classification stage, the data preprocessing phase as well as the most appropriate features were selected to decrease the level of overfitting. In addition, a large amount of training data were used to reduce overfitting. Most of the time, using a bigger dataset helps the model to learn the underlying patterns of the data efficiently, which decreases the level of overfitting.

The proposed method and ML algorithms' detection rates (DRs) are given in Table 8 when classifying types of attacks. After the attacks were separated from the normal network traffic flows, further classification was performed to find out the types of attacks. As can be seen in Table 8, the proposed method outperformed most of the ML classifiers when classifying attack types. For instance, when classifying probe attacks, the proposed method correctly detected 98.67%, while Naive Bayes detected 85.21%, DecisionTree 98.53%, Decision Table 93.10%, SMO 95.12%, and AdaBoost 63.41%, respectively. Similar results were obtained when normal and other attack types including DoS, U2R, and R2L were classified.

Table 8. Performance of proposed method as well as ML algorithms when classifying types of attacks.

Technique	Normal	DoS	Probe	U2R	R2L	
FSACM	99.90	99.91	98.67	82.54	97.0	
NaiveBayes	80.41	88.31	85.21	68.32	34.84	
DecisionTree	99.83	99.89	98.53	43.47	95.41	**Detection Rate (%)**
DecisionTable	99.59	99.83	93.10	47.82	90.32	
SMO	99.46	96.79	95.12	47.82	88.08	
AdaBoost	99.05	98.55	63.41	4.34	0.02	

When the proposed method was tested on the UNSW-NB15 dataset, it was seen that an accuracy rate of 98.84% was achieved for intrusion detection (Table 9). The Decision Table algorithm and Decision Tree algorithm also produced high results, but they were not as successful as the proposed method. In addition, the detection rates of other machine learning techniques in the dataset were quite low compared to the proposed technique in determining the attack types. At the same time, the SMO algorithm could not detect "backdoor", "analysis", or "worms" attack types. The AdaBoost algorithm, on the other hand, was not successful in detecting attack types other than "normal" and "generic". Although these algorithms were successful in identifying the attacks, they made the wrong choice when determining the type of attack. The traces of these attacks in the dataset are very close to each other. Therefore, the tested generic ML algorithms do not work effectively when the traces are not sharp.

The performance of the proposed method and machine learning algorithms in classifying attack types can be seen in Table 10. After the attacks were separated from the normal network traffic flows, further classification was performed to find the attack types. As can be seen from Table 10, the proposed method outperformed machine learning classifiers in classifying attack types. For example, when classifying DoS attacks, the proposed method was 98.66%, Naive Bayes 41.32%, Decision Tree 50.79%, Decision Table 46.34%, SMO 51.43%, and AdaBoost 0.01%, respectively. Similar results were obtained when other attack types such as normal and backdoor and analysis were classified.

Table 9. Performances of the proposed method on UNSW-NB15 dataset versus other machine learning algorithms when distinguishing attacks from the normal network traffic.

Technique	False Positive	Precision	F-Measure	Accuracy
FSACM	0.025	97.27	97.13	98.84
Naive Bayes	0.052	80.07	74.8	71.82
Decision Tree	0.030	85.92	85.81	86.24
Decision Table	0.033	82.11	81.90	82.61
SMO	0.026	85.25	85.95	86.46
AdaBoost	0.280	75.15	69.45	63.13

Table 10. Performance of proposed method as well as ML algorithms when classifying types of attacks.

Tech.	Normal	Backd.	Anal.	Fuzz.	Shell.	Recon.	Exp.	DoS	Worm	Gener.	
FSACM	99.81	98.15	98.23	98.96	99.12	98.41	99.18	98.66	98.15	99.56	
N.Bayes	69.93	16.35	46.34	57.51	60.00	81.73	58.11	41.32	38.46	96.15	
D.Tree	96.81	2.88	4.64	56.93	46.96	80.7	73.38	50.79	69.23	97.94	**DR(%)**
D.Table	95.38	1.92	2.03	47.41	29.57	78.41	63.13	46.34	38.46	95.88	
SMO	99.90	0.01	0.01	58.67	16.53	74.01	69.30	51.43	0.01	96.51	
AdaB.	91.59	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	96.17	

When the proposed method was tested on the CIC-IDS2017 dataset, it was observed that an accuracy rate of 99.52–99.91% was achieved for intrusion detection on each .csv file (Table 11). Other algorithms produced high results but were not as successful as the proposed method. In addition, the success rate varies depending on the type of attack contained in the files and the traces shown by the attacks. More detailed analyses and evaluations will be performed on updated datasets, including the CIC-IDS2017 dataset, in future studies.

Table 11. Performances of the proposed method on CIC-IDS2017 dataset versus other machine learning algorithms when distinguishing attacks from the normal network traffic.

	FSACM	Naive Bayes	Decision Tree	Decision Table	AdaBoost
Tuesday Working Hours	99.87	97.15	99.68	99.18	98.97
Wednesday Working Hours	99.63	98.8	99.49	98.86	93.51
Thursday Working Hours Morning Web Attacks	99.52	94.89	99.14	99.16	98.34
Thursday Working Hours Afternoon Infilteration	99.91	97.93	99.81	99.84	99.19
Friday Working Hours Morning	99.88	96.46	99.83	99.82	99.23
Friday Working Hours Afternoon DDos	99.89	99.78	99.89	99.85	99.57
Friday Working Hours Afternoon PortScan	99.90	99.59	99.87	99.81	99.58

To illustrate the proposed method's performance effectiveness more clearly, the suggested method was also compared with leading methods in the literature. The results of the comparisons are presented in Table 12. When evaluating the methods that were used in the literature, there were not many studies that focused on developing algorithms for both feature selection as well as attack detection. A few studies in the literature focused on developing algorithms that performed both feature selection as well as attack detection. Although Khammassi and Krichen's [43] study achieved high accuracy, our proposed method outperformed the presented papers' performances in terms of the utilized features.

Even though the results of the study by Dhanabal and Shantharajah [44] were evaluated well in terms of the number of features used, the suggested method was outperformed in terms of accuracy. In addition, the previous version of our study results was also better than most of the other studies in terms of accuracy and the number of features that were used. Adding new algorithms as well as updating the existing algorithms in our previous method increased model performances in terms of DR and accuracy while using fewer features in this study.

In addition to developed research with machine learning methods, there are many industrial IDS solutions such as Solarwinds, Bro, Suricata, and Snort. A comparison of our proposed new methodology with these solutions is listed below:

Detection Accuracy: Commercial IDS solutions typically have a large and well-established database of known attack patterns (signatures) and advanced machine learning algorithms to detect anomalies. They often offer high detection accuracy for known attacks. The proposed technique is based on signature and anomaly detection, and may be effective in detecting novel/unknown attacks but might produce false positives.

Ease of Deployment: Commercial IDS solutions often come with user-friendly interfaces and setup wizards, making them relatively easy to deploy. The proposed technique, being a research-based algorithm, may require more technical expertise to implement and configure correctly.

Maintenance and Updates: Commercial IDS solutions are actively maintained by dedicated security teams. They receive regular updates, including new attack signatures and vulnerability patches. The proposed technique's maintenance and update schedule would depend on the researchers behind it.

Cost: Commercial IDS solutions have licensing costs, which can vary widely depending on features and scalability. The proposed technique may be open-source or research-oriented, potentially reducing initial costs.

In summary, the proposed novel technique may have certain advantages in detecting novel attacks due to its anomaly-based approach. Still, it lacks some features such as support, updates, and reliability that commercial IDS solutions provide. Additionally, the proposed technique may be combined with other commercial IDS solutions to improve its overall security posture.

Scalability is important, especially in the context of intrusion detection systems for large-scale networks for real-world applications. The computational complexity of the proposed method's algorithms' running time was in the accepted range, so the performance of the proposed method in terms of DR, accuracy, false positives, etc., was not affected significantly when the size of the analyzed data input grew. In addition, proper preprocessing, feature selection, and dimensionality reduction stages were applied to reduce the computational burden of the proposed method while maintaining its effectiveness. However, the proposed method was tested on limited hardware resources including CPU, memory, and storage. Thus, it is not very scalable for large network traffic. We aim to test it on bigger hardware resources for future work to handle larger-scale networks.

Table 12. Comparing proposed method results with state-of-the-art methods' results in the literature.

Paper	Feature Selection	Classification Technique	Num_of Features	Dataset	Accuracy Rate (%)
Li et al., 2012 [45]	Feature removal method gradually	SVM	19	KDD '99	98.62
Karimi et al., 2013 [46]	Hybrid filtering feature selection	Naive Bayes	16	KDD '99	98.28
Saxena and Richariya 2014 [47]	Standard information gain	Hybrid PSO-SVM Approach	18	KDD '99	99.4
Dhanabal and Shantharajah 2015 [44]	Correlation-based method	J48, SVM, and Naive Bayes	6	KDD '99	98.88, 95.2, 73.32
Moustafa and Slay 2016 [48]	Feature correlation	Several machine learning algorithms	12	UNSW-NB15	85.56
Aghdam and Kabiri 2016 [49]	Ant Colony Optimization-based	-	19	KDD '99	98.9
Hasan et al., 2016 [50]	Higher variable importance score	Random Forest	25	KDD '99	91.9
Khammassi and Krichen 2017 [43]	GA-LR wrapper approach	C4.5, RF, and NBTree	18	KDD '99	99.8, 99.9, 99.85
Janarthanan and Zargari 2017 [51]	ARM algorithm	Random Forest	8	UNSW-NB15	82.99
Manzoor and Kumar 2017 [52]	Information gain and correlation	Based on ANN	25	KDD '99	97.91
Moukhafi et al., 2018 [53]	Particle Swarm Optimization algorithm	Hybrid technique: GA and SVM	16	KDD '99	96.38
Pham et al., 2018 [54]	Ensemble model	J48	35	KDD '99	84.25
Kanimozhi and Jacob 2019 [55]	ARM and CfsSubsetEval	Based on ANN	5	UNSW-NB15	96.00
Chandak et al., 2019 [56]	Ranker- and heuristic-based techniques	C4.5 Decision Tree	27	KDD '99	92.98
Selvakumar and Muneeswaran [57]	Filter- and wrapper-based method with firefly algorithm	C4.5- and Bayesian Network-based	10	KDD '99	90.27
Almasoudy et al., 2020 [58]	Differential Evolution Wrapper Feature Selection	Five and binary classification	9	KDD '99	80.15, 87.53
Kasongo and Sun 2020 [59]	XGBoost-based feature selection	Several machine learning algorithms	19	UNSW-NB15	90.85
Iwendi et al., 2020 [60]	Correlation-based feature selection	Bagging and AdaBoost classifier	13	KDD '99	99.4
Narayasami et al., 2021 [12]	Bat algorithm	SVM	25	KDD '99	94.16
Kocher and Kumar 2021 [61]	Hybrid methods: filter and wrapper	Several machine learning algorithms	23	UNSW-NB15	98.42
Ozkan-Okay et al., 2021 [38]	FSAP	SABADT	17	KDD '99	99.65
Proposed Method FSACM	FSAP new version	SABADT new version	10/11	KDD '99 UNSW-NB15	99.89/98.84

6. Limitation and Future Work

The proposed method performed better than other studies in terms of speed and accuracy rate when separating attacks from normal traffic as well as classifying attack types. However, according to the experimental results, the detection rate is slightly lower in classifying the small amount of attack types or the traces are not sharp in the dataset. The proposed method correctly classified the unseen attacks in the testing dataset; however, those attacks are slightly old. We aim to test our method on new unseen attacks in the future. Moreover, we tested our method on the KDD '99 and UNSW-NB15 datasets for this study. We would like to extend our method and test it on different intrusion detection datasets. In future studies, we also aim to enhance the accuracy measure by improving our proposed algorithms as well as developing new algorithms. To increase the level of scalability for the presented method, we aim to test it on bigger hardware resources in future work to handle larger-scale networks.

Integration into existing systems, real-time monitoring, and response mechanisms are crucial real-world application considerations to evaluate the feasibility of the proposed method. In this context, a new dataset development effort has been undertaken, yielding successful results on a small dataset. However, there are plans for the further enhancement and expansion of this dataset in future studies. This will contribute to better aligning the intrusion detection system with real-world conditions and ensuring more reliable outcomes. These improvements will also facilitate the more efficient implementation of integration processes, real-time monitoring, and rapid response mechanisms. Therefore, our future research endeavors will delve deeper into the practical applicability of IDS and aim to support its successful utilization in real-world scenarios.

7. Conclusions

Cyber-related attacks are evolving and increasing at exponential rates, and no method can recognize these modern attacks. Traditional IDSs are not good enough against modern cyberattacks. To build an effective IDS, feature creation, selection, and learning phases need to be evaluated carefully. All these processes are equally important to effectively fight against intrusions. In this paper, we explained a novel feature selection method which eliminates the redundant, irrelevant, and less-important features from the datasets. In addition, a hybrid (signature- and anomaly-based) classification technique is suggested to detect attacks both faster and with higher accuracy. The proposed method was performed on the KDD '99 as well as UNSW-NB15 datasets. The KDD '99 and UNSW-NB15 training datasets were analyzed while creating the signatures and rules to train the model. The created model was tested on the KDD '99 as well as UNSW-NB15 testing datasets. When the obtained results were compared against the known machine learning methods, better performance results were achieved based on the detection rates varying between 0.2% and 23%. In the suggested work, the test dataset, which contained the attack categories that were not included in the training dataset, was used in the test of the model created with the extracted signatures and rules. After intrusions were distinguished from the normal network traffic, the types of attacks were also identified. At this stage, we classified the types of attack as Dos, Probe, U2R, and U2L or Backdoor, Analysis, Generic, Fuzzers, Worms, etc. Considering the achieved accuracy values, it can be seen that unknown attack categories were also recognized with high accuracy. Furthermore, the proposed methodology outperformed machine learning algorithms in terms of memory and time usage. In addition, the proposed model was tested on the CIC-IDS2017 dataset for detecting attacks. An accuracy rate ranging from 99.52% to 99.91% was achieved. When comparing the proposed study against the state-of-the-art studies in the literature, only a few studies have developed new approaches for both the feature selection as well classification phases. It can be said that the performance of the proposed method is better than other methods in terms of the number of features used and the accuracy rate obtained. With these aspects, it can be said that the proposed method makes significant contributions to the literature.

Author Contributions: Conceptualization, M.O.-O., Ö.A., R.S., S.K., T.I. and I.S.; methodology, M.O.-O. and R.S.; software, M.O.-O. and Ö.A.; validation, S.K., T.I. and I.S.; investigation, M.O.-O., S.K., T.I. and I.S.; writing—original draft preparation, M.O.-O., Ö.A., R.S. and T.I.; writing—review and editing, S.K., T.I. and I.S.; visualization, M.O.-O., R.S. and T.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Aslan, Ö.; Aktuğ, S.S.; Ozkan-Okay, M.; Yilmaz, A.A.; Akin, E. A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. *Electronics* **2023**, *12*, 1333.
2. Yuvaraj, N.; Raja, R.A.; Karthikeyan, T.; Praghash, K. Improved authentication in secured multicast wireless sensor network (MWSN) using opposition frog leaping algorithm to resist man-in-middle attack. *Wirel. Pers. Commun.* **2022**, *123*, 1715–1731.
3. Potteti, S.; Parati, N. Intrusion detection system using hybrid Fuzzy Genetic algorithm. In Proceedings of the 2017 International Conference on Trends in Electronics and Informatics (ICEI), Tirunelveli, India, 11–12 May 2017; pp. 613–618.
4. Williams, C.M.; Chaturvedi, R.; Chakravarthy, K. Cybersecurity risks in a pandemic. *J. Med. Internet Res.* **2020**, *22*, e23692. [CrossRef]
5. Arpaci, I.; Aslan, O. Development of a scale to measure cybercrime-awareness on social media. *J. Comput. Inf. Syst.* **2023**, *63*, 695–705. [CrossRef]
6. Aslan, Ö.A.; Samet, R. A comprehensive review on malware detection approaches. *IEEE Access* **2020**, *8*, 6249–6271. [CrossRef]
7. Ozkan-Okay, M.; Samet, R.; Aslan, Ö.; Gupta, D. A comprehensive systematic literature review on intrusion detection systems. *IEEE Access* **2021**, *9*, 157727–157760.
8. Otair, M.; Ibrahim, O.T.; Abualigah, L.; Altalhi, M.; Sumari, P. An enhanced grey wolf optimizer based particle swarm optimizer for intrusion detection system in wireless sensor networks. *Wirel. Netw.* **2022**, *28*, 721–744.
9. Feng, B.; Zhou, H.; Li, G.; Zhang, Y.; Sood, K.; Yu, S. Enabling machine learning with service function chaining for security enhancement at 5G edges. *IEEE Netw.* **2021**, *35*, 196–201. [CrossRef]
10. Wang, Y.; Ma, J.; Sharma, A.; Singh, P.K.; Gaba, G.S.; Masud, M.; Baz, M. An exhaustive research on the application of intrusion detection technology in computer network security in sensor networks. *J. Sens.* **2021**, *2021*, 5558860. [CrossRef]
11. Amiri, F.; Yousefi, M.R.; Lucas, C.; Shakery, A.; Yazdani, N. Mutual information-based feature selection for intrusion detection systems. *J. Netw. Comput. Appl.* **2011**, *34*, 1184–1199. [CrossRef]
12. Narayanasami, S.; Sengan, S.; Khurram, S.; Arslan, F.; Murugaiyan, S.K.; Rajan, R.; Peroumal, V.; Dubey, A.K.; Srinivasan, S.; Sharma, D.K. Biological feature selection and classification techniques for intrusion detection on BAT. *Wirel. Pers. Commun.* **2022**, *127*, 1763–1785. [CrossRef]
13. Thakkar, A.; Lohiya, R. Fusion of statistical importance for feature selection in Deep Neural Network-based Intrusion Detection System. *Inf. Fusion* **2023**, *90*, 353–363. [CrossRef]
14. Fu, G.; Li, B.; Yang, Y.; Wei, Q. A Multi-Distance Ensemble and Feature Clustering Based Feature Selection Approach for Network Intrusion Detection. In Proceedings of the 2022 International Symposium on Sensing and Instrumentation in 5G and IoT Era (ISSI), Shanghai, China, 17–18 November 2022; pp. 160–164.
15. El-Rashidy, M.A.; Mohamed, R.G.; El-Fishawy, N.A.; Shouman, M.A. An effective text plagiarism detection system based on feature selection and SVM techniques. *Multimed. Tools Appl.* **2023**, 1–38. [CrossRef]
16. Singh, H. Performance analysis of unsupervised machine learning techniques for network traffic classification. In Proceedings of the 2015 Fifth International Conference on Advanced Computing & Communication Technologies, Haryana, India, 21–22 February 2015; pp. 401–404.
17. Cai, J.; Luo, J.; Wang, S.; Yang, S. Feature selection in machine learning: A new perspective. *Neurocomputing* **2018**, *300*, 70–79.
18. Lyu, Y.; Feng, Y.; Sakurai, K. A survey on feature selection techniques based on filtering methods for cyber attack detection. *Information* **2023**, *14*, 191. [CrossRef]
19. Maldonado, J.; Riff, M.C.; Neveu, B. A review of recent approaches on wrapper feature selection for intrusion detection. *Expert Syst. Appl.* **2022**, *198*, 116822.
20. Solorio-Fernández, S.; Carrasco-Ochoa, J.A.; Martínez-Trinidad, J.F. A review of unsupervised feature selection methods. *Artif. Intell. Rev.* **2020**, *53*, 907–948.
21. Olusola, A.A.; Oladele, A.S.; Abosede, D.O. Analysis of KDD '99 intrusion detection dataset for selection of relevance features. In Proceedings of the World Congress on Engineering and Computer Science, WCECS, San Francisco, CA, USA, 20–22 October 2010; Volume 1, pp. 20–22.

22. Mohanabharathi, R.; Kalaikumaran, M.T.; Karthi, S. Feature selection for wireless intrusion detection system using filter and wrapper model. *Int. J. Mod. Eng. Res. (IJMER)* **2012**, *2*, 1552–1556.
23. Bostani, H.; Sheikhan, M. Hybrid of binary gravitational search algorithm and mutual information for feature selection in intrusion detection systems. *Soft Comput.* **2017**, *21*, 2307–2324. [CrossRef]
24. Aminanto, M.E.; Tanuwidjaja, H.C.; Yoo, P.D.; Kim, K. Wi-Fi intrusion detection using weighted-feature selection for neural networks classifier. In Proceedings of the 2017 International Workshop on Big Data and Information Security (IWBIS), Jakarta, Indonesia, 23–24 September 2017; pp. 99–104.
25. Mishra, P.; Varadharajan, V.; Tupakula, U.; Pilli, E.S. A detailed investigation and analysis of using machine learning techniques for intrusion detection. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 686–728. [CrossRef]
26. Mohammadi, S.; Mirvaziri, H.; Ghazizadeh-Ahsaee, M.; Karimipour, H. Cyber intrusion detection by combined feature selection algorithm. *J. Inf. Secur. Appl.* **2019**, *44*, 80–88. [CrossRef]
27. Li, X.; Chen, W.; Zhang, Q.; Wu, L. Building auto-encoder intrusion detection system based on random forest feature selection. *Comput. Secur.* **2020**, *95*, 101851. [CrossRef]
28. Zhou, Y.; Cheng, G.; Jiang, S.; Dai, M. Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Comput. Netw.* **2020**, *174*, 107247. [CrossRef]
29. Nancy, P.; Muthurajkumar, S.; Ganapathy, S.; Santhosh Kumar, S.; Selvi, M.; Arputharaj, K. Intrusion detection using dynamic feature selection and fuzzy temporal decision tree classification for wireless sensor networks. *IET Commun.* **2020**, *14*, 888–895. [CrossRef]
30. Nazir, A.; Khan, R.A. A novel combinatorial optimization based feature selection method for network intrusion detection. *Comput. Secur.* **2021**, *102*, 102164. [CrossRef]
31. Al-Safi, A.H.S.; Hani, Z.I.R.; Zahra, M.A. Using a hybrid algorithm and feature selection for network anomaly intrusion detection. *J. Mech. Eng. Res. Dev.* **2021**, *44*, 253–262.
32. Krishnaveni, S.; Sivamohan, S.; Sridhar, S.; Prabakaran, S. Efficient feature selection and classification through ensemble method for network intrusion detection on cloud computing. *Clust. Comput.* **2021**, *24*, 1761–1779. [CrossRef]
33. Quincozes, S.E.; Passos, D.; Albuquerque, C.; Mossé, D.; Ochi, L.S. An extended assessment of metaheuristics-based feature selection for intrusion detection in CPS perception layer. *Ann. Telecommun.* **2022**, *77*, 457–471. [CrossRef]
34. Prasad, M.; Gupta, R.K.; Tripathi, S. A multi-level correlation-based feature selection for intrusion detection. *Arab. J. Sci. Eng.* **2022**, *47*, 10719–10729. [CrossRef]
35. Albulayhi, K.; Abu Al-Haija, Q.; Alsuhibany, S.A.; Jillepalli, A.A.; Ashrafuzzaman, M.; Sheldon, F.T. IoT intrusion detection using machine learning with a novel high performing feature selection method. *Appl. Sci.* **2022**, *12*, 5015. [CrossRef]
36. Sangaiah, A.K.; Javadpour, A.; Ja'fari, F.; Pinto, P.; Zhang, W.; Balasubramanian, S. A hybrid heuristics artificial intelligence feature selection for intrusion detection classifiers in cloud of things. *Clust. Comput.* **2023**, *26*, 599–612. [CrossRef]
37. Subramani, S.; Selvi, M. Multi-objective PSO based feature selection for intrusion detection in IoT based wireless sensor networks. *Optik* **2023**, *273*, 170419. [CrossRef]
38. Ozkan-Okay, M.; Aslan, Ö.; Eryigit, R.; Samet, R. SABADT: Hybrid intrusion detection approach for cyber attacks identification in WLAN. *IEEE Access* **2021**, *9*, 157639–157653. [CrossRef]
39. Yu, S.J.; Koh, P.; Kwon, H.; Kim, D.S.; Kim, H.K. Hurst parameter based anomaly detection for intrusion detection system. In Proceedings of the 2016 IEEE International Conference on Computer and Information Technology (CIT), Nadi, Fiji, 8–10 December 2016; pp. 234–240.
40. Belavagi, M.C.; Muniyal, B. Performance evaluation of supervised machine learning algorithms for intrusion detection. *Procedia Comput. Sci.* **2016**, *89*, 117–123. [CrossRef]
41. Saranya, T.; Sridevi, S.; Deisy, C.; Chung, T.D.; Khan, M.A. Performance analysis of machine learning algorithms in intrusion detection system: A review. *Procedia Comput. Sci.* **2020**, *171*, 1251–1260. [CrossRef]
42. Almseidin, M.; Alzubi, M.; Kovacs, S.; Alkasassbeh, M. Evaluation of machine learning algorithms for intrusion detection system. In Proceedings of the 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, 14–16 September 2017; pp. 000277–000282.
43. Khammassi, C.; Krichen, S. A GA-LR wrapper approach for feature selection in network intrusion detection. *Comput. Secur.* **2017**, *70*, 255–277. [CrossRef]
44. Dhanabal, L.; Shantharajah, S. A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. *Int. J. Adv. Res. Comput. Commun. Eng.* **2015**, *4*, 446–452.
45. Li, Y.; Xia, J.; Zhang, S.; Yan, J.; Ai, X.; Dai, K. An efficient intrusion detection system based on support vector machines and gradually feature removal method. *Expert Syst. Appl.* **2012**, *39*, 424–430. [CrossRef]
46. Karimi, Z.; Kashani, M.M.R.; Harounabadi, A. Feature ranking in intrusion detection dataset using combination of filtering methods. *Int. J. Comput. Appl.* **2013**, *78*, 21–27 [CrossRef]
47. Saxena, H.; Richariya, V. Intrusion detection in KDD '99 dataset using SVM-PSO and feature reduction with information gain. *Int. J. Comput. Appl.* **2014**, *98*, 25–29.
48. Moustafa, N.; Slay, J. The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD '99 data set. *Inf. Secur. J. Glob. Perspect.* **2016**, *25*, 18–31. [CrossRef]

49. Aghdam, M.H.; Kabiri, P. Feature selection for intrusion detection system using ant colony optimization. *Int. J. Netw. Secur.* **2016**, *18*, 420–432.
50. Hasan, M.A.M.; Nasser, M.; Ahmad, S.; Molla, K.I. Feature selection for intrusion detection using random forest. *J. Inf. Secur.* **2016**, *7*, 129–140. [CrossRef]
51. Janarthanan, T.; Zargari, S. Feature selection in UNSW-NB15 and KDDCUP'99 datasets. In Proceedings of the 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), Edinburgh, UK, 19–21 June 2017; pp. 1881–1886.
52. Manzoor, I.; Kumar, N. A feature reduced intrusion detection system using ANN classifier. *Expert Syst. Appl.* **2017**, *88*, 249–257.
53. Moukhafi, M.; El Yassini, K.; Bri, S. A novel hybrid GA and SVM with PSO feature selection for intrusion detection system. *Int. J. Adv. Sci. Res. Eng.* **2018**, *4*, 129–134. [CrossRef]
54. Pham, N.T.; Foo, E.; Suriadi, S.; Jeffrey, H.; Lahza, H.F.M. Improving performance of intrusion detection system using ensemble methods and feature selection. In Proceedings of the Australasian Computer Science Week Multiconference, Brisband, QLD, Australia, 29 January–2 February 2018; pp. 1–6.
55. Kanimozhi, V.; Jacob, P. UNSW-NB15 dataset feature selection and network intrusion detection using deep learning. *Int. J. Recent Technol. Eng.* **2019**, *7*, 443–446.
56. Chandak, T.; Ghorpade, C.; Shukla, S. Effective analysis of feature selection algorithms for network based intrusion detection system. In Proceedings of the 2019 IEEE Bombay Section Signature Conference (IBSSC), Mumbai, India, 26–28 July 2019; pp. 1–5.
57. Selvakumar, B.; Muneeswaran, K. Firefly algorithm based feature selection for network intrusion detection. *Comput. Secur.* **2019**, *81*, 148–155.
58. Almasoudy, F.H.; Al-Yaseen, W.L.; Idrees, A.K. Differential evolution wrapper feature selection for intrusion detection system. *Procedia Comput. Sci.* **2020**, *167*, 1230–1239. [CrossRef]
59. Kasongo, S.M.; Sun, Y. Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. *J. Big Data* **2020**, *7*, 1–20. [CrossRef]
60. Iwendi, C.; Khan, S.; Anajemba, J.H.; Mittal, M.; Alenezi, M.; Alazab, M. The use of ensemble models for multiple class and binary class classification for improving intrusion detection systems. *Sensors* **2020**, *20*, 2559. [CrossRef]
61. Kocher, G.; Kumar, G. Analysis of Machine Learning Algorithms with Feature Selection for Intrusion Detection Using UNSW-NB15 Dataset. 2021. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3784406 (accessed on 8 May 2023).