Journal of Computer Science

Research Article

A Misuse and Anomaly Intrusion Detection System Using Hybrid Supervised and Unsupervised Data Mining Approaches

Homa Molavi[1] and Mohammad Khanbabaei[2]

[1]*University of Greenwich, School of Business, Operations & Strategy, London, United Kingdom*
[2]*Department of Information Technology Management, Science and Research Branch, Islamic Azad University, Tehran, Iran*

Article history
Received: 11-07-2025
Revised: 28-12-2025
Accepted: 20-05-2026

Corresponding Author:
Mohammad Khanbabaei
Department of Information Technology Management, Islamic Azad University, Science and Research Branch, Iran
Email: mohammadkhanbabaei@srbiau.ac.ir

Abstract: Intrusion Detection Systems (IDSs) play a crucial role in monitoring computer network security. Data mining and machine learning techniques facilitate the identification of intrusion patterns within large volumes of data. Hybrid data mining models have become increasingly popular in IDSs due to their enhanced effectiveness. This study presents a novel hybrid IDS that combines misuse and anomaly detection by integrating supervised, unsupervised, and outlier detection methods from data mining, implemented in three phases. First, data pre-processing techniques are applied to prepare the dataset. Second, the K-means clustering algorithm is used for cluster profiling. Next, association rule mining and outlier detection techniques characterize normal and attack patterns. Third, various classical and ensemble learning algorithms are employed to classify the patterns in the dataset. Evaluating the proposed model using the NSL-KDD dataset demonstrates its superior performance compared to previous studies. The model employs the association rule mining algorithm to generate valuable if-then patterns for both misuse and anomaly detection. Additionally, it utilizes classic and ensemble supervised machine learning methods to classify attack and normal records within the IDS. Ultimately, the proposed model uncovers and characterizes hidden intrusion patterns, thereby enhancing the overall effectiveness of IDSs.

Keywords: Data Mining, Intrusion Detection, Misuse Detection, Anomaly Detection, Outlier Detection

Introduction

In the current era, businesses face significant pressure to allocate substantial financial resources to security measures to ensure the continuity of their operations and maintain their market reputation (More et al., 2024; Hossain and Islam, 2023; Ahmad et al., 2021). Cyber-attacks targeting an organization's network can impose considerable financial burdens on the computational infrastructure of these enterprises (Ahmad et al., 2021). Consequently, the development of effective Intrusion Detection Systems (IDSs) is imperative.

In the field of intrusion detection, there are three primary classifications of systems: first, signature-based intrusion detection systems, also known as misuse detection; second, anomaly detection systems; and third, hybrid detection systems that combine distinctive features from both approaches. Misuse detection IDSs attempt to identify harmful patterns previously observed in network traffic; however, they cannot detect unknown intrusions. In contrast, anomaly detection IDSs use learning algorithms to predict and identify new, previously unseen intrusion behaviors. These systems may exhibit a high false alarm rate because they classify new intrusions based on patterns recognized in prior network transactions, while new behavioral patterns may have emerged subsequently (Aldweesh et al., 2020; Ahmad et al., 2021; Kasongo and Sun, 2020). Contemporary IDSs face two significant challenges: Reduced accuracy and an increased incidence of false alarms. These issues compel network experts to devote additional effort to managing alerts triggered by irregular network traffic (Aldallal, 2022). Consequently, researchers have sought to overcome these challenges by employing misuse and

anomaly detection systems and developing hybrid IDS models, particularly through data mining and machine learning techniques.

For example, to develop an IDS, researchers employ data mining and machine learning techniques, including Case-Based Reasoning (CBR), Support Vector Machines (SVM), Artificial Neural Networks (ANN), and feature selection algorithms. Both supervised and unsupervised methods are utilized, such as combining K-means clustering, naïve Bayes feature selection, and C4.5 decision tree classification (Yang et al., 2022; Khraisat et al., 2019).

In recent years, hybrid misuse and anomaly IDSs have been developed to address the limitations of the two traditional IDS types (Bhati and Khari, 2021; Guezzaz et al., 2022; Hajisalem and Babaie, 2018; Kaur and Singh, 2020; Meryem and Ouahidi, 2020; Talukder et al., 2023). For example, researchers have proposed novel hybrid IDSs by combining various data mining techniques, such as SVM, Kernel Principal Component Analysis (KPCA), Genetic Algorithms (GA), naïve Bayes, random forest, C4.5 decision trees, K-Nearest Neighbor (KNN), stacking ensemble classifiers, and feature selection (Cavusoglu, 2019). Other approaches include compound models incorporating feature scaling through normalization, feature extraction via Principal Component Analysis (PCA), and six classifiers to detect Internet of Things (IoT) network attacks (Kayode et al., 2022); the use of Synthetic Minority Over-Sampling Technique (SMOTE) for data balancing combined with extreme gradient boosting (XGBoost) for feature selection, alongside random forest, decision tree, KNN, and neural networks for classification (Talukder et al., 2023); integration of optimization methods such as Particle Swarm Optimization (PSO) with machine learning algorithms including KNN, decision trees, and neural networks (Sivagaminathan et al., 2023); and hybrid models combining K-means clustering, SMOTE balancing, PCA, decision trees, random forest, and XGBoost (Talukder et al., 2025).

Maseer et al. (2021) developed a hybrid machine learning approach that combines supervised methods such as neural networks, decision trees, KNN, naïve Bayes, random forests, SVM, and convolutional neural networks with unsupervised algorithms, including Expectation-Maximization (EM), K-means clustering, and Self-Organizing Maps (SOM), to detect anomalies in IDS.

Similar to hybrid data mining approaches, ensemble learning classifiers such as random forest, adaptive boosting (AdaBoost), and bootstrap aggregation (Bagging) (Kaja et al., 2019; Yang et al., 2022; Zhou et al., 2020), as well as stacking ensemble learning (Cavusoglu, 2019), decision trees, KNN, and deep neural networks (Gao et al., 2019), have been applied in IDS. Ensemble learning improves the generalization and accuracy of the final model by combining multiple classifiers. These algorithms integrate several weak classifiers to enhance the performance of the final classifier (Hossain and Islam, 2023; Ahmad et al., 2021).

The aim of this research is to improve the performance metrics of IDS by proposing a model that integrates feature selection, clustering, association rule mining, outlier detection, and classification techniques within the data mining process. This approach addresses existing gaps in IDS accuracy and false alarm rates. By developing a hybrid IDS that combines anomaly detection and misuse detection through supervised, unsupervised, and outlier detection methods, this study presents a novel and comprehensive solution.

To Elaborate, the Proposed Model is Based on the Following Approaches.

Misuse detection using unsupervised learning starts with cluster analysis to separate normal and attack records. Subsequently, association rule mining techniques are applied to classify normal and attack behaviors within the respective clusters. Notably, the hybrid approach combining clustering and association rules for unsupervised misuse detection constitutes an original contribution with the potential to enhance both classification accuracy and detection rates.

Anomaly detection using unsupervised learning employs clustering followed by outlier detection to identify both attack and normal anomaly patterns within the dataset. The dataset consists of normal and attack records, as well as anomaly and non-anomaly records. In this study, anomaly records refer specifically to outlier records. Generally, records in a dataset can be categorized as either normal or attack outliers. An IDS may mistakenly classify an outlier record as an attack when it is actually normal, or vice versa. Such misclassifications can result in a high false alarm rate. To characterize outlier behaviors within each cluster, an association rule mining algorithm is applied. This algorithm detects attack records within normal clusters as outliers (anomalies) and can also identify normal records within attack clusters as outliers (anomalies). Overall, this approach integrates anomaly detection with unsupervised learning. Notably, previous research has not utilized a hybrid combination of clustering, outlier detection, and association rules for anomaly detection in IDS.

Misuse detection using supervised learning employs classification models to distinguish between attack and normal records within each cluster. The results are then combined to create a final classifier. This method can enhance performance metrics, including classification accuracy, detection rate, precision, F-measure, and false alarming rate.

Anomaly detection using supervised learning: Following cluster analysis, this approach employs various classification models to distinguish between normal outliers and attack outliers within their respective clusters. It integrates outlier detection with supervised learning to improve the performance of the IDS model. Notably, this represents the first effort to apply supervised learning

techniques to enhance the effectiveness of anomaly detection models.

The proposed model comprises three phases: data preparation and preprocessing (to identify enhanced behavioral patterns in the intrusion detection dataset), misuse and anomaly detection using unsupervised learning, and misuse and anomaly detection using supervised learning.

Related Work

To conduct a systematic literature review of IDS, researchers have extensively examined data mining and machine learning techniques (Ali et al., 2022; Salo et al., 2018), as well as various technologies and approaches (Abdulganiyu et al., 2023; Nasir et al., 2022; Ozkan-Okay et al., 2021). They have also analyzed performance metrics (Yang et al., 2022), datasets, and methodologies (Amarudin et al., 2020; Yang et al., 2022) employed in IDS. Additionally, Khraisat et al. (2019) proposed a taxonomy for IDS based on their characteristics, categorizing them as statistics-based, pattern-based, rule-based, state-based, and heuristic-based systems.

The following section reviews relevant research on the application of data mining and machine learning techniques in IDS.

Geetha et al. (2018) proposed a rule-based decision tree approach to detect intrusive activities in wireless sensor networks. They utilized several classifiers, including alternating decision tree, decision stump, J48, logical model tree, naïve Bayes tree, and fast decision tree learner, to classify intrusions. The NSL-KDD dataset was employed to evaluate the performance of these classifiers. Additionally, feature selection algorithms were applied to identify the most relevant features for distinguishing between normal and attack activities. The results demonstrated that the proposed model outperformed the other classifiers.

Kaja et al. (2019) applied K-means clustering to the dataset, followed by several machine learning algorithms to classify attacks. In this study, various preprocessing steps were implemented to enhance the model's ability to distinguish between normal and attack records. These steps included removing variables with low variance and eliminating correlated features using a correlation-based feature selection technique. The results demonstrated that the proposed algorithm achieved an accuracy of 99.95%.

Cavusoglu (2019) developed a hybrid model that combines two feature selection methods filter and wrapper with various machine learning algorithms for IDS. The NSL-KDD dataset was used to evaluate the proposed model. The study applied several machine learning algorithms, including naïve Bayes, random forest, C4.5 decision tree, KNN, and stacking ensemble methods. The results demonstrated high accuracy and a low false alarm rate in classifying normal and attack records within the IDS.

Kasongo and Sun (2020) employed an XGBoost-based feature selection algorithm alongside several machine learning classifiers, including SVM, KNN, logistic regression, neural networks, and decision trees, for IDS. They used binary classification to distinguish between normal and attack records. The feature selection method effectively identified the importance of each feature. The results demonstrated that applying this feature selection algorithm increased the classification accuracy of the decision tree classifier to 90.85%. However, the other classifiers did not achieve comparable accuracy or performance metrics.

Kilincer et al. (2021) developed a procedure to classify records in cybersecurity IDS. This study compared results across several datasets, including NSL-KDD, and evaluated classification performance using various parameters for multiple machine learning algorithms. The algorithms applied in this study included SVM, KNN, decision trees, and neural networks. The results indicated that the decision tree classifier outperformed the others, achieving an accuracy of 99.92% on the NSL-KDD dataset.

Naseri and Gharehchopogh (2022) proposed a hybrid machine learning approach for intrusion detection that incorporates various feature selection methods. The study employed several machine learning algorithms, including KNN, SVM, decision tree, naïve Bayes, random forest, and AdaBoost. The NLS-KDD dataset was used to evaluate the proposed model. The results demonstrated that the hybrid model generally outperformed the individual classifiers in terms of accuracy, recall, precision, and F-measure.

Bakro et al. (2023) employed filter-based and automated feature selection methods in conjunction with ensemble classifiers for IDS within the cloud computing domain. Their model integrated SVM and XGBoost, while other deep learning algorithms were enhanced using various feature selection techniques. Additionally, multiple preprocessing steps were implemented to improve the performance of the machine learning methods utilized. The proposed model outperformed the conventional approaches evaluated in this study. The results demonstrated that the model achieved strong performance metrics, including an accuracy of 99.01%, along with high precision, recall, and F-measure scores on the NSL-KDD dataset.

Alghamdi and Bellaiche (2023) proposed a hybrid model combining ensemble learning and deep learning for IDS in the IoT. Their model utilized Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), and ANN as base classifiers within the ensemble framework. The study conducted both binary and multi-class classification to detect intrusions. Preprocessing techniques, including feature engineering, were applied to improve the dataset and enhance the final classification performance. The results indicated that binary

classification outperformed multi-class classification in terms of accuracy.

Hossain and Islam (2023) employed ensemble machine learning methods, including random forest, gradient boosting, AdaBoost, XGBoost, Bagging, and stacking, to develop an IDS. Additionally, feature selection and extraction techniques such as correlation analysis, mutual information, and PCA were utilized to identify the most relevant feature subsets within the IDS dataset. The results demonstrated that random forest outperformed the other ensemble methods, achieving an accuracy of 99%.

Talukder et al. (2023) employed several machine learning and deep learning algorithms to develop a hybrid IDS. The proposed model utilized random forest, decision tree, KNN, Multi-Layer Perceptron (MLP), and CNN algorithms. The SMOTE was applied to balance the target feature values, and feature scaling was used to standardize the feature values. Additionally, the XGBoost algorithm was employed to select the most important features in the dataset. The proposed model achieved an accuracy of 99.99% on the KDD-CUP dataset.

More et al. (2024) enhanced an IDS using several supervised machine learning methods combined with a feature selection approach. They employed logistic regression, SVM, decision trees, and random forests to detect intrusions in the applied dataset. Additionally, they used correlation analysis and random sampling to improve the accuracy of the proposed model. The results demonstrated that the random forest classifier outperformed the others, achieving an accuracy of 98.63%.

Korium et al. (2024) presented a machine learning model for detecting intrusions in the internet of vehicles environment. Initially, they applied several data preprocessing techniques, including normalization, outlier handling, and feature selection using regression. Subsequently, they employed ensemble learning classifiers such as random forest, XGBoost, categorical boosting (CatBoost), and light gradient boosting (LightGBM). The final results demonstrated that the proposed model achieved a high accuracy of approximately 99.8%.

Li et al. (2024) employed feature selection and feature extraction methods alongside various machine learning techniques to optimize IDS. Their results demonstrated that feature extraction methods outperform feature selection in terms of classification performance. In this study, decision tree, random forest, KNN, naïve Bayes, and MLP were utilized as machine learning algorithms. Several preprocessing steps, including data normalization, feature elimination, handling missing values, removing duplicate records, and feature encoding, were applied to enhance the classification of normal and attack instances within the dataset. For feature selection, a feature correlation method was used to identify the most relevant features relative to the target variable. In feature extraction, PCA was employed to reduce data dimensionality. The best-performing model combined feature extraction with KNN, achieving an accuracy of 89.10% in binary classification.

Vivek and Veeravalli (2025) integrated Apriori association rule mining with ensemble learning techniques, including stacking (comprising logistic regression, random forest, and SVM) and AdaBoost (incorporating decision trees, gradient boosting, and XGBoost) models for IDS. Association rules were used to generate frequent feature sets, uncovering meaningful dependencies between features in the dataset. The results demonstrated that the proposed model outperformed versions without association rules in terms of accuracy and detection rate.

Talukder et al. (2025) presented a hybrid machine learning model for IDS that incorporates data balancing and dimensionality reduction techniques. They employed K-means clustering and the SMOTE method to balance the dataset. Additionally, PCA was used to reduce data dimensionality. Several classifiers, including decision trees, random forests, and XGBoost, were utilized to detect intrusions. The results demonstrated that the combination of K-means clustering, SMOTE, PCA, and random forest achieved the best performance, with an accuracy of 99.94%.

The aforementioned previous studies have influenced the proposed model in this research, while also presenting certain limitations. Several key influences from these studies are highlighted as follows. First, a variety of supervised learning techniques, including single classifiers, ensemble methods, and hybrid classifiers, were employed to develop models supporting IDS. Second, feature selection and extraction methods were applied to identify the most relevant features, thereby enhancing the performance of machine learning algorithms in IDS. Third, preprocessing techniques such as normalization and data balancing were utilized to improve overall model performance. Fourth, most studies used five evaluation metrics-accuracy, detection rate (recall), false alarming rate, precision, and F-measure to assess their results. Finally, the NSL-KDD dataset was commonly used to evaluate the effectiveness of the proposed models.

In addition, the main shortcomings of the aforementioned studies are outlined as follows. First, these studies have utilized data mining and machine learning techniques individually. They employed anomaly-based and misuse-based IDSs separately, applying supervised learning, unsupervised learning, and outlier detection methods independently. Second, most studies have relied more heavily on supervised learning than on unsupervised or outlier detection approaches for developing IDSs. He et al. (2024) noted that a large number of studies have focused on classification and supervised methods in IDSs, while less attention has been given to intrusion detection based on unsupervised outlier

detection methods. Third, most previous studies did not integrate unsupervised learning and outlier detection methods with supervised learning to uncover more meaningful patterns for IDS. Consequently, they were unable to fully leverage the advantages of concurrently employing supervised, unsupervised, and outlier detection methods. Fourth, association rule mining and outlier detection, two key data mining and machine learning approaches, have not been comprehensively applied in IDS research. These methods have the potential to reveal insightful patterns hidden within IDS datasets.

Based on the aforementioned influences and limitations of recent IDS studies, the novelties and contributions of this study are presented as follows.

First, this study presents an integrated hybrid framework that combines supervised, unsupervised, and outlier detection learning methods for misuse and anomaly IDS. This framework comprises four approaches, as outlined below.

Misuse detection using unsupervised learning approach: Applying clustering techniques to segment normal and attack records, combined with association rule mining to classify patterns within each cluster.

Anomaly detection using unsupervised learning approach: This involves clustering normal and attack records, applying outlier detection techniques to identify outliers within each cluster, and employing association rule mining algorithms to characterize these outliers.

Misuse detection using supervised learning approach: This method involves clustering the dataset, applying classification algorithms to distinguish between normal and attack patterns within each cluster, and selecting the best classifier for each cluster. These classifiers are then combined to form a final, comprehensive classifier.

Anomaly detection using supervised learning approach: First, clustering is applied to segment the dataset. Then, outlier detection is performed within each cluster to identify anomalies. Finally, classification algorithms are used to categorize patterns as either normal or attack outliers.

Second, since most studies have employed supervised machine learning methods for IDS, this study presents a hybrid model that integrates supervised, unsupervised, and outlier detection machine learning techniques for IDS. Specifically, clustering, association rule mining, and outlier detection methods are utilized in the proposed model. First, the dataset is divided into two clusters using clustering. Then, for each cluster, the relationships between features are identified through association rule mining to classify normal and attack records. Additionally, outlier detection is applied to distinguish normal and outlier records, which are also described using association rule mining. This study effectively combines these methods to enhance IDS performance.

In conclusion, the novel hybrid model integrates IDS with data mining and machine learning techniques, resulting in four distinct approaches. This model simultaneously employs supervised, unsupervised, and outlier detection methods to enhance both misuse and anomaly detection within IDS. This pioneering framework effectively combines data mining strategies to improve the accuracy and capability of misuse and anomaly detection models in IDS, operating concurrently to describe and predict intrusion behavior.

Background

Intrusion Detection Systems

Due to the unprecedented volume and severity of security threats targeting networks, IDSs have attracted increasing interest over the past two decades (Alkasassbeh and Al-Haj Baddar, 2023). IDSs employ advanced security techniques to detect malicious activities on hosts and networks. As a type of anomaly detection system, IDSs are used to maintain the security of networks and computer systems (Alamleh et al., 2023). These systems can identify and alert on anomalies in normal behavior, which is critical for ensuring the resilience and robustness of networks (Alkasassbeh and Al-Haj Baddar, 2023).

Data Mining Techniques

Data mining is the process of extracting valuable information from large datasets by applying various techniques to uncover hidden patterns, models, and behaviors. This study utilizes feature selection, clustering, association rules, outlier detection, and classification methods including both basic and ensemble classifiers to develop a novel hybrid model for detecting intrusive behaviors in the IDS dataset. The following paragraphs provide an overview of the key data mining techniques employed in the proposed model.

Clustering

This technique segments the dataset into multiple clusters by considering the distances between individual records. Records within a cluster are expected to be close to each other while maintaining a significant distance from records in other clusters. Various functions can be used to compute these inter-record distances. Numerous clustering algorithms are available for dataset segmentation (Khanbabaei et al., 2018). This study specifically focuses on two prominent approaches: The K-means algorithm and the Kohonen network for clustering.

K-Means Cluster Analysis

Cluster analysis is a data mining technique used to categorize cases within a dataset into distinct groups or clusters. The primary purpose of clustering analysis is to partition data by aggregating highly similar data points and separating those with low similarity. The main goal of data clustering research is to divide an unlabeled dataset

into K groups based on similarity. This task can be efficiently performed by standard algorithms, especially for low-dimensional datasets. Clustering is a widely used unsupervised machine learning approach applied in data mining and data analysis. The K-means algorithm is described as follows (Kaja et al., 2019).

If the dataset value x = 2 and k represents the number of clusters, K-means minimizes the squared error between z and y as shown in Equation (1), where y is the mean of cluster y.

If the dataset value $X = \{x_i\}, i = 1, \ldots, n$ and k represents the number of clusters, K-means minimizes the squared error between x_i and μ_k as shown in Equation (1), where μ_k is the mean of cluster C_K:

$$J(C_K) = \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 \quad (1)$$

Ultimately, K-means minimizes the sum of the squared errors across all k clusters using Equation (2):

$$J(c) = \sum_{k=1}^{n} J(C_k) \quad (2)$$

In the presented model, the K-means algorithm is employed using the Euclidean distance function. The algorithm's maximum number of iterations is set to 50, and the number of clusters varies from 2 to 4.

This study employs the K-means clustering algorithm for two primary reasons:

1. Enhanced computational efficiency: It serves as a preprocessing technique that reduces computational complexity, thereby improving the overall performance of the proposed hybrid IDS
2. Pattern identification: The algorithm enables cluster profiling of both normal and attack records. This profiling is performed using a combination of supervised, unsupervised, and outlier detection learning approaches to uncover valuable patterns within the data

The objective of using K-means in the proposed model is to cluster data into two categories: Normal records and attack records. This clustering approach effectively distinguishes between normal and attack records in the dataset and reduces the complexity of implementing subsequent algorithms in the model. K-means requires less computational power and is easier and faster to implement and understand compared to other clustering algorithms (Kaja et al., 2019). Moreover, Yang et al. (2022); Ahmad et al. (2021) noted that K-means, as a popular clustering algorithm, is the most widely used unsupervised machine learning method in IDS studies. Several studies have incorporated K-means in their hybrid IDS models, including (Sivagaminathan et al., 2023; Maseer et al., 2021; Kaja et al., 2019; Talukder et al. 2025).

Kohonen Network

The Self-Organizing Map (SOM), also known as the Kohonen network, is a type of unsupervised learning algorithm designed to reveal inherent structures within data (Weber, 2023).

When presented with an input vector x = (x1, x2, ..., xd), the SOM calculates the distances di = ‖wi - x‖ to each neuron, where wi = (wi1, wi2, ..., wid) represents the weight vector of the ith neuron among M neurons. The neuron with the smallest di becomes the best matching neuron (BMN), indicated by di = 1, while all others are set to 0. The BMN's weights, along with those of its neighbors, are updated using the Kohonen update rule: wt+1i ← wti + ht· (xt - wti). This process enables the SOM to project high-dimensional input data onto a two-dimensional grid, preserving similar patterns (Tsui et al., 2023). In this study, the SOM is employed to achieve two objectives: reducing dimensionality and determining the optimal number of clusters for subsequent input into the K-means clustering algorithm.

Feature Selection

Feature selection is a data preprocessing technique that effectively removes irrelevant and redundant features from a given task. This process reduces data dimensionality and the computational burden on machine learning algorithms, improves classification accuracy and detection rates, decreases false alarming rates, and enhances model generalization by identifying a relevant subset of features (Xu et al., 2023; Khraisat et al., 2019). To eliminate unrelated variables from the IDS dataset, this research employs a single feature selection algorithm. For example, Cavusoglu (2019) utilized feature selection techniques to improve the IDS dataset, thereby enhancing the accuracy of results extracted from machine learning algorithms.

Association Rules

Association rule mining techniques are a fundamental component of data mining, having been extensively studied and applied since their initial recognition in 1993. They are used to extract meaningful insights from large transactional databases. Specifically, these techniques provide a method to uncover hidden relationships among items or entities within transactional databases, data warehouses, or other data repositories. The traditional approach to evaluating the quality of association rules in a given problem relies on three key metrics: Support, confidence, and lift (Diaz-Garcia et al., 2023). An association rule can be represented in an if-then format with antecedent and consequent parts: $X \rightarrow Y(c, s)$, where X and Y belong to the set I. In this rule, s and c denote the support and confidence indices, respectively. Support indicates the proportion of records in which X and Y appear together. Confidence is the ratio of the number of records containing both X and Y to the number of records

containing only X (Diaz-Garcia et al., 2023). The association rule algorithm can be applied to classification problems by incorporating the target variable into the consequent part of the rules. In this context, the Apriori algorithm is one of the most widely used algorithms within the association rule framework. If we have a collection of n transactions, where $T = \{T_1, T_2, \dots, T_n\}$ and I is the set of all items, denoted as $I = \{i_1, i_2, \dots, i_m\}$, and also T_j is a subset of T with $1 \leq j \leq n$, then T_j belongs to I (Diaz-Garcia et al., 2023).

In this study, the Apriori algorithm is employed in the proposed model for two purposes:

(1) Classifying normal and attack behaviors in the dataset
(2) Describing the behavior of normal and attack outliers within the dataset

For the first objective, the Apriori algorithm is integrated with clustering to classify normal and attack records within each cluster. For the second objective, a combination of the Apriori algorithm, clustering, and outlier detection techniques is used to detect and characterize normal and attack outliers in each cluster. The Apriori algorithm is a popular unsupervised learning technique used to mine frequent itemsets and uncover relationships between itemsets in a dataset. It generates a variety of if-then rules, consisting of antecedent and consequent parts, to describe the relationships between items (Vivek and Veeravalli, 2025). In this study, the Apriori algorithm is applied to identify attractive, simple, and interpretable if-then rules that reveal relationships between features and the target feature, facilitating the classification of normal and attack records as well as the detection of normal and attack outliers in the IDS.

Outlier Detection

An event or observation is considered an outlier or anomaly if it is uncommon, intrusive, or suspicious and occurs at an irregular distance from the population (Sikder and Batarseh, 2023). Outlier detection can identify unusual patterns caused by these outliers. In this study, outlier detection is applied within each cluster to identify outliers, including both normal and attack records within the dataset.

Outlier detection methods can be categorized in various ways, including statistical, density-based, clustering, distance-based, learning-based, deviation-based, and ensemble techniques (Sikder and Batarseh, 2023). In this study, the outlier detection approach employed is based on deviations. Han et al. (2012) stated that this method identifies outliers by determining the primary characteristics of records within a group, with records that deviate significantly being classified as outliers. For a dataset D containing n records, subsets $\{D\_1, D\_2, \dots, D\_m\}$ are created, where $2 \leq m \leq n$. Outliers are detected using a dissimilarity function defined as follows: if the records within a subset are similar, the function returns a lower value; if they are dissimilar, it returns a higher value. The dissimilarity function is expressed in Equation (3) as follows:

$$DF = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \quad (3)$$

Where n is the number of records $\{x_1, \dots, x_n\}$ and $\bar{x}$ is the mean of these n records in the dataset.

Classification

Classification is one of the most common data mining tasks used to categorize records within a dataset. In classification, there is a target variable that the constructed model uses to classify records based on this variable (Khanbabaei et al., 2018). This study employs several classification algorithms to accurately distinguish between normal and attack records within the IDS.

Decision Tree

A decision tree is one of the most popular non-parametric data mining classification techniques. It classifies the values of the target (dependent) variable through an if-then structure utilizing independent variables. This technique employs a top-down induction process using recursive partitioning to form branches based on the most informative attributes. The decision tree splits specific subsets according to attribute values. Finally, leaves, which serve as the terminal nodes, are produced through recursive partitioning and are assigned their respective class values (Ahmad et al., 2021).

One of the most widely used decision tree algorithms is the C4.5 classifier, developed. It is based on the concept of information entropy, denoted as (S), as presented by Kaja et al. (2019) and shown in Equation (4):

$$Entropy\ (S) = -p_1 \log_2(p_1) - p_2 \log_2(p_2) \quad (4)$$

In Equation (4), p_1 and p_2 represent the fractions of class values 1 and 0, respectively, in sample S. During data splitting, the C4.5 algorithm calculates the entropy difference measure, also known as normalized information gain, based on the selected attribute. The attribute with the highest entropy difference is chosen for decision-making, and this process continues recursively on smaller subsets using the partitioning method. In this study, the C5 algorithm, an improved version of C4.5, is used to construct the decision tree. A 10-fold cross-validation technique is applied to divide the dataset into training and testing sets. Two measures, information gain and gain ratio, are employed for tree construction. The maximum tree depth is limited to 20. Pruning and pre-pruning strategies are applied to optimize the decision trees. The confidence level, minimal gain, minimal leaf

size, and minimal size for split indices are set to 0.25, 0.1, 2, and 4, respectively.

Information Gain

This popular algorithm selects relevant features from the dataset by retaining those with high information gain and excluding others. The information gain measure is defined based on the entropy concept described earlier. A higher information gain corresponds to a feature that provides substantial information within the feature set (Bakro et al., 2023).

Gain Ratio

This is another common metric used to select features in a dataset. It improves upon the information gain measure by indicating how effectively each feature can create branches to split the data. In constructing a decision tree, the gain ratio is employed to divide data and split features, thereby forming branches and boundaries (Bakro et al., 2023).

Multi-Layer Perceptron

Artificial Neural Networks (ANNs) are non-parametric data mining methods used in machine learning to model non-linear relationships between input and target variables (Choras and Pawlicki, 2021). A common type of ANN is the Multi-Layer Perceptron (MLP), which typically consists of three layers: input, hidden, and output. Each layer contains multiple interconnected neurons, whose outputs are calculated by applying a weighted sum followed by an activation function. The output of a neuron is computed using Equation (5):

$$O_i = f\left(b_i + \sum_{j=1}^{M} W_{ij} x_j\right) \quad (5)$$

In Equation (5), W_{ij} represents the weight connecting input j to neuron i, and b_i is the bias for neuron i. In this study, the logistic (sigmoid) activation function, described in Equation (6).

Was utilized for MLP:

$$f = \frac{1}{1+e^{-x}} \quad (6)$$

During the model generation phase, the weights and biases are adjusted to minimize the objective function and prevent overfitting. The iterative process is based on the gradient descent learning method. The number of neurons in the input, hidden, and output layers was 17, 19, and 1, respectively. The learning rate and momentum values were set to 0.05 and 0.9, respectively. Additionally, the training cycle was set to 50 iterations. To construct the neural network, the features (variables) were normalized.

Logistic Regression

Logistic regression is a classification model used to predict a binary response for the target variable in a dataset. For example, the response variable y can be assigned a value of 0 or 1 to represent normal and attack records, respectively. Kasongo and Sun (2020) stated that if x is a column vector of M descriptive variables, then $\pi = \Pr(y = 1|x)$represents the response probability. For N observations, logistic regression is based on Equation (7), where α is the intercept parameter and β^T encompasses the variable coefficients:

$$Logit\ (\pi) = \log\left(\frac{\pi}{1-\pi}\right) = \alpha + \beta^T x \quad (7)$$

In logistic regression, the kernel type is set to radial. The kernel gamma and kernel cache size are set to 1 and 20, respectively. Additionally, the parameter C is set to 1.5. The convergence epsilon and maximum number of iterations for training the logistic regression model are set to 0.001 and 100,000, respectively.

K-Nearest Neighbour

K-Nearest Neighbor (KNN) is one of the simplest and non-parametric machine learning classifiers. This algorithm calculates the distance between training samples using measures such as the Euclidean distance (Kayode et al., 2022). In this study, the value of k is set to 4, meaning each instance has four neighbors. A new instance is classified based on the majority class of its neighbors (Drewek-Ossowicka et al., 2021). The distance metric used to compute the distance between instances is the Euclidean distance.

Support Vector Machine

Support Vector Machine (SVM) uses a hyperplane to differentiate between positive and negative classes in the target variable. It is a powerful and robust algorithm for generalization and optimization purposes. However, tuning its parameters is a difficult and challenging task (Kayode et al., 2022). In SVM, a kernel function is employed to map data into a high-dimensional space for better separation. Several kernel functions exist, including linear, quadratic, and Gaussian kernels (Bhati and Khari, 2021). In this study, the SVM uses the Radial Basis Function (RBF) kernel to distinguish the data. The kernel degree and kernel cache size are set to 2 and 200, respectively. The C parameter is set to zero. Additionally, the convergence epsilon and maximum number of iterations for implementing the SVM algorithm are set to 0.001 and 100,000, respectively.

Naïve Bayes

This algorithm employs Bayes' theorem to compute the probability that an instance belongs to each class in the target variable. The purpose of the algorithm is to identify the most probable class for each record or instance. Naïve Bayes assumes that the variables in the dataset are independent of each other (Kaja et al., 2019; Kayode et al., 2022).

Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a machine learning technique that employs a linear function to classify records in a dataset into binary or categorical classes. This method can also be used for dimensionality reduction. LDA applies Bayes' theorem to estimate the probability that a new record belongs to a particular class (Saranya et al., 2020). In this study, a simple LDA model is used to classify records as either normal or attack.

Random Forest

Random forest is an ensemble learning method that combines multiple decision trees and enhances the final outcome through the bagging strategy. This algorithm constructs several trees to reduce variance and mitigate overfitting issues (Hossain and Islam, 2023; Kaja et al., 2019). In this study, 10 trees are used in the random forest. The maximum depth of the trees is set to 20, and a pruning strategy is applied. The confidence value is set at 0.25. The minimum gain, minimum leaf size, and minimum split size are configured as 0.1, 2, and 4, respectively. Finally, a majority voting strategy is employed. Within the random forest, the criterion for the decision trees is selected from “information gain” and “gain ratio.” For each cluster, the decision tree criterion that yields better classification accuracy is chosen.

AdaBoost

Adaptive boosting (AdaBoost) is an ensemble machine learning algorithm that enhances the performance of the final classifier. This algorithm demonstrates strong generalization capabilities and effectively mitigates overfitting in model generalization (Hossain and Islam, 2023; Kaja et al., 2019). When applying AdaBoost to each cluster, a base learner with superior classification accuracy is selected.

Bagging

Bootstrap aggregation (Bagging) is an ensemble learning algorithm that uses majority voting to enhance the performance of base classifiers. This method employs bootstrap sampling to randomly partition the training dataset into multiple subsets. A classifier is trained on each subset to classify the records. In this study, a voting strategy is applied to determine the final outcome (Khanbabaei et al., 2023). When applying Bagging to each cluster, the base learner selected is the one that demonstrates higher classification accuracy compared to the other classifiers.

Stacking

Stacking combines multiple classifiers to achieve improved classification accuracy. This algorithm relies on the ensemble learning strategy to construct classifiers. Stacking involves using a set of base learners and integrating their outputs to produce a final result, which is then used by a meta-learner (Cavusoglu, 2019; Koutanaei et al., 2015). In this study, when applying stacking within each cluster, both the base learners and the meta-learner are selected based on their superior classification accuracy compared to other classifiers.

Materials and Methods

Dataset Description

However, some researchers have used non-real-life datasets, such as the KDD-99 dataset and the DARPA IDS evaluation dataset, to train and test their proposed models. In this study, the NSL-KDD standard dataset is utilized. The NSL-KDD dataset is a modified and improved version of the well-known KDD Cup 99 dataset (Yang et al., 2022).

The NSL-KDD dataset is one of the most commonly used datasets in IDS research (Yang et al., 2022; Ahmad et al., 2021; Naseri and Gharehchopogh, 2022; Aldweesh et al., 2020). Ahmad et al. (2021) reported that 60 percent of studies utilized either the KDD CUP 99 dataset (24 percent) or the NSL-KDD dataset (36 percent) to evaluate their proposed models. Numerous studies, including Ahmad et al. (2021); Drewek-Ossowicka et al. (2021); Yang et al. (2022); Cavusoglu (2019); Bhati and Khari (2019), have employed the NSL-KDD dataset to validate their models. This dataset serves as a benchmark for researchers to compare various IDS approaches developed using data mining and machine learning algorithms. Consequently, results from different studies using this dataset are directly comparable (Yang et al., 2022; Khraisat et al., 2019).

The dataset's features (attributes) are categorized into four groups: Basic, content, time-based traffic, and host-based traffic features. It comprises 42 attributes, as shown in Box 1. These include six binary attributes, three nominal attributes, and the remaining attributes are numeric. The target attribute consists of five classes: One normal class and four attack classes (Table 1) This indicates that connections are classified into two main groups:

1) Attack and
2) Normal

Attack connections encompass 39 types, divided into four categories:

1) Denial of Service (DoS)
2) Remote to Local (R2L)
3) User to Root (U2R)
4) Probing

The dataset contains 22 identified attack types and 17 unidentified attack types, which are embedded in the training and testing datasets, respectively (see Table 2).

Box 1: Features of the NSL-KDD dataset

Features of the NSL-KDD dataset
Duration, protocol_type, service, flag, src_bytes, dst_bytes, land, wrong_fragment, urgent, hot, num_failed_logins, logged_in, num_compromised, root_shell, su_attempted, num_root, num_file_creations, num_shells, num_access_files, num_outbound_cmds, is_host_login, is_guest_login, count, srv_count, serror_rate, srv_serror_rate, rerror_rate, srv_rerror_rate, same_srv_rate, diff_srv_rate, srv_diff_host_rate, dst_host_count, dst_host_srv_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate, dst_host_rerror_rate, dst_host_srv_rerror_rate, label

Table 1: Number of records for each label of the target feature in the NSL-KDD dataset

NSL-KDD dataset	Total records	Labels of the target feature				
		Probe	R2L	U2R	DOS	Normal
Number of records in the NSL-KDD Training dataset	125973	11656	995	52	45927	67343
Number of records in the NSL-KDD Testing dataset	22544	2422	2887	67	7458	9710

Types of attacks in the NSL-KDD dataset are defined as follows (Cavusoglu, 2019; Khraisat et al., 2019):

DOS: DOS attacks cause an increase in network traffic, which can prevent the system from providing service

U2R: In this attack, an intruder gains access to a normal user's account

R2L: R2L attacks occur when an attacker gains local access to a system remotely

Probe: This attack scans the network to identify vulnerabilities for exploitation

Classification Performance Evaluation

Five metrics are used to evaluate the classification results as follows:

1) Classification accuracy
2) Detection rate (also called recall or sensitivity)
3) False alarming rate
4) Precision
5) F-measure

To define these metrics, the concept of the confusion matrix must be introduced, as shown in Figure 1. This concept is applied to compare classification models (Ahmad et al., 2021).

In the confusion matrix, four definitions are presented as follows (Ahmad et al., 2021).

True Positive (TP): The number of attack records correctly predicted.

False Negative (FN): the number of records incorrectly classified as normal actions when they are actually attack actions.

True Negative (TN): The number of normal records correctly identified as normal.

False Positive (FP): The number of records incorrectly classified as attack actions when they are actually normal actions.

Following the explanation of the confusion matrix, five metrics for evaluating classification models are presented using Equations (8) to (12). As noted by Yang et al. (2022); Ahmad et al. (2021), these five metrics are preferred in most studies over alternative measures:

$$\text{Classification accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \text{ (Yang et al., 2022)} \quad (8)$$

$$\text{Detection rate (Recall)} = \frac{TP}{TP+FN} \text{ (Yang et al., 2022)} \quad (9)$$

$$\text{False alarming rate} = \frac{FP}{FP+TN} \text{ (Yang et al., 2022)} \quad (10)$$

$$\text{Precision} = \frac{TP}{TP+FP} \text{ (Yang et al., 2022)} \quad (11)$$

$$\text{F} - \text{measure} = 2 * \frac{precision*recall}{precision+recall} \text{ (Yang et al., 2022)} \quad (12)$$

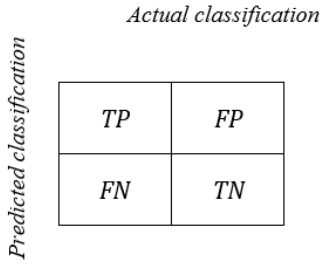

Fig. 1: Confusion matrix

Table 2: Attacks in training and testing datasets by attack class

Attack class	Attacks in the training data	Attacks in the testing data
Probe	Ipsweep, Nmap, Portsweep, Satan	Mscan, Saint
DOS	Back, Land, Neptune, Pod, Smurf, Teardrop	Apache2, Mailbomb, Processtable, Udpstorm
U2R	Buffer_overflow, Loadmodule, Perl, Rootkit	Ps, Httptunnel, Xterm, Worm
R2L	Ftp_write, Guess_password, Imap, Multihop, Phf, Spy,Warezclient, Warezmaster	Named, Snmpgetattack, Xlock, Xsnoop, Snmpguess, Sendmail, Sqlattack

Classification accuracy is the proportion of correctly classified normal and attack records relative to the total number of records. The detection rate represents the percentage of attack records that have been successfully identified. The false alarming rate is defined as the number of normal records incorrectly labeled as attack records divided by the total number of normal records (Yang et al., 2022). Precision is calculated by dividing the number of correctly classified attack records by the total number of records classified as attacks. The F-measure is the harmonic mean of precision and recall (Ahmad et al., 2021).

The compound misuse and anomaly intrusion detection system

In this section, the proposed compound misuse and anomaly IDS is illustrated in Figure 2 and comprises three phases.

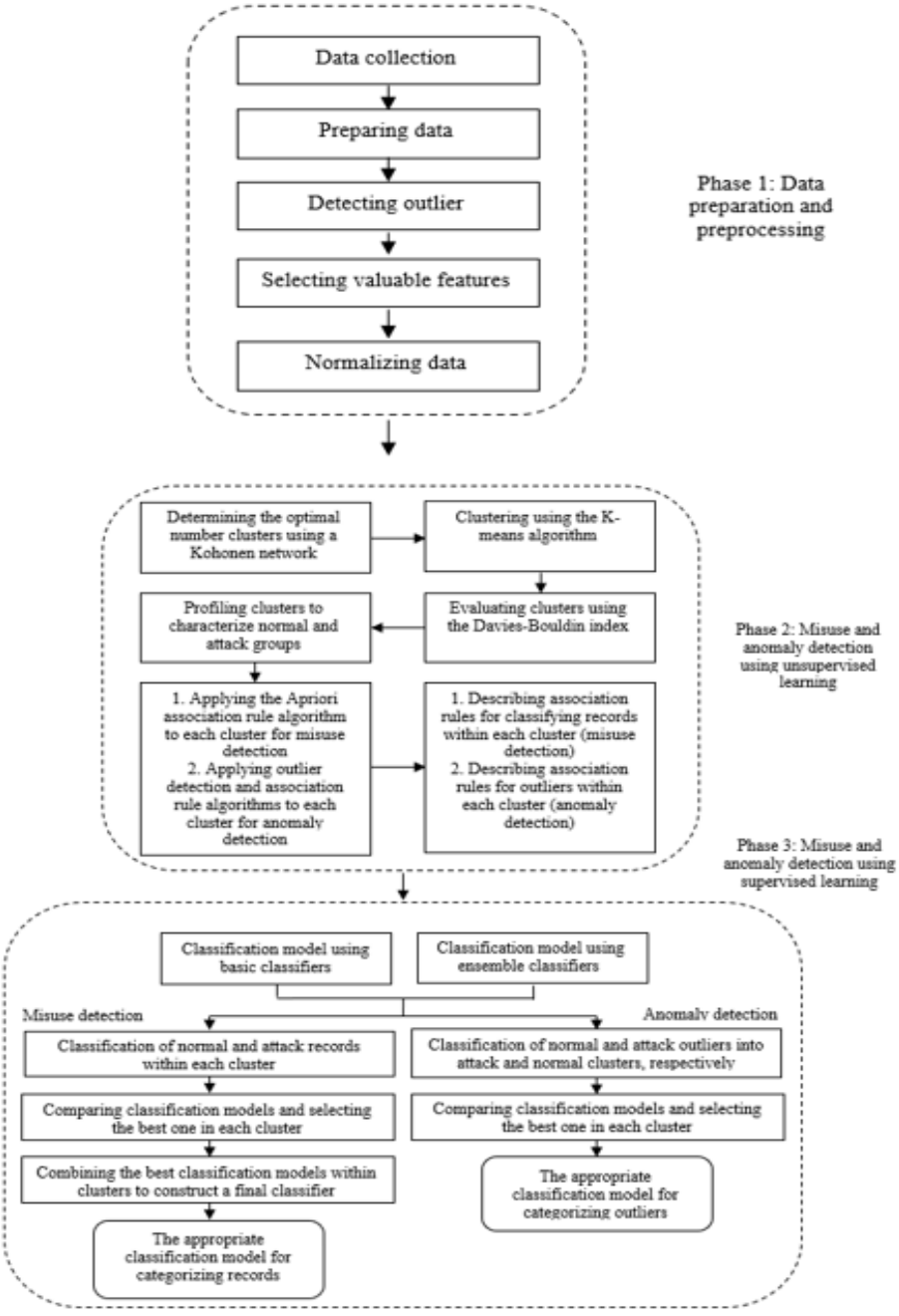

Fig. 2: Proposed compound misuse and anomaly intrusion detection system

1) Data preparation and preprocessing
2) Misuse and anomaly detection using unsupervised learning
3) Misuse and anomaly detection using supervised learning

In the first phase, several steps are undertaken for data preparation and preprocessing, as outlined below:

1. Collecting data
2. Selecting an appropriate tabular data format
3. Removing records with missing values
4. Eliminating attributes with missing values
5. Removing attributes that contain unique values
6. Removing outliers using deviation-based outlier analysis

Following data preparation and outlier detection, a feature selection technique is employed to identify the most relevant features for developing a more effective IDS model. Subsequently, the min-max normalization method is applied exclusively to the numeric attributes in the dataset.

In the second phase, the Kohonen network algorithm is initially employed to determine an improved initial number of clusters, which serves as an input parameter for the K-means clustering algorithm. Subsequently, the K-means algorithm is tested and evaluated using the Davies-Bouldin index to identify the optimal clustering model. Following this, cluster profiling is developed to describe the characteristics of each cluster. For misuse detection using an unsupervised learning approach, the association rule mining algorithm is applied to characterize normal and attack behaviors, with lower support and confidence thresholds used for attack and normal clusters, respectively. In anomaly detection, also using an unsupervised learning approach, an outlier detection technique is first deployed to identify normal and attack outliers within attack and normal clusters, respectively. Then, the association rule mining algorithm is applied to describe these normal and attack outliers in the corresponding clusters, again using lower support and confidence thresholds.

In the third phase, classification techniques are applied to each cluster to classify the records in the IDS. The techniques included are as follows:

1. Classic classifiers
2. Decision tree (gain ratio)
3. Decision tree (information gain)
4. Neural network
5. Logistic regression
6. K-Nearest Neighbour (KNN)
7. Support Vector Machine (SVM)
8. Naïve Bayes
1. Linear Discriminant Analysis (LDA):

1. Ensemble classifiers
2. Random forest
3. AdaBoost
4. Bagging
5. Stacking

The primary objective of the proposed model is to develop a novel compound IDS that improves the identification of intrusion behavior. The application of classification techniques (supervised learning methods) in the third phase constitutes only a part of the overall model. Future research could explore alternative classification techniques within this framework. The 10-fold cross-validation method is employed to train and evaluate the classification models.

In misuse detection using a supervised learning approach, classification techniques are initially employed to categorize normal and attack records within each cluster. Subsequently, the results are compared, and the best classification models from each cluster are combined to construct a final classifier.

To compare the classification algorithms, five metrics are used:

1. Classification accuracy
2. Detection rate
3. Precision
4. F-measure
5. False alarming rate

In anomaly detection using a supervised learning approach, after clustering and applying outlier analysis in the second phase of the model, classification models are employed to distinguish between normal and attack outliers, assigning them to the respective attack and normal clusters.

Results and Discussion

This section presents and discusses the results of the proposed model using the NSL-KDD dataset. The results are explained and analyzed using SPSS Modeler 18, WEKA (Waikato Environment for Knowledge Analysis) version 3.8.6, and RapidMiner Studio 7.1.

Data Preparation and Preprocessing Phase

Data Sampling and Preparation

Referring to Table 1, a random selection approach allocates 50% of the training dataset and 10% of the testing dataset for training and testing the proposed model, respectively (see Table 3). As described in Section 4.3, a data preprocessing technique is applied to improve the overall performance of the proposed model.

Table 3: Number of records in the sample training and testing datasets

Sample training dataset	Sample testing dataset	Sample total dataset
62986	2254	65240

Feature Selection

As a preprocessing technique, feature selection enhances the performance of both descriptive and predictive data mining models. In the proposed model, feature selection is based on the correlation between feature subsets and the class, as well as the intercorrelation among features, as described by Yang et al. (2022); Cavusoglu (2019). According to Weka version 3.8.6, the evaluator function in this method assesses the subset value of attributes by considering the individual predictive ability of each feature along with the degree of redundancy among them. Subsets of features with strong correlation to the target feature and low intercorrelation are preferred. The search method used in the feature selection algorithm is based on the "best first" approach.

To select the appropriate features, the correlation between each independent feature and the target feature (label) is initially computed individually. Features exhibiting a high correlation with the target feature are selected, while those with low correlation are removed from the dataset. The selected features have a greater predictive ability for the target feature. Next, the correlation among the selected features is calculated, and from each group of correlated features, only one is retained while the others are removed. In other words, the first stage involves calculating the correlation between all independent features and the target feature, and the second stage involves computing the inter-correlations among the features selected in the first stage. After completing the feature selection process, seven key features from the dataset are chosen for deployment in the model (see Table 4). Notably, the selected features include a combination of nominal and numeric attributes.

These seven features exhibit the strongest relationships with the target variable in the dataset, providing a superior level of explanatory power compared to other features. They are more highly correlated with the target label and can more effectively distinguish between normal and attack activities. Feature selection reduces the number of features in the dataset, thereby enhancing data mining and machine learning models for intrusion detection. Using fewer features decreases the complexity of the extracted patterns. Both supervised and unsupervised data mining approaches can utilize these selected features to develop misuse and anomaly-based IDS. Consequently, the IDS becomes more flexible and simpler, improving its ability to detect network attacks.

Min-Max Normalization

Min-max normalization is used as a data preprocessing method to improve results and reduce computational errors (Cavusoglu, 2019). This normalization technique is calculated using Equation (13), where $New_{min} = 0$ and $New_{max} = 1$:

$$x_i = New_{min} + (New_{max} - New_{min}) \times (\frac{x_i - x_{min}}{x_{max} - x_{min}}) \quad (13)$$

Misuse and Anomaly Detection Using Unsupervised Learning Phase

In the second phase of the proposed model, the appropriate number of clusters was determined using the Kohonen network algorithm. The neighborhood size, number of epochs, and learning rate were set to 7, 100, and 0.5, respectively. It is important to note that the distance metric, neighborhood function, and learning rate decay were Euclidean, rectangular, and linear, respectively. Additionally, the width and length of the two-dimensional output map were set to 10 and 7, respectively. Table 5 presents the other parameter values of the Kohonen network algorithm.

As shown in Table 5, during the first and second phases, two neighboring neurons and one neighboring neuron, respectively, are influenced by the training process at each update step. Additionally, the initial learning rates (eta) controlling the adaptation of the map are set to 0.3 and 0.1 for the first and second phases, respectively. The number of training iterations to adjust the network weights is set to 20 in the first phase and 150 in the second phase.

Tables 6 and 7 present the number of records in each cluster and the results obtained from applying the Kohonen network algorithm, respectively.

As shown in Table 6, the Kohonen network algorithm identified an appropriate number of clusters four in total to segment all records in the dataset. Within each of these four clusters, the records are similar to one another and distinct from those in the other clusters. The feature values of the records within a single cluster exhibit high similarity. Furthermore, Table 7 presents the topology or structure of the Kohonen network used to cluster the dataset.

Table 4: Selected features from the NSL-KDD dataset

Feature ID	4	5	6	12	26	30	37
Feature name	flag	Src _bytes	Dst _bytes	logged_in	Srv _serror _rate	Diff _srv _rate	dst_host _srv_diff _host_rate
Type of feature	nominal	numeric	numeric	nominal	numeric	numeric	numeric

Table 5: Parameter values of the Kohonen network algorithm

	Neighborhood	Initial Eta	Cycles
Phase 1	2	0.3	20
Phase 2	1	0.1	150

Table 6: Number of records per cluster

Cluster number	1	2	3	4
Number of records (transactions)	24446	16753	13404	10637

This network consists of 17 input layers and 12 output layers (calculated as new field Y = 3 * new field X = 4). In the input layer, each neuron represents a feature of the dataset and passes the input values to the Kohonen layer. The output layer comprises a 2D grid of neurons, each representing a cluster center. These neurons compute a similarity measure between the input data and their associated weight vectors. The neuron with the smallest similarity measure value is declared the winner of the competition.

After determining the optimal number of clusters using the Kohonen network algorithm, the K-means algorithm was applied for cluster analysis. The Kohonen algorithm identified four clusters as the appropriate number for the dataset. Subsequently, the K-means algorithm was employed to cluster the dataset under three scenarios, involving 2, 3, and 4 clusters. To evaluate the quality of clustering, the Davies-Bouldin index was used. Based on the results from the Kohonen network (see Table 6), the K-means clustering algorithm was implemented for the three models corresponding to 2, 3, and 4 clusters. The clustering results, assessed by the Davies-Bouldin index, are presented in Figure 3

Figure 3 presents the Davies-Bouldin index values for three clustering models. As shown in Figure 3, the clustering model with two clusters outperforms the others, as it has the lowest Davies-Bouldin index value. The Davies-Bouldin index evaluates the quality of clustering models by measuring the similarity among records within clusters. Ideally, records within the same cluster should be similar to each other and dissimilar to those in other clusters. Similarity is typically computed using distance measures such as the Euclidean distance. A lower Davies-Bouldin index indicates a better clustering model. In Figure 3, the model with two clusters exhibits the lowest index value, indicating it is the optimal choice. Therefore, the best number of clusters is set to two. Table 8 shows the number of records in each cluster resulting from this optimal clustering model.

Tables 9 and 10 present cluster profiles for clusters 1 and 2, which predominantly consist of attack and normal records, respectively. As shown in Table 9, cluster 1 contains 99.28% of the attack records. Table 10 indicates that 96.56% of the records in cluster 2 are normal. In this paper, clustering is used as a data preprocessing technique to improve the performance of both misuse and anomaly intrusion detection.

Misuse Detection Using Unsupervised Learning Approach

In this section, unsupervised learning techniques are employed to classify normal and attack records within the IDS. This approach comprises two main stages:

(1) Clustering the dataset to identify attack and normal clusters, as detailed in Section 5.2 and Tables 8–10
(2) Applying association rule mining to classify records within the identified attack and normal clusters

By integrating clustering and association rule mining, this method offers a novel approach to misuse detection, enhancing the classification of normal and attack records in the IDS dataset.

Table 7: Output results from the implementation of the Kohonen network algorithm

Number of output layer	Number of input layer	New filed Y in the Kohonen output grid	New filed X in the Kohonen output grid
12	17	3	4

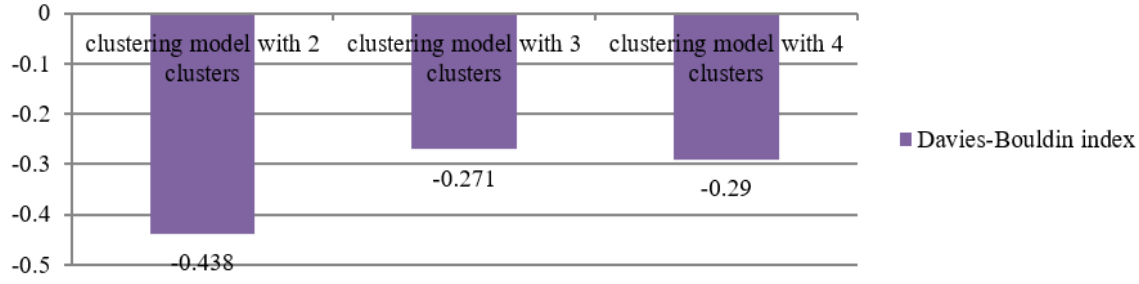

Fig. 3: Davies-Bouldin index values for each clustering model

Table 8: Number of records in each cluster

Number of records in the first cluster	Number of records in the second cluster
29597	35643

Table 9: Average feature values in the first cluster (attack cluster)

Feature name	dst_host_srv_diff_host_rate	Diff_srv_rate	Srv_serror_rate	logged_in	Dst_bytes	Src_bytes	flag	class
value	0.039	0.108	0.6	0>99.91%	27763.741	39851.809	S0>59.67 %	Attack>99.28 %

Table 10: Average feature values in the second cluster (normal cluster)

Feature name	dst_host_srv_diff_host_rate	Diff_srv_rate	Srv_serror_rate	logged_in	Dst_bytes	Src_bytes	flag	class
value	0.026	0.028	0.008	1>72.77%	4081.928	15957.537	SF>94.9%	Normal>96.59%

In this approach, after clustering, the Apriori association rule mining technique is applied within each cluster to classify normal and attack patterns, comprising two components: 1. Antecedent and 2. Consequent. The antecedent represents the relationships between features, while the consequent indicates the normal or attack labels of the records based on the target (class) attribute. By using the association rule mining technique, classification accuracy and detection rate can be improved within each cluster, as it clarifies the behaviors that help identify new normal and attack records in each cluster.

To apply the Apriori algorithm, continuous variables must first be converted into categorical variables. This transformation is achieved through a discretization technique, which divides the range of continuous values into five equal intervals, labeled from 1 to 5, spanning from the minimum to the maximum value.

In each cluster, there are two types of association rules characterized by low and high support and confidence. In the normal cluster, association rules exhibit both low and high support and confidence for attack and normal behaviors, respectively; the opposite pattern is observed in the attack cluster. This approach aims to elucidate the behaviors of normal and attack records within the attack and normal clusters, respectively, to improve classification accuracy and detection rate. Subsequently, lower values of support and confidence are considered within each cluster. In clusters 1 and 2, the minimum support and confidence thresholds for describing the behaviors of normal and attack records were 0.4% and 5%, respectively. Tables 11 and 12 present five examples of rules for classifying normal and attack behaviors in the attack and normal clusters, respectively.

Table 11 presents five examples of association rules used to classify normal records within the attack cluster. The association rule mining algorithm is an unsupervised learning method. Based on the misuse detection approach, this algorithm is applied to classify normal records (behaviors) in the attack cluster. Typically, an attack cluster contains many rules with high support and confidence values that classify attack records. However, the objective here is to classify normal records within the attack cluster. Therefore, association rules with low support and confidence values are considered for classifying normal records in the attack cluster. For example, as shown in Table 11 (row 4), one of the rules in the attack cluster for classifying normal behavior is as follows: If “srv_serror_rate” equals 3, then the label type is normal, with support and confidence values of 0.603% and 57.273%, respectively.

Table 11: Five examples of association rules for classifying normal behaviors within an attack cluster

Row	Antecedent	Consequent	Support (%)	Confidence (%)
1	srv_serror_rate = 3 src_bytes = 0 dst_host_srv_diff_host_rate = 1 dst_bytes = 0 logged_in = 0	Class = normal	0.581	58.491
2	srv_serror_rate = 3 logged_in = 0	Class = normal	0.592	58.333
3	srv_serror_rate = 3 dst_host_srv_diff_host_rate = 1 logged_in = 0	Class = normal	0.592	58.333
4	srv_serror_rate = 3	Class = normal	0.603	57.273
5	srv_serro_rate = 3 dst_host_srv_diff_host_rate = 1	Class = normal	0.603	57.273

Table 12: Five examples of association rules for classifying attack behaviors within a normal cluster

Row	Antecedent	Consequent	Support (%)	Confidence (%)
1	diff_srv_rate = 5 logged_in = 1 dst_host_srv_diff_host_rate = 1 srv_serror_rate = 1	Class = attack	0.632	8.696
2	diff_srv_rate = 5 logged_in = 1 dst_host_srv_diff_host_rate = 1 srv_serror_rate = 1 src_bytes = 1	Class = attack	0.632	8.696
3	diff_srv_rate = 5 logged_in = 1 flag = SF	Class = attack	0.646	8.511
4	diff_srv_rate = 5 logged_in = 1 dst_host_srv_diff_host_rate = 1 src_bytes = 1	Class = attack	0.646	8.511
5	diff_srv_rate = 5 logged_in = 1 flag = SF srv_serror_rate = 1	Class = attack	0.646	8.511

Likewise, Table 12 presents five examples of association rules that classify attack records within the normal cluster. While some rules in the normal cluster can classify normal records with high support and confidence values, the objective here is to identify attack records grouped within the normal cluster. Therefore, association rules with the lowest support and confidence values are selected to classify these attack records within the normal cluster. For example, row 3 in Table 12 shows if diff_srv_rate = 5, logged_in = 1, and flag = SF, then the label type is attack, with support and confidence values of 0.646% and 8.511%, respectively.

Anomaly Detection Using Unsupervised Learning Approach

This section introduces a novel approach that leverages unsupervised learning methods for anomaly detection in IDS. The implementation of this approach involves the following stages:

(1) Clustering the dataset records (see Section 5.2 and Tables 8–10)
(2) Applying an outlier detection technique to identify attack outliers within the normal cluster and normal outliers within the attack cluster
(3) Utilizing association rule mining to characterize the behaviors of normal and attack outliers in their respective clusters

The proposed approach contributes a new model that integrates three unsupervised learning techniques clustering, outlier detection, and association rule mining to effectively detect anomalies in IDS datasets.

In this approach, after performing cluster analysis, an outlier detection technique is applied to each cluster. There are two types of outliers. In the normal cluster, outlier detection primarily identifies outliers as attack records, and vice versa. Within the normal cluster, applying outlier detection to 1% of the total dataset reveals three peer groups, with the number of outliers in peer groups 1, 2, and 3 being 0, 351, and 5, respectively. Similarly, in the attack cluster, applying outlier detection to 1% of the total dataset identifies two peer groups, with 263 and 32 outliers in peer groups 1 and 2, respectively.

After performing outlier analysis to identify normal and attack outliers within each cluster, the association rule mining technique is applied to characterize the behaviors of normal and attack outliers in the attack and normal clusters, respectively, using lower support and confidence thresholds. This approach reduces the false alarming rate by enabling more accurate recognition of outlier behavior. Tables 13 and 14 present five examples of association rules that describe the behaviors of normal and attack outliers in the attack and normal clusters, respectively.

Table 13 presents five examples of association rules that characterize the behavior of normal records within the attack cluster. These normal records are considered outliers in the attack cluster, exhibiting normal behavior. This represents an outlier detection approach using unsupervised learning. Specifically, the association rule mining algorithm, as an unsupervised method, can identify and describe normal outliers within the attack cluster. The association rules, defined by minimum support and confidence thresholds in the attack cluster, effectively characterize these normal outliers. For example, as shown in Table 13 (row 4), one of the outlier behaviors in the attack cluster that characterizes normal outlier behavior is as follows:

Table 13: Five examples of association rules illustrating the behavior of normal outliers within an attack cluster

Row	Antecedent	Consequent	Support (%)	Confidence (%)
1	dst_host_srv_diff_host_rate = 2 flag = S0	Class = normal	0.625	9.091
2	dst_host_srv_diff_host_rate = 2 flag = S0 src_bytes = 0 dst_bytes = 0 logged_in = 0	Class = normal	0.625	9.091
3	dst_host_srv_diff_host_rate = 2 srv_serror_rate = 5 flag = S0 dst_bytes = 0	Class = normal	0.569	10
4	dst_host_srv_diff_host_rate = 2 src_bytes = 0	Class = normal	0.739	23.077
5	Diff_srv_rate = 4 dst_host_srv_diff_host_rate = 1 dst_bytes = 0	Class = normal	1.08	5.263

Table 14: Five examples of association rules illustrating the behavior of attack outliers within a normal cluster

Row	Antecedent	Consequent	Support (%)	Confidence (%)
1	logged_in = 1 flag = SF diff_srv_rate = 1	Class = attack	64.006	7.294
2	logged_in = 1 flag = SF dst_host_srv_diff_host_rate = 1 dst_bytes = 1	Class = attack	64.006	7.294
3	logged_in = 1 flag = SF dst_bytes = 1 src_bytes = 1	Class = attack	64.006	7.294
4	logged_in = 1 flag = SF diff_srv_rate = 1 dst_host_srv_diff_host_rate = 1 srv_serror_rate = 1	Class = attack	64.004	7.294
5	flag = SF diff_srv_rate = 1 dst_host_srv_diff_host_rate = 1 dst_bytes = 1	Class = attack	63.855	7.311

If “dst_host_srv_diff_host_rate” and “src_bytes” are set to 2 and 0, respectively, then the label type is normal. This rule has a support of 0.739% and a confidence of 23.077%.

Furthermore, Table 14 presents five examples of association rules that characterize attack records within the normal cluster. These attack records are regarded as outliers because the majority of records in the normal cluster are normal. Therefore, association rules with the lowest support and confidence values in the normal cluster effectively describe the behavior of the attack records. For example, as shown in Table 14 (row 1), if logged_in = 1, flag = SF, and diff_srv_rate = 1, then the label type is attack. This rule has a support and confidence of 64.006% and 7.294%, respectively.

Misuse and Anomaly Detection Using Supervised Learning Phase

Misuse Detection Using Supervised Learning Approach

In this section, several supervised learning algorithms are used to classify normal and attack records within each cluster. This approach consists of three main stages:

(1) Clustering the dataset (as explained in Section 5.2)
(2) Classifying normal and attack records within each cluster and identifying the best classifier
(3) Combining the results from the two clusters to construct the final classification model for distinguishing normal and attack records in the IDS dataset

In this approach, twelve classification algorithms are applied within each cluster to distinguish between normal and attack records in the IDS dataset. These algorithms include C5 decision trees (using gain ratio and information gain criteria), neural networks, logistic regression, naïve Bayes, KNN, random forest, SVM, LDA, AdaBoost, Bagging, and stacking. Many of these methods have been widely utilized in previous IDS studies (Yang et al., 2022).

The classification results are compared, and the best classifier in each cluster is selected based on performance metrics to combine them into a final classification model. In the first cluster, due to the imbalanced data in the target feature, an under-sampling procedure was applied to balance the dataset

Tables 15 and 16 present the classification results for the attack and normal clusters, respectively. Additionally, Figures 4 and 5 illustrate the results for the attack and normal clusters separately:

As shown in Table 15 and Figure 4, the best classification model for the attack cluster is AdaBoost, an ensemble classifier, due to its highest classification accuracy, detection rate, and F-measure, along with a low false alarming rate compared to the other classification algorithms. Furthermore, according to Table 16 and Figure 5, the optimal classification model for normal clusters is stacking, another ensemble technique, because it achieves the highest classification accuracy, detection rate, and F-measure, as well as a low false alarming rate relative to other algorithms.

Table 15: Performance indicators resulting from the implementation of classification algorithms on the attack cluster

Performance index	C5 decision tree (gain ratio)	C5 decision tree (information gain)	Neural network	Logistic regression	Naïve Bayes	KNN
Classification accuracy	98.56%	98.80%	98.08%	95.24%	83.33%	97.62%
Detection rate	96.94%	97.37%	96.94%	94.74%	63.16%	94.74%
Precision	100.00%	100.00%	98.96%	94.74%	100.00%	100.00%
F-measure	98.45%	98.67%	97.94%	94.74%	77.42%	97.30%
False alarming rate	0.00%	0.00%	0.91%	4.35%	0.00%	0.00%
Performance index	Random forest	SVM	LDA	AdaBoost	Bagging	Staking
Classification accuracy	98.56%	95.24%	97.62%	99.40%	98.56%	98.29%
Detection rate	96.94%	94.74%	94.74%	100.00%	96.94%	96.43%
Precision	100.00%	94.74%	100.00%	98.73%	100.00%	100.00%
F-measure	98.45%	94.74%	97.30%	99.36%	98.45%	98.18%
False alarming rate	0.00%	4.35%	0.00%	1.14%	0.00%	0.00%

- Random forest (criterion: information gain)
- AdaBoost (base learner: C5 decision tree with criterion: information gain)
- Bagging (base learner: C5 decision tree with criterion: information gain)
- Stacking (model learner: C5 decision tree with criterion: information gain. Base learners: C5 decision tree with criterion: information gain, decision tree with criterion: gain ratio, and random forest)

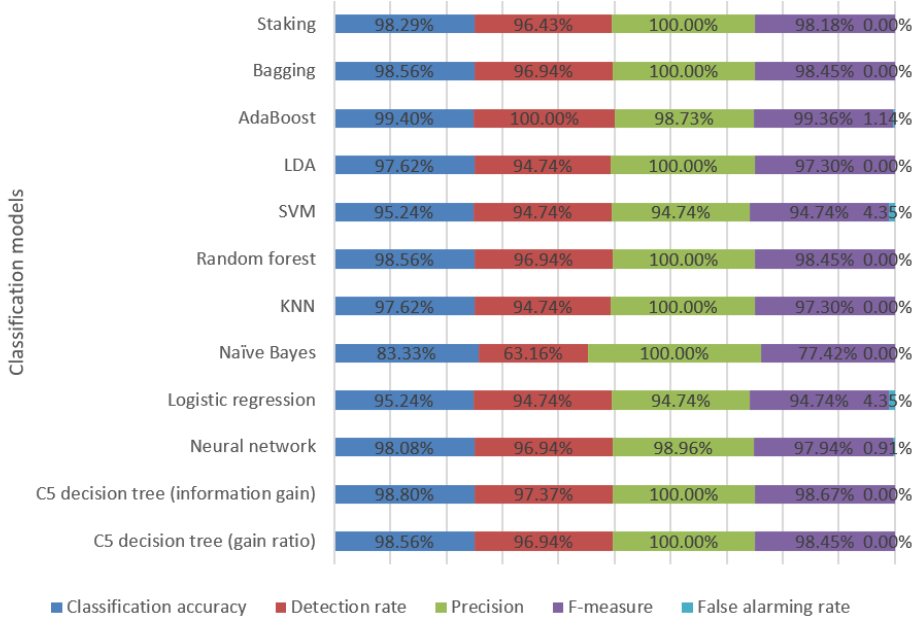

Fig. 4: Comparison of the results from implementing classification algorithms on the attack cluster

Table 16: Performance indicators resulting from the implementation of classification algorithms on the normal cluster

Performance index	C5 decision tree (gain ratio)	C5 decision tree (information gain)	Neural network	Logistic regression	Naïve Bayes	KNN
Classification accuracy	96.68%	97.93%	63.49%	82.99%	63.90%	97.93%
Detection rate	100.00%	97.30%	99.10%	77.48%	100.00%	97.30%
Precision	93.28%	98.18%	55.84%	84.31%	56.06%	98.18%
F-measure	96.52%	97.74%	71.43%	80.75%	71.84%	97.74%
False alarming rate	6.15%	1.54%	66.92%	12.31%	66.92%	1.54%
Performance index	Random forest	SVM	LDA	AdaBoost	Bagging	Staking
Classification accuracy	89.63%	82.40%	66.67%	98.76%	98.34%	99.17%
Detection rate	93.69%	72.05%	96.94%	100.00%	97.30%	100.00%
Precision	85.25%	88.71%	59.04%	97.37%	99.08%	98.23%
F-measure	89.27%	79.52%	73.39%	98.67%	98.18%	99.11%
False alarming rate	13.85%	8.27%	60.63%	2.31%	0.77%	1.54%

- Random forest (criterion: information gain)
- AdaBoost (base learner: C5 decision tree with criterion: information gain)
- Bagging (base learner: C5 decision tree with criterion: information gain)
- Stacking (model learner: C5 decision tree with criterion: information gain. Base learners: C5 decision tree with criterion: information gain, decision tree with criterion: gain ratio, and KNN)

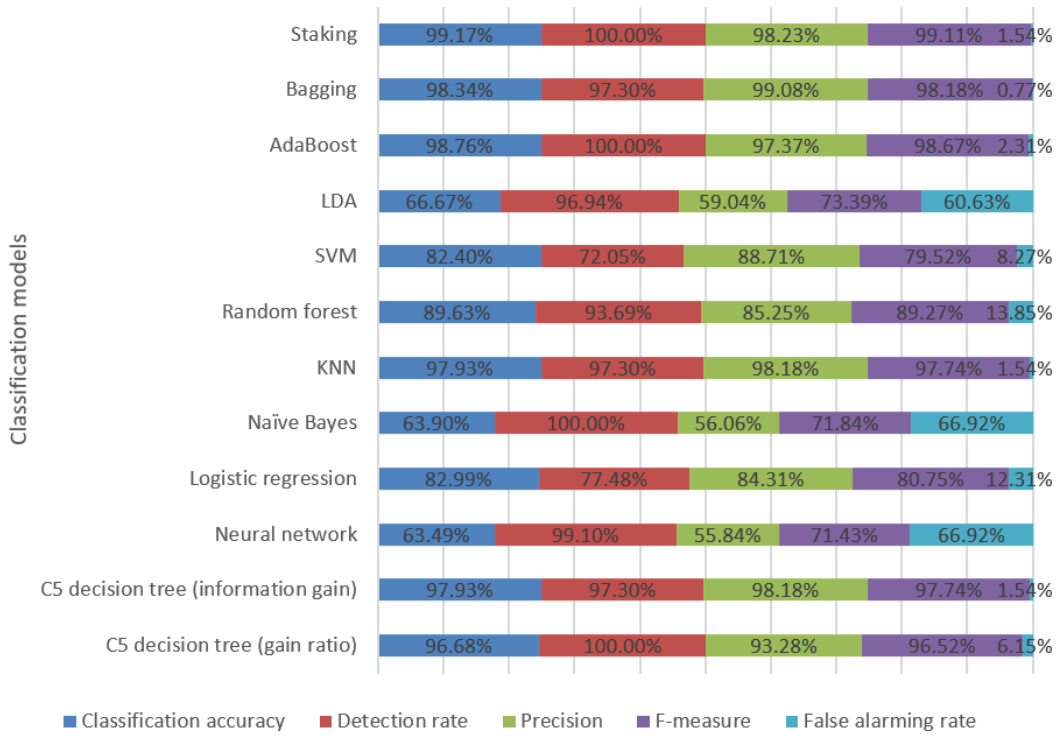

Fig. 5: Comparison of the results from implementing classification algorithms on the normal cluster

After applying the classification algorithms to each cluster, the best classifiers are combined to construct a final classification model. Table 17 and Figure 6 present a comparison of the final classification results for misuse detection using this supervised learning approach against other single and ensemble classifiers across five performance metrics

As shown in Table 17 and Figure 6, the final classification model achieved the highest classification accuracy, detection rate, and F-measure, with values of 99.26%, 100%, and 99.21%, respectively. Additionally, the false alarming rate of

the final classification model was lower (1.38%) compared to most other individual classifiers. Following the final classification model, Bagging and stacking two ensemble classifiers ranked as the best classifiers based on accuracy, precision, and F-measure metrics.

Table 17: Comparison of the proposed final model with other classical and ensemble classifiers

Performance index	Final model	C5 decision tree (gain ratio)	C5 decision tree (information gain)	Neural network	Logistic regression	Naïve Bayes	KNN
Classification accuracy	99.26%	97.37%	98.26%	93.11%	93.28%	84.94%	98.88%
Detection rate	100.00%	97.37%	98.08%	98.91%	97.91%	96.37%	98.54%
Precision	98.44%	97.76%	98.71%	89.48%	90.46%	79.96%	99.39%
F-measure	99.21%	97.57%	98.39%	93.96%	94.04%	87.40%	98.97%
False alarming rate	1.38%	2.64%	1.52%	13.76%	12.21%	28.58%	0.71%
Performance index	Final model	Random forest	SVM	LDA	AdaBoost	Bagging	Staking
Classification accuracy	99.26%	92.25%	93.32%	90.35%	97.96%	98.93%	98.92%
Detection rate	100.00%	93.80%	98.06%	94.51%	97.43%	99.11%	98.74%
Precision	98.44%	92.06%	90.43%	88.47%	98.78%	98.92%	99.25%
F-measure	99.21%	92.92%	94.09%	91.39%	98.10%	99.01%	99.00%
False alarming rate	1.38%	9.57%	12.28%	14.58%	1.42%	1.29%	0.88%

- Random forest (criterion: information gain)
- AdaBoost (base learner: C5 decision tree with criterion: information gain)
- Bagging (base learner: C5 decision tree with criterion: information gain)
- Stacking (model learner: C5 decision tree with criterion: information gain. Base learners: C5 decision tree with criterion: information gain, decision tree with criterion: gain ratio, and KNN

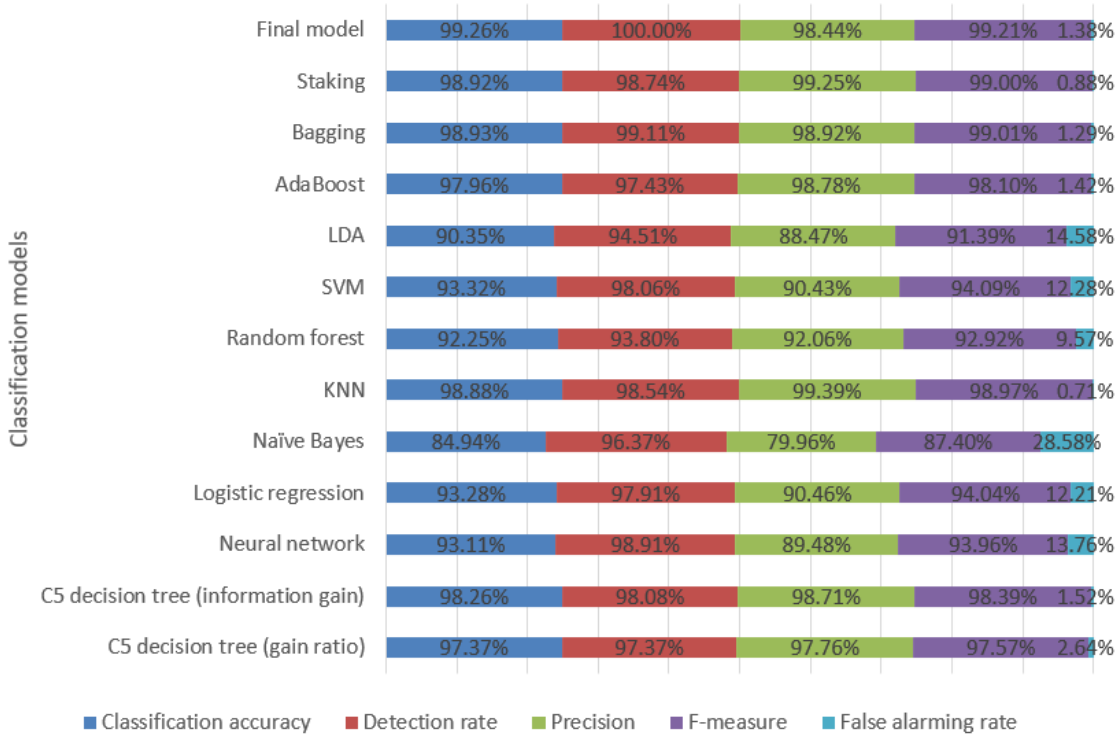

Fig. 6: Comparison of the proposed final model with other classifiers

In conclusion, for misuse detection using a supervised learning approach, the final classification model constructed by combining AdaBoost and stacking algorithms proves to be the most effective classifier for IDS. Notably, this model is built using two ensemble learning classifiers, which underscores the high accuracy of ensemble methods in distinguishing between normal and attack records in an IDS. The final classification model demonstrates superior performance in detection rate compared to the model proposed by Cavusoglu (2019). applied several types of SVM for IDS; however, their best accuracy was 98.7%, which is lower than that of the final model in this study. Gao et al. (2019) designed an adaptive ensemble machine learning model using classifiers such as decision trees, KNN, and deep neural networks.

The accuracy of their proposed model was 84.23%, which is significantly lower than that of the current study. Naseri and Gharehchopogh (2022) presented a hybrid model combining feature selection algorithms with KNN, SVM, decision tree, naïve Bayes, random forest, and AdaBoost classifiers; however, their model achieved only 90% classification accuracy.

Bakro et al. (2023) applied a hybrid approach involving feature selection, ensemble methods, and deep learning to develop their IDS model. The accuracy of their model on the NSL-KDD dataset was 99.01%.5.3.2. Anomaly detection using supervised learning approach.

This section presents the use of supervised learning methods to detect anomalies in the IDS dataset. The approach consists of three main stages. First, the dataset is clustered into two groups, as explained in Section 5.2, with the results shown in Tables 8–10. Second, an outlier detection technique (described in Section 5.2.2) is applied to identify outliers within each cluster. In the normal cluster, the goal is to detect attack outliers, and vice versa. Third, supervised learning (classification) techniques are used to classify the outliers in each cluster. Specifically, normal outliers are classified within the attack cluster, and attack outliers are classified within the normal cluster. Using this approach, the false alarming rate can be reduced because outlier behavior can be predicted with greater accuracy. Importantly, within each cluster, an under-sampling strategy was employed to mitigate the effects of imbalanced data by reducing the number of records associated with one of the target variable's values. Specifically, in the normal cluster, the number of normal records was reduced to three times the number of attack records, and vice versa. Based on this methodology, this study represents the first attempt to apply supervised learning techniques to enhance the capabilities of anomaly detection models.

Figure 7 and Table 18 illustrate the performance of the single and ensemble classifiers deployed in the attack cluster for classifying normal outliers. Additionally, Table 19 and Figure 8 present the performance of the classification algorithms used in the normal cluster to identify attack outliers.

As shown in Table 18 and Figure 7, the best classification models for identifying normal outliers within the attack cluster based on five performance metrics are the neural network, logistic regression, SVM, and AdaBoost. These classifiers effectively detect and predict normal outliers in the attack cluster. The values for the five-performance metrics accuracy, detection rate, precision, F-measure, and false alarming rate are identical for these classifiers, recorded as 99.43%, 100%, 90.91%, 95.24%, and 0.6%, respectively.

In addition, as shown in Table 19 and Figure 8, Bagging and AdaBoost are the most effective classifiers for identifying attack outliers within the normal cluster. Bagging outperforms AdaBoost in terms of classification accuracy, detection rate, precision, and F-measure, with values of 96%, 97.30%, 97.30%, and 97.30%, respectively. However, AdaBoost surpasses Bagging regarding the false alarming rate, which is 7.69%.

Table 18: Performance indicators resulting from the implementation of classification algorithms on the attack cluster

Performance index	C5 decision tree (gain ratio)	C5 decision tree (information gain)	Neural network	Logistic regression	Naïve Bayes	KNN
Classification accuracy	97.73%	98.30%	99.43%	99.43%	66.48%	97.73%
Detection rate	70.00%	80.00%	100.00%	100.00%	100.00%	70.00%
Precision	87.50%	88.89%	90.91%	90.91%	14.49%	87.50%
F-measure	77.78%	84.21%	95.24%	95.24%	25.32%	77.78%
False alarming rate	0.60%	0.60%	0.60%	0.60%	35.54%	0.60%
Performance index	Random forest	SVM	LDA	AdaBoost	Bagging	Staking
Classification accuracy	96.59%	99.43%	98.30%	99.43%	98.86%	98.86%
Detection rate	40.00%	100.00%	100.00%	100.00%	80.00%	80.00%
Precision	100.00%	90.91%	76.92%	90.91%	100.00%	100.00%
F-measure	57.14%	95.24%	86.96%	95.24%	88.89%	88.89%
False alarming rate	0.00%	0.60%	1.81%	0.60%	0.00%	0.00%

- Random forest (criterion: information gain)
- AdaBoost (base learner: logistic regression)
- Bagging (base learner: SVM)
- Stacking (model learner: C5 decision tree with criterion: information gain. Base learners: neural network, SVM, and logistic regression)

Table 19: Performance indicators resulting from the implementation of classification algorithms on the normal cluster

Performance index	C5 decision tree (gain ratio)	C5 decision tree (information gain)	Neural network	Logistic regression	Naïve Bayes	KNN
Classification accuracy	75.00%	83.33%	83.33%	58.33%	66.67%	83.33%
Detection rate	77.78%	77.78%	100.00%	44.44%	55.56%	77.78%
Precision	87.50%	100.00%	81.82%	100.00%	100.00%	100.00%
F-measure	82.35%	87.50%	90.00%	61.54%	71.43%	87.50%
False alarming rate	33.33%	0.00%	66.67%	0.00%	0.00%	0.00%
Performance index	Random forest	SVM	LDA	AdaBoost	Bagging	Staking
Classification accuracy	75.00%	75.00%	25.00%	94.59%	96.00%	86.49%
Detection rate	88.89%	100.00%	0.00%	93.10%	97.30%	82.76%
Precision	80.00%	75.00%	50.00%	100.00%	97.30%	100.00%
F-measure	84.21%	85.71%	16.67%	96.43%	97.30%	90.57%
False alarming rate	66.67%	100.00%	0.00%	0.00%	7.69%	0.00%

- Random forest (criterion: gain ratio)
- AdaBoost (base learner: C5 decision tree with criterion: information gain)
- Bagging (base learner: C5 decision tree with criterion: information gain)
- Stacking (model learner: C5 decision tree with criterion: information gain. Base learners: neural network, decision tree, KNN)

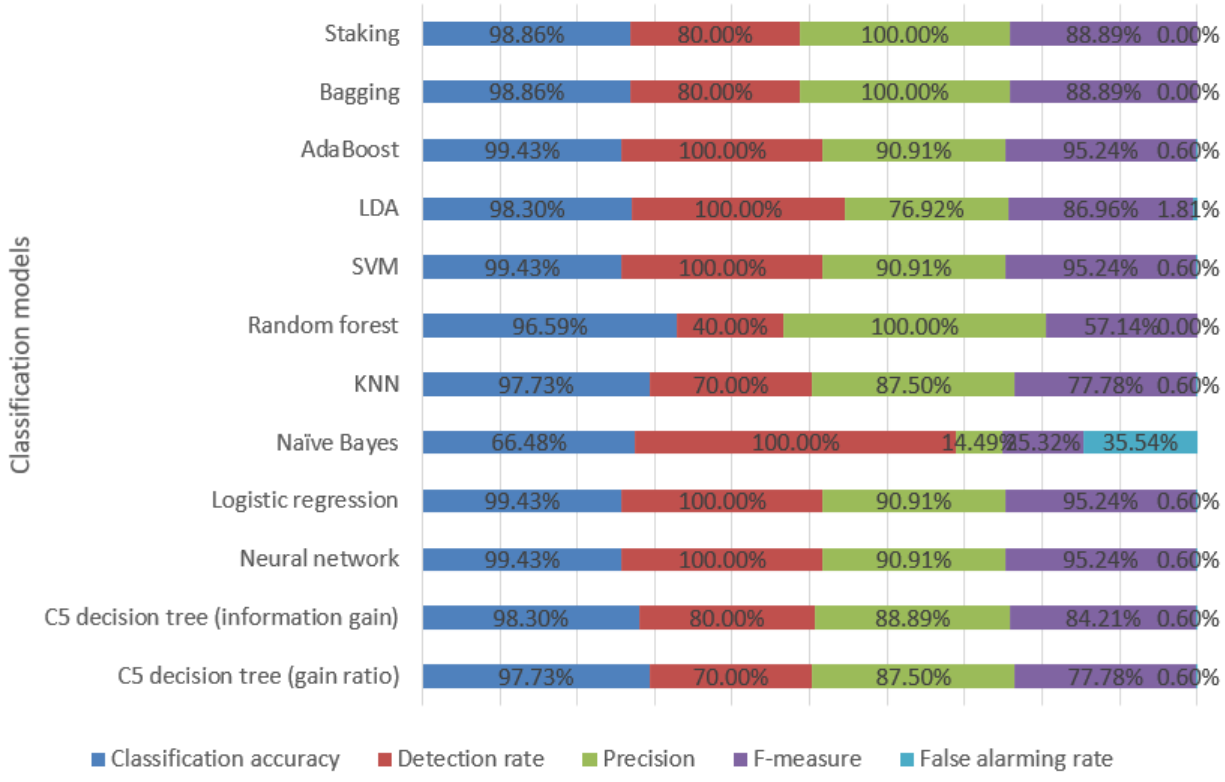

Fig. 7: Comparison of the performance of outlier classification models on the attack cluster

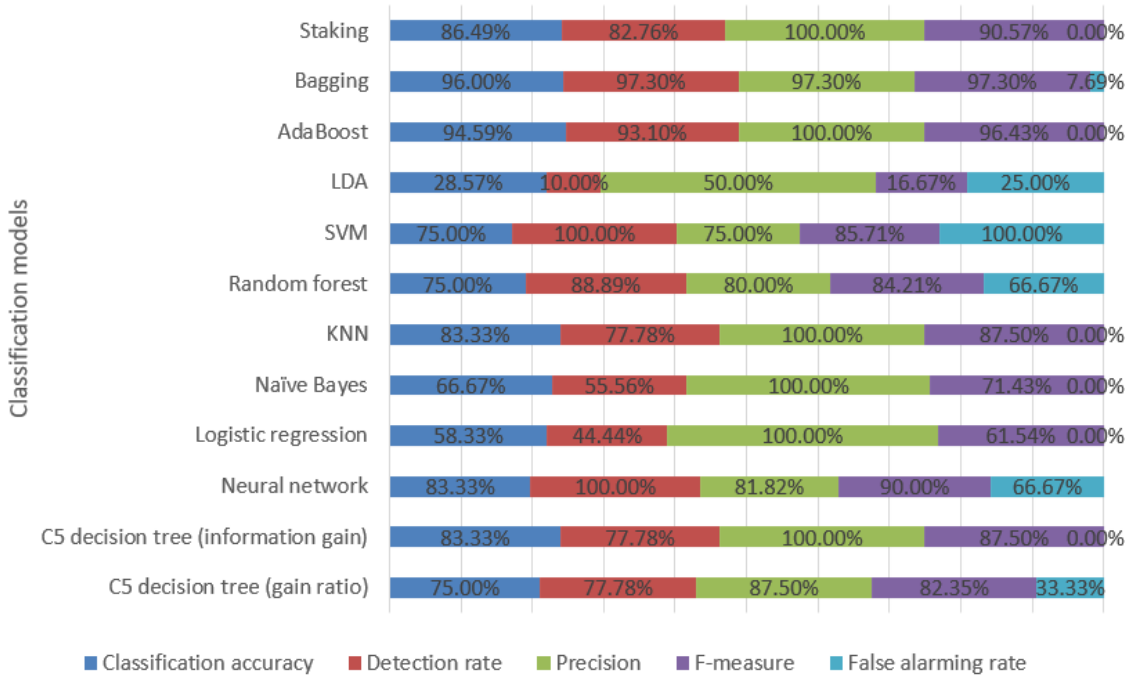

Fig. 8: Comparison of the performance of outlier classification models on the normal cluster

In this section, we present a novel hybrid machine learning model designed to classify normal and attack outliers. The model integrates clustering, outlier detection, and multiple single and ensemble classifiers to accurately identify and predict normal and attack outliers within their respective clusters. To our knowledge, this is the first study to combine these four machine learning techniques into a unified model for outlier classification in IDS.

Regarding the results, the following comparison can be made. This paper introduces a novel compound model that integrates misuse and anomaly-based IDS on one side, and hybrid supervised, unsupervised, and outlier detection data mining and machine learning approaches on the other. Previous studies have applied misuse and anomaly detection methods individually (see Sections 1 and 2). Additionally, some research has employed supervised, unsupervised, and outlier detection techniques separately within IDS (see Sections 1 and 2). However, these efforts have faced challenges such as reduced classification accuracy and increased false alarming rates. Furthermore, several other studies have investigated hybrid approaches to IDS, but they have not fully leveraged the benefits of integrating supervised, unsupervised, and outlier detection methods within a comprehensive hybrid framework. For example, Kaja et al. (2019) applied four classifiers-C4.5 decision tree, random forest, naïve Bayes, and AdaBoost to each cluster of the dataset to detect anomalies. The current study distinguishes itself by employing association rules for both misuse and anomaly detection, unlike previous research. While prior studies primarily relied on accuracy for evaluation, our combined model improves multiple performance metrics. This approach not only enhances accuracy but also introduces more sophisticated methods for integrating supervised, unsupervised, and outlier detection techniques in misuse and anomaly-based IDS. Furthermore, our model effectively reduces the number of IDS-related features through a feature selection algorithm, thereby improving both effectiveness and efficiency.

In conclusion, this study makes significant contributions through four key approaches.

This study presents a composite model for misuse and anomaly detection by employing hybrid supervised and unsupervised learning approaches in data mining. In the unsupervised learning phase, after applying a feature selection algorithm and other preprocessing techniques, two clustering algorithms-Kohonen and K-means were used to segment the dataset into attack and normal clusters. The study introduces a novel procedure to identify the most suitable clustering algorithm and determine the optimal number of clusters. Subsequently, association rule mining was employed to classify attack and normal rules within the respective clusters. This work represents the first attempt to integrate hybrid feature selection, clustering, and association rule mining algorithms for misuse detection in the IDS domain.

Building on the clustering approach, the study employs an outlier detection algorithm for each cluster to identify attack and normal outliers within the normal and attack clusters, respectively. This unique methodology is then combined with the association rule technique to characterize the behavior of both attack and normal outliers within their corresponding clusters. This novel procedure presents a comprehensive unsupervised model that integrates feature selection, clustering, outlier detection, and association rule algorithms, thereby making a significant contribution to the field of anomaly detection.

As the third contribution, the study employs 12 algorithms, including both classical and ensemble classifiers, to classify attack and normal records within each cluster. The optimal classifier for each cluster is selected based on five performance metrics. Subsequently, a final classifier is developed by combining the two best classifiers associated with each cluster. The results demonstrate the superiority of this combined classifier over other alternatives. This contribution integrates feature selection, clustering, and both classical and ensemble learning classifiers, resulting in a robust misuse detection approach for IDS.

Finally, this study extends its contributions by applying twelve classical and ensemble algorithms to classify attack and normal outliers within the respective normal and attack clusters. This pioneering effort integrates multiple machine learning techniques including feature selection, clustering, classical and ensemble classification, and outlier detection to develop an anomaly detection approach for IDS. Notably, outlier detection was employed to identify attack and normal anomalies within their corresponding clusters. Subsequently, classical and ensemble classifiers were used to categorize attack and normal records. The findings highlight the superior performance of ensemble learning classifiers in most cases compared to classical classifiers.

Conclusion and Future Suggestions

Recently, numerous studies have employed data mining and machine learning techniques to enhance misuse and anomaly intrusion detection. Some have demonstrated the effectiveness of hybrid methods that combine different algorithms to improve Intrusion Detection Systems (IDSs). In this context, our study introduces a novel composite model for misuse and anomaly IDSs, integrating supervised, unsupervised, and outlier detection approaches within data mining techniques. This innovative fusion is expected to significantly improve IDS performance, aligning with the ongoing trend of strengthening security through advanced data analysis methods. The study was conducted in three phases:

1. Data preparation and preprocessing
2. Misuse and anomaly detection using unsupervised learning
3. Misuse and anomaly detection using supervised learning

After preprocessing and identifying the appropriate features, the second phase focused on misuse detection using unsupervised learning through cluster profiling. Subsequently, the association rule mining algorithm was applied to classify normal and attack behaviors within clusters, aiming to improve classification accuracy and detection rates. For anomaly detection using unsupervised learning, outlier detection techniques were applied to each cluster, followed by the use of association rule mining to characterize normal and attack outlier behaviors within clusters, thereby reducing the false alarming rate. This study's second phase contributes by integrating feature selection, clustering (including K-means and Kohonen algorithms), association rule mining, and outlier detection for both misuse and anomaly detection in IDS.

In the third phase, misuse detection was performed using supervised learning with 12 classical and ensemble classifiers to classify normal and attack records within each cluster. After combining the results into two clusters, the final classifier was identified as the best compared to the others, without employing feature selection or clustering algorithms. For anomaly detection using supervised learning, these 12 classifiers were applied to classify normal and attack outliers within two clusters to reduce the false alarming rate. This phase contributes to the study by integrating feature selection, clustering methods (including k-means and Kohonen algorithms), and both classical and ensemble classification algorithms for misuse and anomaly detection in IDS.

In conclusion, it is essential to emphasize that the primary objective of the proposed novel compound model is to improve performance metrics such as classification accuracy, detection rate, precision, F-measure, and false alarming rate. This improvement is achieved by simultaneously leveraging data mining and machine learning techniques, thereby addressing the limitations of standalone misuse and anomaly intrusion detection methods. Notably, this study represents the first effort to integrate misuse and anomaly detection within an IDS while concurrently combining unsupervised and supervised learning approaches in the data mining domain. Ultimately, this innovative compound model has the potential to significantly advance both misuse and anomaly detection by harnessing the strengths of data mining techniques within the IDS field.

This study has several limitations. First, to evaluate the proposed model, a sampling procedure was used to select a subset of data. Different data samples may produce varying results from the machine learning and data mining techniques applied in this study. However, the proposed

model can be adapted for each data sample. Second, this study employed only one feature selection algorithm to identify an appropriate number of features; using alternative feature selection methods could yield different outcomes. Third, an under-sampling technique was applied to address the issue of imbalanced data. Other data balancing methods might result in different outputs for the proposed model. Fourth, the model utilized a limited set of clustering, association rule mining, and classification techniques for IDS. Numerous other machine learning methods could be incorporated, but their use was beyond the scope of this study.

Future research could explore integrating various well-established feature selection, clustering, outlier detection, classification, and association rule mining algorithms to further enhance the proposed model. Additionally, a wide range of feature selection techniques based on filter and wrapper approaches can be employed in future studies to improve the model. Beyond feature selection, numerous feature extraction algorithms, such as PCA, can be utilized to reduce data dimensionality. Furthermore, subsequent researchers might apply alternative data balancing methods, such as oversampling and SMOTE, to address dataset imbalance. The incorporation of deep learning algorithms for classification, alongside classical and ensemble methods, also presents a promising avenue. Finally, considering the applicability of the proposed model within the emerging IoT domain is worthwhile. Given the rapid expansion of IoT, investigating the model's potential impact and effectiveness in securing IoT systems could provide valuable insights.

Acknowledgment

Thank you to the publisher for their support in the publication of this research article. We are grateful for the resources and platform provided by the publisher, which have enabled us to share our findings with a wider audience. We appreciate the efforts of the editorial team in reviewing and editing our work, and we are thankful for the opportunity to contribute to the field of research through this publication.

Funding Information

No funding was received to support the preparation of this manuscript.

The authors declare that they have no financial or non-financial conflicts of interest.

Authors Contributions

All authors contributed to the study's conception and design, data analysis, and manuscript writing. Additionally, all authors reviewed earlier versions of the manuscript, provided feedback, and approved the final version.

Ethics

This study utilizes a publicly available dataset commonly used by researchers in the field of intrusion detection systems. A detailed description of the dataset is provided in Section 4.1. Since the data is publicly accessible and does not involve human subjects, there are no ethical concerns related to its use. Furthermore, the authors confirm that this work represents original research that has not been previously submitted or published. All authors have reviewed the manuscript and approved its final version.

Availability of Data and Materials

The study utilizes the standard NSL-KDD dataset.

Competing of Interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

Abdulganiyu, O. H., Ait Tchakoucht, T., & Saheed, Y. K. (2023). A systematic literature review for network intrusion detection system (IDS). *International Journal of Information Security*, *22*(5), 1125–1162. https://doi.org/10.1007/s10207-023-00682-2

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). *Network intrusion detection system: A systematic study of machine learning and deep learning approaches*. Transactions on Emerging Telecommunications Technologies.https://doi.org/10.1002/ett.4150

Alamleh, A., Albahri, O. S., Zaidan, A. A., Alamoodi, A. H., Albahri, A. S., Zaidan, B. B., Qahtan, S., binti Ismail, A. R., Malik, R. Q., Baqer, M. J., Jasim, A. N., & Al-Samarraay, M. S. (2023). Multi-Attribute Decision-Making for Intrusion Detection Systems: A Systematic Review. *International Journal of Information Technology & Decision Making*, *22*(01), 589–636. https://doi.org/10.1142/s021962202230004x

Aldallal, A. (2022). Toward Efficient Intrusion Detection System Using Hybrid Deep Learning Approach. *Symmetry*, *14*(9), 1916. https://doi.org/10.3390/sym14091916

Aldweesh, A., Derhab, A., & Emam, A. Z. (2020). Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. In *Knowledge-Based System* (Vol. 189, p. 105124). https://doi.org/10.1016/j.knosys.2019.105124

Alghamdi, R., & Bellaiche, M. (2023). An ensemble deep learning-based IDS for IoT using Lambda architecture. *Cybersecurity*, *6*(1), 5. https://doi.org/10.1186/s42400-022-00133-w

Ali, R., Ali, A., Iqbal, F., Hussain, M., & Ullah, F. (2022). Deep Learning Methods for Malware and Intrusion Detection: A Systematic Literature Review. *Security and Communication Networks*, *2022*, 1–31. https://doi.org/10.1155/2022/2959222

Alkasassbeh, M., & Al-Haj Baddar, S. (2023). Intrusion Detection Systems: A State-of-the-Art Taxonomy and Survey. *Arabian Journal for Science and Engineering*, *48*(8), 10021–10064. https://doi.org/10.1007/s13369-022-07412-1

Amarudin, Ferdiana, R., & Widyawan. (2020). A Systematic Literature Review of Intrusion Detection System for Network Security: Research Trends, Datasets and Methods. *2020 4th International Conference on Informatics and Computational Sciences (ICICoS)*, 1–6. https://doi.org/10.1109/icicos51170.2020.9299068

Bakro, M., Kumar, R. R., Alabrah, A. A., Ashraf, Z., Bisoy, S. K., Parveen, N., Khawatmi, S., & Abdelsalam, A. (2023). Efficient Intrusion Detection System in the Cloud Using Fusion Feature Selection Approaches and an Ensemble Classifier. *Electronics*, *12*(11), 2427. https://doi.org/10.3390/electronics12112427

Bhati, N. S., & Khari, M. (2021). A Survey on Hybrid Intrusion Detection Techniques. *Research in Intelligent and Computing in Engineering*, *1254*, 815–825. https://doi.org/10.1007/978-981-15-7527-3_77

Cavusoglu, Ü. (2019). A new hybrid approach for intrusion detection using machine learning methods. *Applied Intelligence*, *49*(7), 2735–2761. https://doi.org/10.1007/s10489-018-01408-x

Choras, M., & Pawlicki, M. (2021). Intrusion detection approach based on optimised artificial neural network. *Neurocomputing*, *452*, 705–715. https://doi.org/10.1016/j.neucom.2020.07.138

Diaz-Garcia, J. A., Ruiz, M. D., & Martin-Bautista, M. J. (2023). A survey on the use of association rules mining techniques in textual social media. *Artificial Intelligence Review*, *56*(2), 1175–1200. https://doi.org/10.1007/s10462-022-10196-3

Drewek-Ossowicka, A., Pietrołaj, M., & Rumiński, J. (2021). A survey of neural networks usage for intrusion detection systems. *Journal of Ambient Intelligence and Humanized Computing*, *12*(1), 497–514. https://doi.org/10.1007/s12652-020-02014-x

Gao, X., Shan, C., Hu, C., Niu, Z., & Liu, Z. (2019). An Adaptive Ensemble Machine Learning Model for Intrusion Detection. *IEEE Access*, *7*, 82512–82521. https://doi.org/10.1109/access.2019.2923640

Geetha, S., Dulhare, U. N., & Sivatha Sindhu, S. S. (2018). Intrusion Detection using NBHoeffding Rule based Decision Tree for Wireless Sensor Networks. *2018 Second International Conference on Advances in Electronics, Computers and Communications (ICAECC)*, 1–5. https://doi.org/10.1109/icaecc.2018.8479483

Guezzaz, A., Azrour, M., Benkirane, S., Mohy-Eddine, M., Attou, H., & Douiba, M. (2022). A Lightweight Hybrid Intrusion Detection Framework using Machine Learning for Edge-Based IIoT Security. *The International Arab Journal of Information Technology*, *19*(5), 822–830. https://doi.org/10.34028/iajit/19/5/14

Hajisalem, V., & Babaie, S. (2018). A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection. *Computer Networks*, *136*, 37–50. https://doi.org/10.1016/j.comnet.2018.02.028

Han, J., Kamber, M., & Pei, J. (2012). Outlier Detection. *Data Mining: Concepts and Techniques (Third Edition)*, 543–584. https://doi.org/10.1016/B978-0-12-381479-1.00012-5

He, K., Kim, D. D., & Asghar, M. R. (2024). NIDS-Vis: Improving the generalized adversarial robustness of network intrusion detection system. *Computers & Security*, *145*, 104028. https://doi.org/10.1016/j.cose.2024.104028

Hossain, Md. A., & Islam, Md. S. (2023). Ensuring network security with a robust intrusion detection system using ensemble-based machine learning. *Array*, *19*, 100306. https://doi.org/10.1016/j.array.2023.100306

Kaja, N., Shaout, A., & Ma, D. (2019). An intelligent intrusion detection system. *Applied Intelligence*, *49*(9), 3235–3247. https://doi.org/10.1007/s10489-019-01436-1

Kasongo, S. M., & Sun, Y. (2020). Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset. *Journal of Big Data*, *7*(1), 105. https://doi.org/10.1186/s40537-020-00379-6

Kaur, S., & Singh, M. (2020). Hybrid intrusion detection and signature generation using Deep Recurrent Neural Networks. *Neural Computing and Applications*, *32*(12), 7859–7877. https://doi.org/10.1007/s00521-019-04187-9

Kayode Saheed, Y., Idris Abiodun, A., Misra, S., Kristiansen Holone, M., & Colomo-Palacios, R. (2022). A machine learning-based intrusion detection for detecting internet of things network attacks. *Alexandria Engineering Journal*, *61*(12), 9395–9409. https://doi.org/10.1016/j.aej.2022.02.063

Khanbabaei, M., Parsi, P., & Farhadi, N. (2023). Using data mining to integrate recency-frequency-monetary value analysis and credit scoring methods for bank customer behaviour analysis. *International Journal of Data Mining, Modelling and Management*, *15*(4), 369–392. https://doi.org/10.1504/ijdmmm.2023.134598

Khanbabaei, M., Sobhani, F. M., Alborzi, M., & Radfar, R. (2018). Developing an integrated framework for using data mining techniques and ontology concepts for process improvement. *Journal of Systems and Software*, *137*, 78–95. https://doi.org/10.1016/j.jss.2017.11.019

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges. *Cybersecurity*, *2*(1), 20. https://doi.org/10.1186/s42400-019-0038-7

Kilincer, I. F., Ertam, F., & Sengur, A. (2021). Machine learning methods for cyber security intrusion detection: Datasets and comparative study. *Computer Networks*, *188*, 107840. https://doi.org/10.1016/j.comnet.2021.107840

Korium, M. S., Saber, M., Beattie, A., Narayanan, A., Sahoo, S., & Nardelli, P. H. J. (2024). Intrusion detection system for cyberattacks in the Internet of Vehicles environment. *Ad Hoc Networks*, *153*, 103330. https://doi.org/10.1016/j.adhoc.2023.103330

Koutanaei, F. N., Sajedi, H., & Khanbabaei, M. (2015). A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring. *Journal of Retailing and Consumer Services*, *27*, 11–23. https://doi.org/10.1016/j.jretconser.2015.07.003

Li, J., Othman, M. S., Chen, H., & Yusuf, L. M. (2024). Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning. *Journal of Big Data*, *11*(1), 36. https://doi.org/10.1186/s40537-024-00892-y

Maseer, Z. K., Yusof, R., Bahaman, N., Mostafa, S. A., & Foozy, C. F. M. (2021). Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset. *IEEE Access*, *9*, 22351–22370. https://doi.org/10.1109/access.2021.3056614

Meryem, A., & Ouahidi, B. E. (2020). Hybrid intrusion detection system using machine learning. *Network Security*, *2020*(5), 8–19. https://doi.org/10.1016/s1353-4858(20)30056-8

More, S., Idrissi, M., Mahmoud, H., & Asyhari, A. T. (2024). Enhanced Intrusion Detection Systems Performance with UNSW-NB15 Data Analysis. *Algorithms*, *17*(2), 64. https://doi.org/10.3390/a17020064

Naseri, T. S., & Gharehchopogh, F. S. (2022). A Feature Selection Based on the Farmland Fertility Algorithm for Improved Intrusion Detection Systems. *Journal of Network and Systems Management*, *30*(3), 40. https://doi.org/10.1007/s10922-022-09653-9

Nasir, M. H., Khan, S. A., Khan, M. M., & Fatima, M. (2022). Swarm Intelligence inspired Intrusion Detection Systems A systematic literature review. *Computer Networks*, *205*, 108708. https://doi.org/10.1016/j.comnet.2021.108708

Ozkan-Okay, M., Samet, R., Aslan, O., & Gupta, D. (2021). A Comprehensive Systematic Literature Review on Intrusion Detection Systems. *IEEE Access*, *9*, 157727–157760. https://doi.org/10.1109/access.2021.3129336

Salo, F., Injadat, M., Nassif, A. B., Shami, A., & Essex, A. (2018). Data Mining Techniques in Intrusion Detection Systems: A Systematic Literature Review. *IEEE Access*, *6*, 56046–56058. https://doi.org/10.1109/access.2018.2872784

Saranya, T., Sridevi, S., Deisy, C., Chung, T. D., & Khan, M. K. A. A. (2020). Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review. *Procedia Computer Science*, *171*, 1251–1260. https://doi.org/10.1016/j.procs.2020.04.133

Sikder, M. N. K., & Batarseh, F. A. (2023). Outlier detection using AI: a survey. *AI Assurance*, 231–291. https://doi.org/10.1016/b978-0-32-391919-7.00020-2

Sivagaminathan, V., Sharma, M., & Henge, S. K. (2023). Intrusion detection systems for wireless sensor networks using computational intelligence techniques. *Cybersecurity*, *6*(1), 1–15. https://doi.org/10.1186/s42400-023-00161-0

Talukder, M. A., Hasan, K. F., Islam, M. M., Uddin, M. A., Akhter, A., Yousuf, M. A., Alharbi, F., & Moni, M. A. (2023). A dependable hybrid machine learning model for network intrusion detection. *Journal of Information Security and Applications*, *72*, 103405. https://doi.org/10.1016/j.jisa.2022.103405

Talukder, M. A., Khalid, M., & Sultana, N. (2025). A hybrid machine learning model for intrusion detection in wireless sensor networks leveraging data balancing and dimensionality reduction. *Scientific Reports*, *15*(1), 4617. https://doi.org/10.1038/s41598-025-87028-1

Tsui, K.-L., Chen, V., Jiang, W., Yang, F., & Kan, C. (2023). Data Mining Methods and Applications. *Springer Handbook of Engineering Statistics*, 797–816. https://doi.org/10.1007/978-1-4471-7503-2_38

Vivek, V., & Veeravalli, B. (2025). ARMBoost+: Empowering stacking, ensemble, and boosting models for network intrusion detection with dynamic rule repository. *Journal of Network and Computer Applications*, *243*, 104292. https://doi.org/10.1016/j.jnca.2025.104292

Weber, F. (2023). Case Studies on the Use of AI-Based Business Analytics. *Artificial Intelligence for Business Analytics*, 113–136. https://doi.org/10.1007/978-3-658-37599-7_4

Xu, J., Qu, K., Sun, Y., & Yang, J. (2023). Feature selection using self-information uncertainty measures in neighborhood information systems. *Applied Intelligence*,*53*(4), 4524–4540. https://doi.org/10.1007/s10489-022-03760-5

Yang, Z., Liu, X., Li, T., Wu, D., Wang, J., Zhao, Y., & Han, H. (2022). A systematic literature review of methods and datasets for anomaly-based network intrusion detection. *Computers & Security*, *116*, 102675. https://doi.org/10.1016/j.cose.2022.102675

Zhou, Y., Mazzuchi, T. A., & Sarkani, S. (2020). M-AdaBoost-A based ensemble system for network intrusion detection. *Expert Systems with Applications*, *162*, 113864. https://doi.org/10.1016/j.eswa.2020.113864