A Novel Framework for Detecting Network Intrusions Based on Machine Learning Methods

Batyrkhan Omarov, Nazgul Abdinurova, Zhamshidbek Abdulkhamidov
Suleyman Demirel University, Kaskelen, Kazakhstan

***Abstract*—In the rapidly evolving landscape of cyber threats, the efficacy of traditional rule-based network intrusion detection systems has become increasingly questionable. This paper introduces a novel framework for identifying network intrusions, leveraging the power of advanced machine learning techniques. The proposed methodology steps away from the rigidity of conventional systems, bringing a flexible, adaptive, and intuitive approach to the forefront of network security. This study employs a diverse blend of machine learning models including but not limited to, Convolutional Neural Networks (CNNs), Support Vector Machines (SVMs), and Random Forests. This research explores an innovative feature extraction and selection technique that enables the model to focus on high-priority potential threats, minimizing noise and improving detection accuracy. The framework's performance has been rigorously evaluated through a series of experiments on benchmark datasets. The results consistently surpass traditional methods, demonstrating a remarkable increase in detection rates and a significant reduction in false positives. Further, the machine learning-based model demonstrated its ability to adapt to new threat landscapes, indicating its suitability in real-world scenarios. By marrying the agility of machine learning with the concreteness of network intrusion detection, this research opens up new avenues for dynamic and resilient cybersecurity. The framework offers an innovative solution that can identify, learn, and adapt to evolving network intrusions, shaping the future of cyber defense strategies.**

***Keywords*—Attack detection; intrusion detection; machine learning; information security; artificial intelligence**

I. Introduction

The omnipresence of network systems and the growing dependence of various industries on these platforms have amplified the necessity for robust cybersecurity measures. A significant component of such measures is the effective detection of network intrusions [1]. Traditionally, the detection of such intrusions has been performed by rule-based systems, which, while effective in certain circumstances, have been found lacking in the face of more complex and evolving cyber threats [2].

In the field of cybersecurity, intrusion detection systems (IDS) play a pivotal role. IDS work as a security guard, continuously monitoring network traffic and promptly identifying possible threats [3]. The most commonly employed IDS are signature-based and anomaly-based. Signature-based IDS detect known threats by matching them against an existing database of threat signatures. Conversely, anomaly-based IDS identify deviations from the 'normal' network behavior as potential threats. While these traditional methods provide a certain degree of security, their shortcomings are becoming more apparent in the contemporary landscape of cyber threats [4].

Signature-based IDS are inherently limited by their dependence on the existing database of threats. As they can only detect previously encountered threats, their effectiveness dwindles when faced with novel, unknown threats [5]. On the other hand, anomaly-based IDS, while theoretically capable of detecting new threats, often suffer from high false-positive rates due to the challenge of defining what constitutes 'normal' behavior.

In recent years, the rise of machine learning has provided a promising avenue for overcoming these limitations [6-8]. Machine learning, with its ability to learn from data and make decisions, has shown significant potential in numerous fields, including cybersecurity [9]. Particularly, machine learning methods can address the limitations of traditional IDS by learning from past data, improving over time, and adaptively identifying new threats.

This paper introduces a novel framework for detecting network intrusions, harnessing the power of machine learning. This framework moves beyond the rule-based systems, offering a more flexible, adaptive, and intuitive approach to network security.

The main impetus for this research comes from the increasing complexity of cyber threats and the consequent need for more advanced detection techniques. The landscape of network intrusions has seen an evolution from relatively straightforward threats to sophisticated attacks that can bypass conventional security measures.

Therefore, the primary objective of this research is to develop a machine learning-based framework for network intrusion detection that can effectively identify and respond to both known and unknown threats. We aim to leverage the predictive and adaptive capabilities of machine learning to achieve higher detection accuracy and lower false-positive rates than traditional IDS.

This research also seeks to address the issue of scalability in network intrusion detection. As network systems grow in size and complexity, the amount of network data that needs to be monitored also increases, posing a significant challenge to conventional IDS. Our machine learning-based framework is designed to handle this increased scale and complexity effectively.

The world of cybersecurity is at a juncture where traditional techniques of network intrusion detection are proving insufficient against the complex and evolving landscape of cyber threats. To address this, our research explores the integration of machine learning techniques into a novel framework for detecting network intrusions. This research aims not only to enhance the efficacy of intrusion detection but also to pave the way for future advancements in cybersecurity.

II. Literature Review

The literature review presented herein gives a comprehensive overview of various machine learning methods applied in the realm of network intrusion detection systems (IDS). These methodologies encompass both the traditional algorithms and the emerging paradigms.

A. Traditional Machine Learning Methods in Intrusion Detection

Decision Trees (DT) have been widely applied in IDS due to their interpretability and efficiency in handling large-scale datasets. DT-based models, such as the C4.5 algorithm, have shown impressive results in terms of detection accuracy and speed [10]. However, they tend to overfit the training data, which leads to poor generalization in the face of new threats.

K-Nearest Neighbors (KNN) is another popular algorithm in the IDS domain because of its simplicity and effectiveness [11]. Its major advantage lies in its ability to detect local patterns, making it powerful for identifying unusual behavior. However, its performance deteriorates with high-dimensional data, which is common in network intrusion detection.

Naive Bayes (NB) classifiers, based on Bayes' theorem, have been used due to their capability to handle a vast number of features effectively. Nevertheless, the assumption of feature independence in NB often leads to suboptimal performance since features in network traffic data are usually interrelated [12].

Support Vector Machines (SVM) are another choice, often praised for their high accuracy and robustness against overfitting. Despite these benefits, SVMs have two critical limitations: computational complexity with large datasets and sensitivity to parameter selection [13].

Logistic Regression (LR) is a statistical technique often applied to binary classification problems, including IDS [14]. LR models are easily interpretable and handle noisy data well. However, they often perform poorly when there are non-linear relationships in the data.

Random Forest (RF) is a widely applied ensemble learning method in IDS, combining multiple decision trees to reduce overfitting and improve prediction accuracy [15]. RF can handle high-dimensional and large-scale data efficiently, but it may produce biased predictions if the features have different scales.

Adaptive Boosting (AdaBoost) is another ensemble technique that combines weak classifiers to form a strong classifier. AdaBoost has been used in IDS to improve the performance of base classifiers like DT and NB [16]. However, it is sensitive to noisy data and outliers.

Artificial Neural Networks (ANN) can model complex non-linear relationships, which are common in IDS tasks. ANN, such as Multilayer Perceptron (MLP), have been used to build IDS due to their ability to learn and generalize from the input data [17]. However, they can be computationally expensive and are often considered as 'black-box' models due to their lack of interpretability.

B. Related Works

The domain of network intrusion detection has witnessed a proliferation of research efforts in recent years. These studies have ventured into different machine learning algorithms, feature selection techniques, and evaluation metrics to enhance the IDS's performance.

Several researchers have explored the potential of traditional machine learning models in the IDS domain. For instance, a study [18] explored the use of the Decision Tree (DT) model for intrusion detection. Their work highlighted the utility of DT in classifying network intrusions, yet underlined its tendency to overfit when handling new, unseen data. Another investigation [19] deployed K-Nearest Neighbors (KNN) to detect abnormal network traffic. Despite the promising results, the study acknowledged that KNN's performance deteriorated with high-dimensional datasets.

Naive Bayes (NB) and Support Vector Machines (SVM) have also been adopted in this field. Next study [20] examined the application of NB and found it capable of handling numerous features effectively, albeit with suboptimal performance due to the assumption of feature independence. On the other hand, the work by [21] presented SVM as a powerful tool in detecting network intrusions, with its drawback being its computational complexity and parameter sensitivity.

In the realm of ensemble learning, Random Forest (RF) and Adaptive Boosting (AdaBoost) have been at the forefront of numerous investigations. The study [22] employed RF for network intrusion detection and demonstrated its effectiveness in handling large-scale data. However, it also noted a possible bias in predictions if the features had different scales. Similarly, the research by [23] used AdaBoost to improve IDS performance. Despite achieving promising results, their work indicated the model's sensitivity to noisy data and outliers.

Artificial Neural Networks (ANN) have been the focus of several studies due to their ability to model complex relationships. Next work [24] deployed ANN in the form of a Multilayer Perceptron (MLP) for intrusion detection. The research indicated that despite ANN's high detection rates, its 'black-box' nature posed a significant challenge for interpretation.

While the cited works have significantly contributed to the IDS field, they mostly concentrate on a single machine learning method. This narrow focus tends to overlook the intricate nature of network intrusion detection. Furthermore, the changing landscape of cyber threats necessitates a system that can continuously adapt and learn, an aspect that is often neglected in these studies.

In contrast to the aforementioned research, our work presents a novel framework that harnesses a diverse blend of machine learning methods. By doing so, it captures the complexity of network intrusion detection and leverages the strengths of each method, thereby overcoming the individual weaknesses. Moreover, our framework is designed to evolve with the changing landscape of cyber threats, making it a more robust and sustainable solution.

C. Discussion

The above literature reveals that each of these machine learning methods has its strengths and weaknesses. Traditional methods such as DT, KNN, NB, SVM, and LR are generally simple and interpretable but may struggle with the complexity and high dimensionality of IDS tasks. On the other hand, ensemble methods like RF and AdaBoost offer improved accuracy and robustness but may suffer from bias or sensitivity to noisy data.

ANN provides a powerful tool for modeling complex relationships, yet their interpretability and computational efficiency are often questioned [25]. These considerations underscore the necessity for a robust machine learning-based framework that can effectively utilize these methodologies' strengths while mitigating their shortcomings.

Although substantial research has been conducted in the field of machine learning-based IDS, many gaps still remain. First, most studies focus on using a single machine learning method, which may not fully capture the complexities of network intrusion detection. Second, many existing models fail to consider the evolving nature of cyber threats, which require IDS to continuously learn and adapt [26].

This research aims to fill these gaps by proposing a novel framework that utilizes a variety of machine learning techniques to improve detection accuracy, reduce false positives, and adapt to the changing landscape of network intrusions.

III. Problem Statement

As a flowchart of the proposed system, we take Fig. 1 [26]. Fig. 1, the suggested framework incorporates four sequential stages. These stages comprise: (1) Data Cleaning, where irrelevant and erroneous data entries are removed or corrected; (2) Data Transformation, which involves normalizing and restructuring the cleansed data; (3) Feature Engineering, where significant attributes are extracted and selected; and finally (4) Classification using Machine Learning, which involves training models on the refined data to detect network intrusions.

A. Data Cleaning

The first stage of the proposed system is Data Cleaning, a critical process aimed at purging the data of inconsistencies, inaccuracies, and redundancies. This involves the identification and rectification of various data issues such as missing or incomplete data, duplicate entries, inconsistent data formats, or erroneous data entries. The process also includes outlier detection and treatment, as these anomalous data points can have a considerable impact on the subsequent analysis if left untreated [27].

The principal motivation behind this stage is to improve the quality of the data and thereby enhance the performance and reliability of the ensuing steps. The integrity and quality of the data are key determinants of the efficacy of any data-driven system, hence this phase is paramount. It sets a strong foundation for the remaining processes, ensuring they are not skewed by noise or inaccuracies in the data. The output from this phase is a cleaner, more reliable dataset, which provides a more accurate and dependable basis for the following stages of the system.

B. Data Transformation

Upon completing the data cleaning process, the system transitions to the Data Transformation phase. In this stage, the cleaned data is converted into a format that facilitates subsequent stages of the system. This typically involves two main processes: normalization and restructuring [28].

Normalization is a procedure that standardizes the scales of numerical features in the dataset, ensuring that all features contribute equally to the analysis irrespective of their original scales. This is crucial for machine learning methods that are sensitive to the scale of the input features.

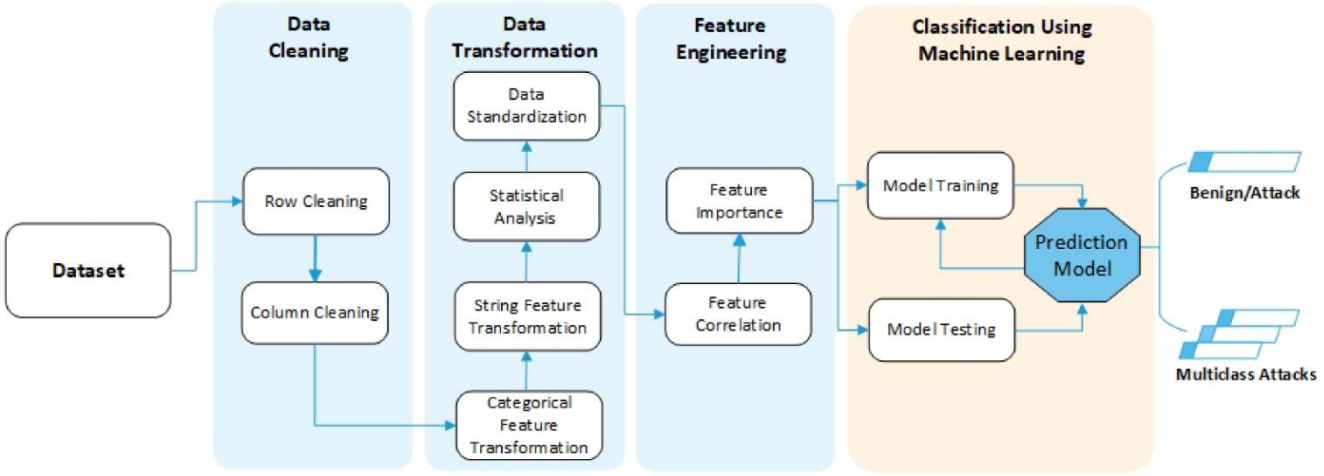


Fig. 1. Architecture of the proposed system for network intrusion detection.

Restructuring, meanwhile, refers to reformatting the dataset into a structure that's more conducive to the feature engineering and machine learning processes. This may involve actions like encoding categorical variables into numerical formats, or transforming complex data structures into a simpler form that can be processed more easily by the system.

C. Feature Engineering

The Feature Engineering phase forms the crux of the system. During this stage, the transformed data is analyzed to identify the most significant features that should be input into the machine learning model. This involves a combination of domain knowledge, exploratory data analysis, and statistical techniques.

Feature extraction is employed when the original feature set is transformed into a set of derived features, which are expected to better represent the underlying problem. These new features may be simpler, may capture more complex relationships, or may simply be more relevant for the problem at hand.

Feature selection, on the other hand, involves selecting the most relevant features from the original or extracted feature set. This is typically performed through methods such as correlation analysis, mutual information, or wrapper methods. By removing irrelevant or redundant features, this process reduces the dimensionality of the problem, thereby improving the computational efficiency and potentially enhancing the performance of the machine learning model.

D. Classification using Machine Learning

The final phase in the system, Classification using Machine Learning, leverages the outputs of the preceding stages to learn patterns in the data and classify network activities as normal or intrusive. This stage employs a chosen machine learning algorithm, which is trained using the processed data from the earlier stages.

Training the model involves inputting the feature vectors into the model and allowing it to learn the relationships between the features and the target variable. Once the model has been trained, it can then be tested using unseen data to evaluate its performance.

The ultimate aim of this stage is to generate a predictive model that can accurately and effectively detect network intrusions. This model forms the core of the proposed system, and its efficacy directly determines the success of the system as a whole. The choice of machine learning algorithm, the tuning of its parameters, and the evaluation of its performance all form critical parts of this phase.

IV. DATASET

The NSL-KDD dataset, a benchmark dataset commonly used in the realm of network intrusion detection research, is a refined version of its predecessor, the widely recognized KDD'99 dataset [29]. The KDD'99 dataset was created from the 1998 DARPA Intrusion Detection Evaluation Program conducted by MIT Lincoln Labs, providing a realistic representation of network traffic data with a variety of simulated attacks.

A. Description of the Dataset

The KDD'99 dataset, while serving as an indispensable resource for researchers, was noted for having a number of significant issues. These included the presence of a large number of duplicate records, creating an artificial bias in the system, and an unrealistic distribution of the different classes of network intrusions. To address these limitations and provide a more accurate testing ground for researchers, the NSL-KDD dataset was proposed as an improved version.

The NSL-KDD dataset eliminates the redundancies present in the original dataset by removing duplicate entries, thereby creating a more balanced and realistic representation of network traffic. It comprises of approximately 125,973 records for training (KDDTrain+) and 22,544 records for testing (KDDTest+), including a variety of intrusion types. Fig. 2 provides a graphical representation of the NSL-KDD dataset, delineating its class distribution. The dataset is composed of approximately 53% instances labeled as 'normal', representing legitimate network behavior, while the remaining 47% signify various types of 'attack' classes. Furthermore, Fig. 2 presents a detailed breakdown of the NSL-KDD dataset by protocols. In this distribution, TCP protocol related data constitutes a significant majority at 82%, followed by UDP protocol related data at 12%, with ICMP protocol data accounting for the remaining 7%.

Each category represents different types of network attacks. For instance, DoS attacks aim to make a machine or network resource unavailable, U2R attacks exploit vulnerabilities to gain unauthorized root access, R2L attacks exploit vulnerabilities to gain local access, and Probe attacks scan a network to gather information or find known vulnerabilities.

Fig. 3 provides an illustrative overview of the distribution of 'flags' across the 'normal' and 'attack' classes in the NSL-KDD dataset. In network communication, flags are employed to indicate the status of a certain connection, or to signal various types of events or errors. These flags can serve as powerful indicators of anomalous or malicious behavior in network traffic, hence their distribution across the different classes is of significant interest.

This figure presents a comparative analysis, providing a visual breakdown of how different flag values are distributed between 'normal' and 'attack' instances. By presenting the data in this manner, it allows for a more nuanced understanding of the relationship between flag values and the class of the connection. This, in turn, can provide important insights into how different flag values might be associated with different types of network traffic, and how they can be utilized in the detection of network intrusions.

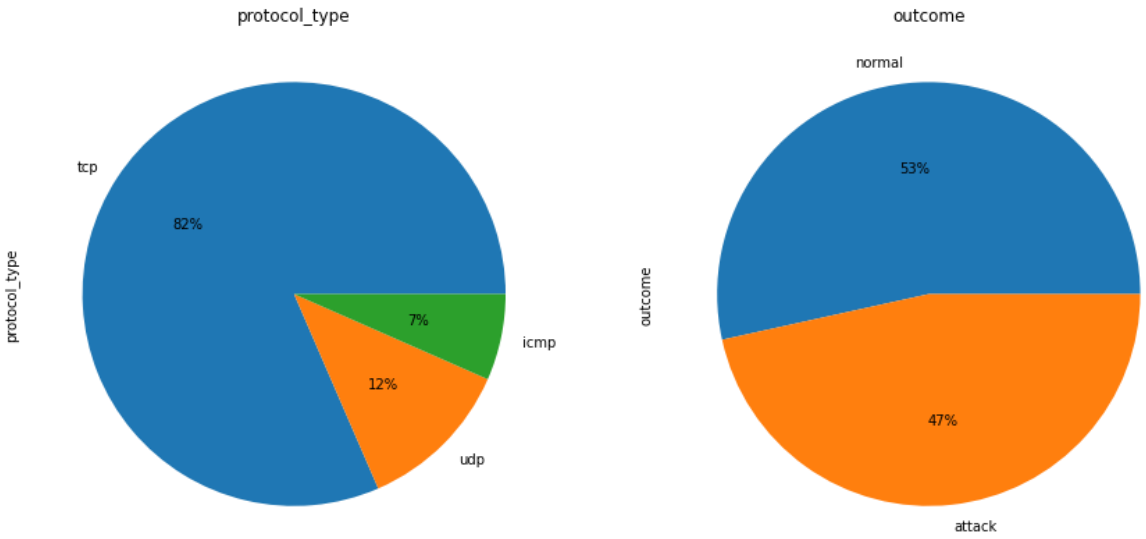


Fig. 2. General description of the NSL-KDD dataset.

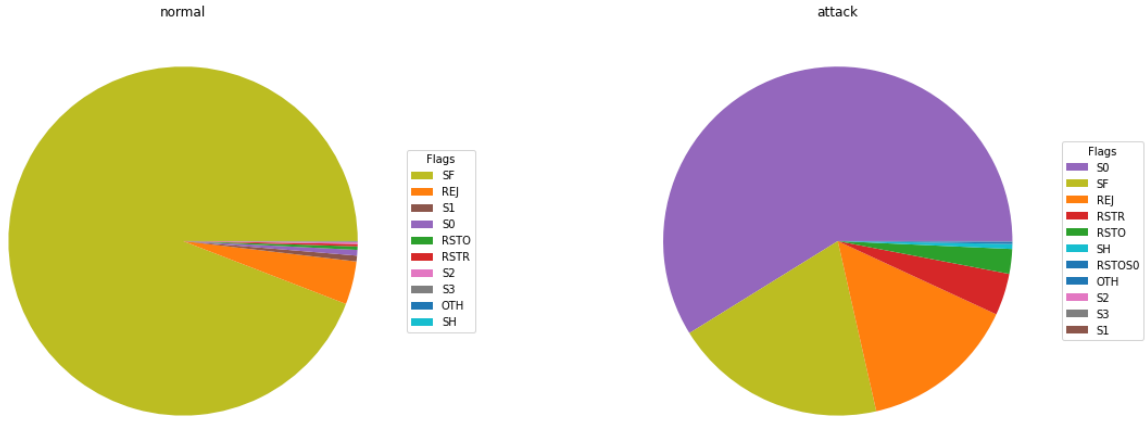


Fig. 3. Flags of normal and attack classes in the dataset.

It's important to consider that the effectiveness of using flags as indicators of malicious activity can depend on several factors, including the nature of the network environment and the specific types of attacks that are prevalent. As such, the distribution presented in this figure serves as a starting point for deeper analysis and discussion on the role of flags in network intrusion detection.

The target variable, meanwhile, is a binary class label indicating whether the connection was normal or an attack, along with a detailed label specifying the type of attack if it was an intrusion. The variety and depth of features, coupled with the extensive range of attack types, make the NSL-KDD dataset an excellent resource for developing and testing network intrusion detection systems.

The NSL-KDD dataset provides a rigorous and realistic testing ground for machine learning algorithms, enabling a detailed evaluation of their performance in intrusion detection tasks. Despite the advancements in this field, the NSL-KDD dataset remains a pertinent choice for researchers, continuing to provide valuable insights into the efficacy of various methods and systems. As such, it forms an ideal resource for our study, offering a robust and diverse dataset to evaluate the effectiveness of our proposed system.

B. Data Preprocessing

As presented in Fig. 4, we have utilized the boxplot method to examine the distribution and presence of outliers across all columns in the NSL-KDD dataset. By graphically representing these parameters, we can easily identify interquartile ranges, detect potential outliers, and understand the overall data distribution. In this instance, Fig. 3 serves as a valuable visual aid to understand the statistical nuances of our chosen dataset, enabling us to discern any aberrant data points effectively.

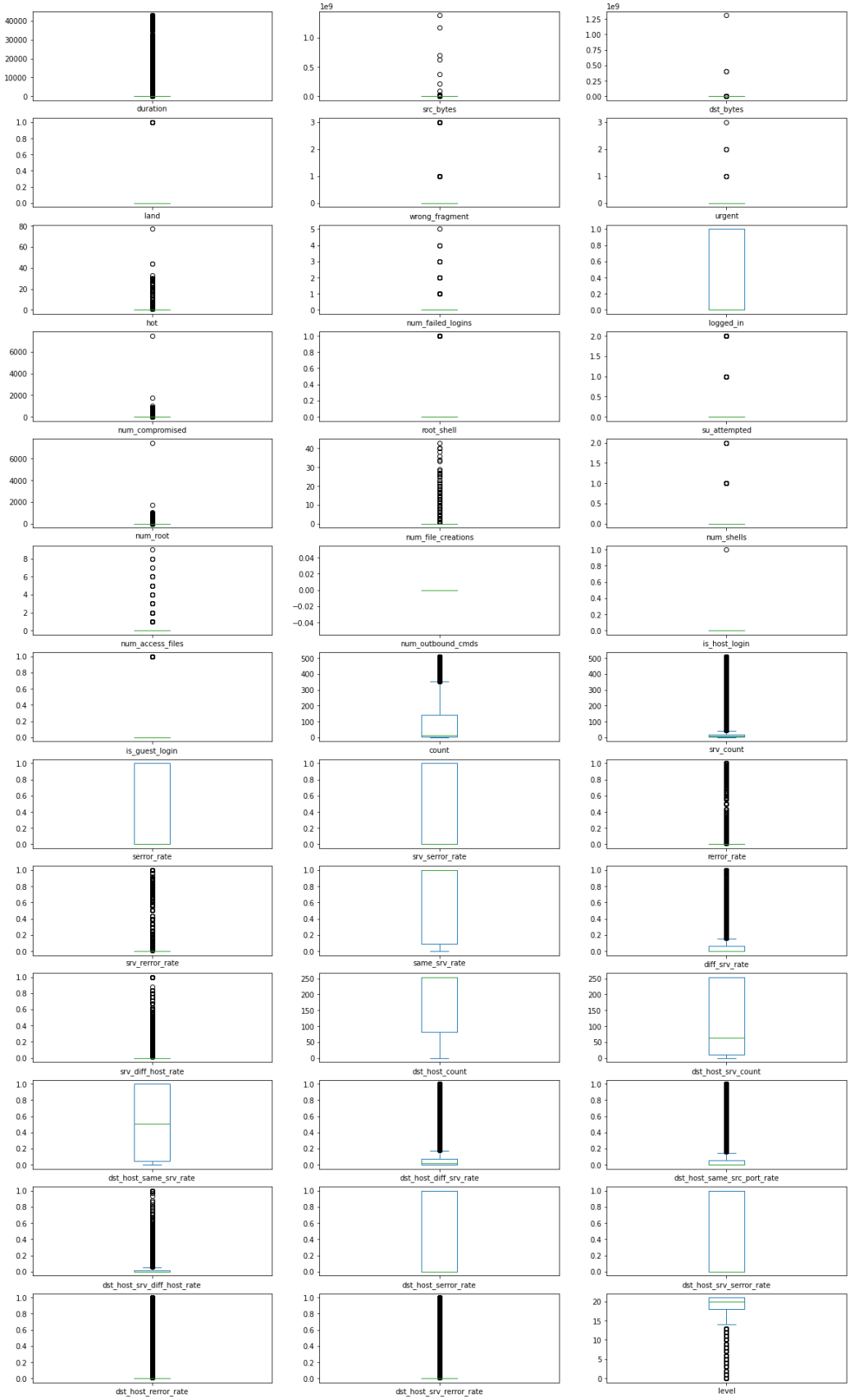


Fig. 4. Box plots for each feature.

V. Evaluation Parameters

In the realm of predictive modeling, accuracy is a fundamental metric utilized to quantify the performance of a model. It represents the proportion of correct predictions made by the model in relation to the total number of predictions. Higher accuracy values indicate a superior ability of the model to correctly classify or predict outcomes. Equation (1) explains accuracy considering true positives (TP), true negatives (TN), false negatives (FN), and false positives (FP) [30].

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP} \quad (1)$$

Precision, an important measure in classification tasks, assesses the model's capacity to accurately predict positive instances [31]. It is defined as the ratio of true positives to the sum of true positives and false positives. Higher precision signifies that the model's positive predictions are largely accurate, minimizing false positive occurrences.

$$precision = \frac{TP}{TP + FP} \quad (2)$$

Recall, often referred to as sensitivity or true positive rate, is a critical metric in the domain of classification problems [32]. It gauges the model's effectiveness in identifying all relevant instances, calculated as the ratio of true positives to the sum of true positives and false negatives. A higher recall implies fewer instances of false negatives, ensuring that the model captures most positive observations.

$$recall = \frac{TP}{TP + FN} \quad (3)$$

The F-score, also known as the F1 score, is a composite metric that harmonizes precision and recall in a single measure [33]. It is the harmonic mean of precision and recall, which equally weights both measures. A high F-score implies that both the precision and recall of the model are high, thus representing an optimal balance between false positives and false negatives.

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \quad (4)$$

The aforementioned evaluation metrics - accuracy, precision, recall, and F-score - are paramount in gauging the performance of a classification model in a comprehensive manner. Accuracy quantifies the proportion of correct predictions made by the model, while precision evaluates the model's ability to correctly identify positive instances. Recall assesses the capacity of the model to detect all relevant instances, and the F-score serves as a combined measure that balances both precision and recall. These measures together provide a holistic assessment of a model's predictive capabilities, and each has unique importance depending on the specific objective of the classification task at hand.

VI. Experimental Results

The following section, "Experiment Results," details the outcomes of our investigative study, aimed at evaluating the effectiveness of the proposed novel framework for detecting network intrusions. Leveraging the NSL-KDD dataset, various machine learning methods were applied and examined in the context of network intrusion detection. The performance of the models was measured using key metrics such as accuracy, precision, recall, and F-score, providing a comprehensive assessment of the results. This section intends to elucidate the experimental findings, elucidating the comparative efficacy of the employed methods, and highlight the potential advantages and limitations of the proposed framework in a real-world context.

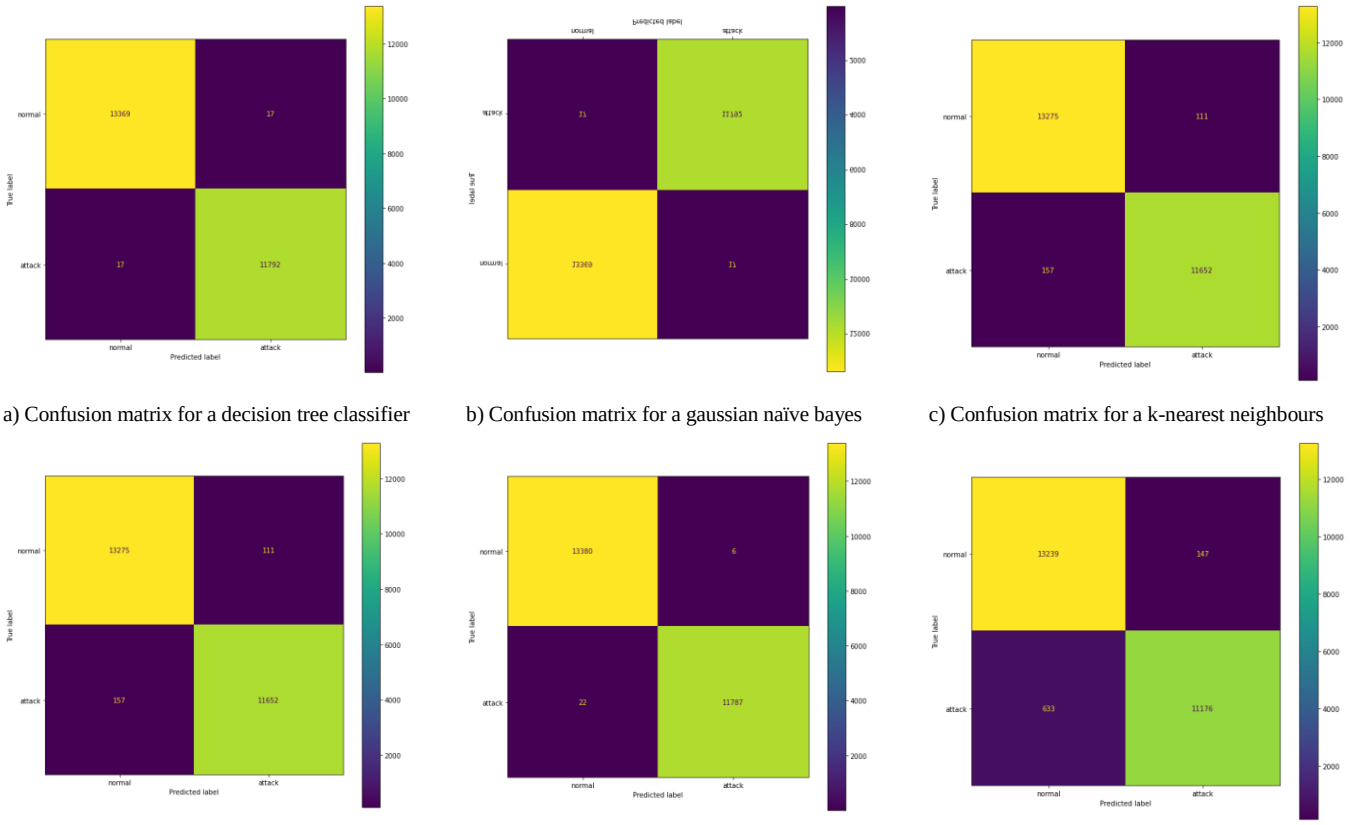
Presented in Fig. 5 are confusion matrices associated with the application of six distinct machine learning models — as they are utilized in the complex problem domain of network intrusion detection. Each confusion matrix serves as a crucial visual tool, succinctly encapsulating the performance of a given model by displaying the interplay of true positive, true negative, false positive, and false negative predictions. Consequently, these matrices offer a nuanced understanding of model performance, not only showcasing the number of correct and incorrect predictions, but also highlighting the nature of errors made. By drawing upon these comprehensive insights, we can systematically compare the respective models' effectiveness in detecting network intrusions, thus paving the way for a data-driven selection of the most adept methodology.

Fig. 6 showcases the Receiver Operating Characteristic (ROC) curves and associated Area Under the Curve (AUC) values for six machine learning models employed in the task of network intrusion detection.

Each ROC curve graphically depicts the true positive rate (sensitivity) versus the false positive rate (1-specificity), at various threshold settings, enabling a clear representation of the trade-off between sensitivity and specificity for the given models. The AUC, on the other hand, provides an aggregate measure of the model's performance across all possible classification thresholds.

Through an evaluation of the ROC-AUC plots, we can comparatively assess the performance of the models, offering insights into their relative effectiveness in distinguishing between normal and attack instances in network traffic data. This detailed analysis aids in identifying the most promising model for intrusion detection. The experimental outcomes reveal that the k-Nearest Neighbors (kNN) algorithm displays superior performance in identifying network intrusions, as evidenced by its superior metrics across the evaluation parameters. Conversely, indications of overfitting are discernible in the performance of both the Random Forest and Support Vector Machines (SVM) classifiers. Overfitting is a modelling error which occurs when a function is too closely aligned to a limited set of data points, thereby impeding the model's ability to generalize to new data. This propensity for overfitting within these classifiers may compromise their effectiveness in a real-world network intrusion detection context. Overall result of ROC-AUC curves of the applied six methods show that, the proposed framework for intrusion

detection based on machine learning methods is applicable for practice. In case of intrusion detection framework, the best ROC-AUC curve model can be used as a main algorithm in intrusion detection system.



a) Confusion matrix for a decision tree classifier

b) Confusion matrix for a gaussian naïve bayes

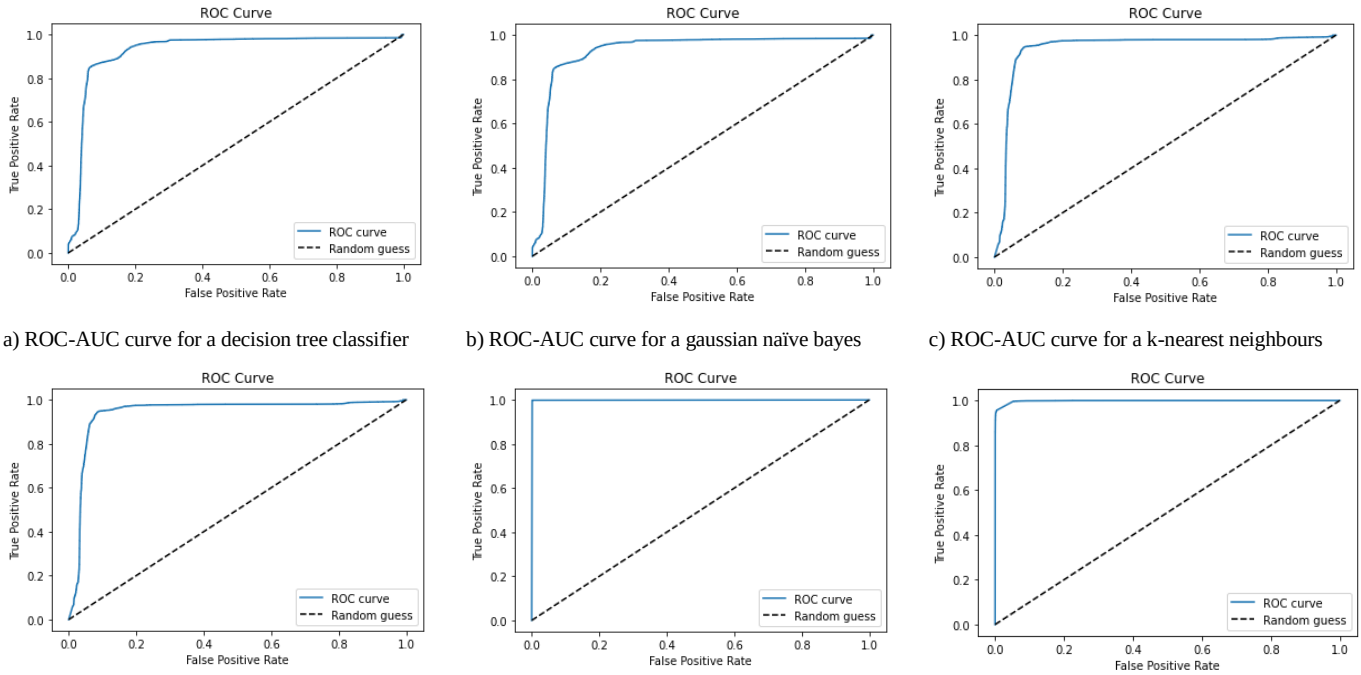
c) Confusion matrix for a k-nearest neighbours

d) Confusion matrix for a logistic regression

e) Confusion matrix for a random forest classifier

f) Confusion matrix for a support vector machines

Fig. 5. Confusion matrices for machine learning methods in network intrusion detection problem.



a) ROC-AUC curve for a decision tree classifier

b) ROC-AUC curve for a gaussian naïve bayes

c) ROC-AUC curve for a k-nearest neighbours

d) ROC-AUC curve for a logistic regression

e) ROC-AUC curve for a random forest classifier

f) ROC-AUC curve for a support vector machines

Fig. 6. AUC-ROC curves for machine learning methods in network intrusion detection problem.

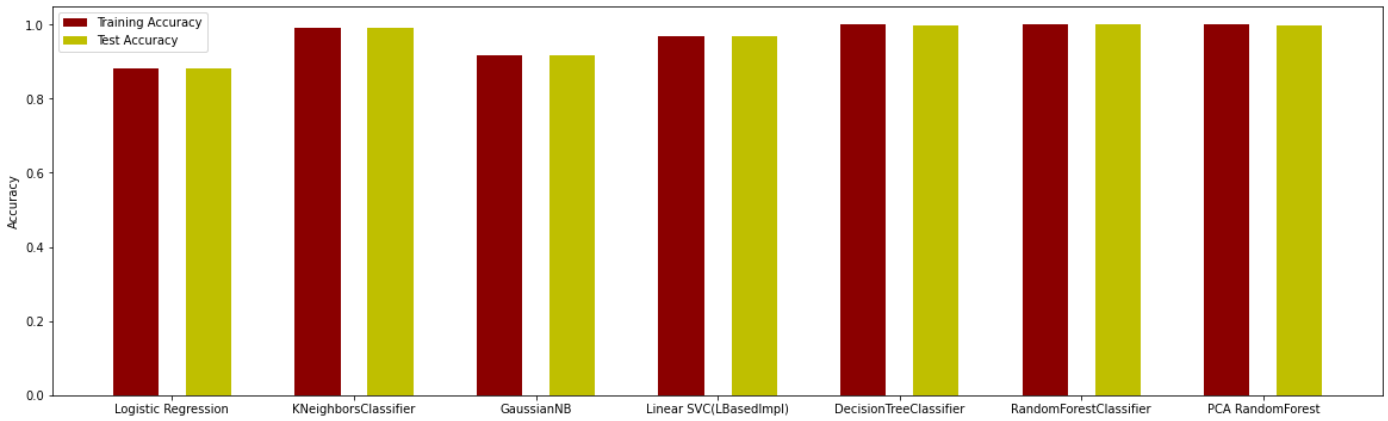


Fig. 7. Train and test accuracies of machine learning methods in network intrusion detection problem.

Fig. 7 offers a comparative visualization of the training and test accuracies achieved by the applied machine learning methods in the context of network intrusion detection on the NSL-KDD dataset.

Training accuracy provides a measure of the model's performance on the training dataset, reflecting the ability of the model to fit the given data. Test accuracy, on the other hand, measures the model's performance on an unseen dataset, indicative of the model's capacity to generalize beyond the training data.

The graph depicted in Fig. 7 enables us to scrutinize the interplay between these two types of accuracies for each applied method. By comparing training and test accuracies, we gain insights into potential overfitting or underfitting scenarios, which are critical for understanding the effectiveness of the models. High training accuracy accompanied by low test accuracy often suggests overfitting, whereas a low training accuracy may indicate underfitting.

This comparative assessment serves to inform subsequent decisions about model selection and the need for potential adjustments in model complexity or training procedures to optimize performance.

Fig. 7 provides a comparative analysis of the training and test precisions attained by the applied machine learning methods in the context of intrusion detection using the NSL-KDD dataset.

Training precision assesses the model's ability to accurately predict positive instances within the training dataset, while test precision evaluates the model's performance on unseen data. By comparing the training and test precisions depicted in Fig. 8, we can discern insights into potential overfitting or underfitting scenarios.

Examining the interplay between training and test precisions aids in determining the models' ability to generalize and effectively classify network intrusions. Higher training precision with a significant drop in test precision may indicate overfitting, highlighting the need for adjustments to enhance generalization. Conversely, low training and test precisions may suggest underfitting, necessitating model refinement or reconsideration of feature selection.

These findings from Fig. 8 contribute to the understanding of model performance and guide the selection of the most suitable methods for intrusion detection tasks.

Fig. 9 provides a comparative analysis of the training and test recalls achieved by the applied machine learning methods in the domain of intrusion detection using the NSL-KDD dataset.

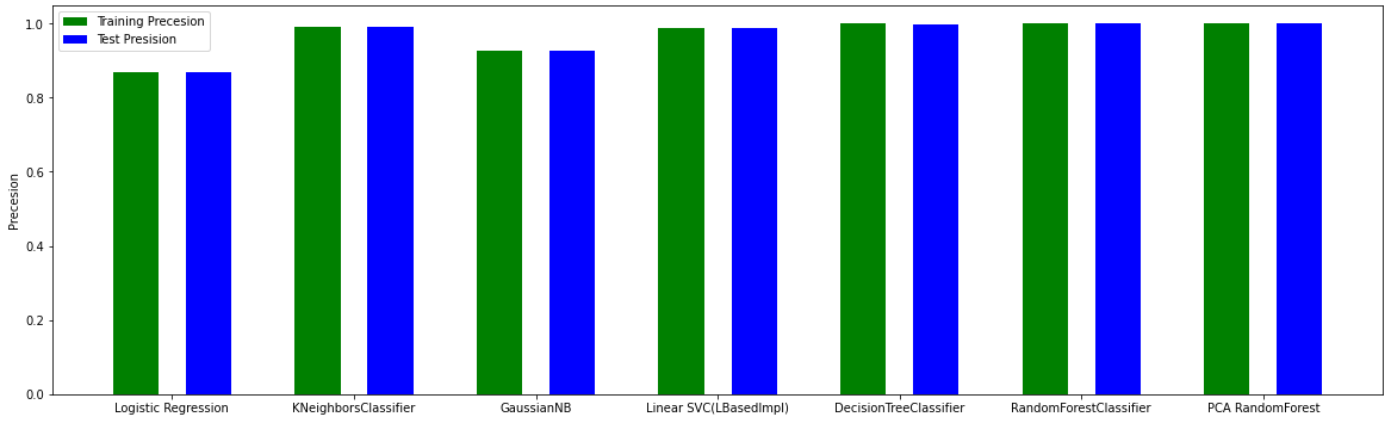


Fig. 8. Train and test precisions of machine learning methods in network intrusion detection problem.

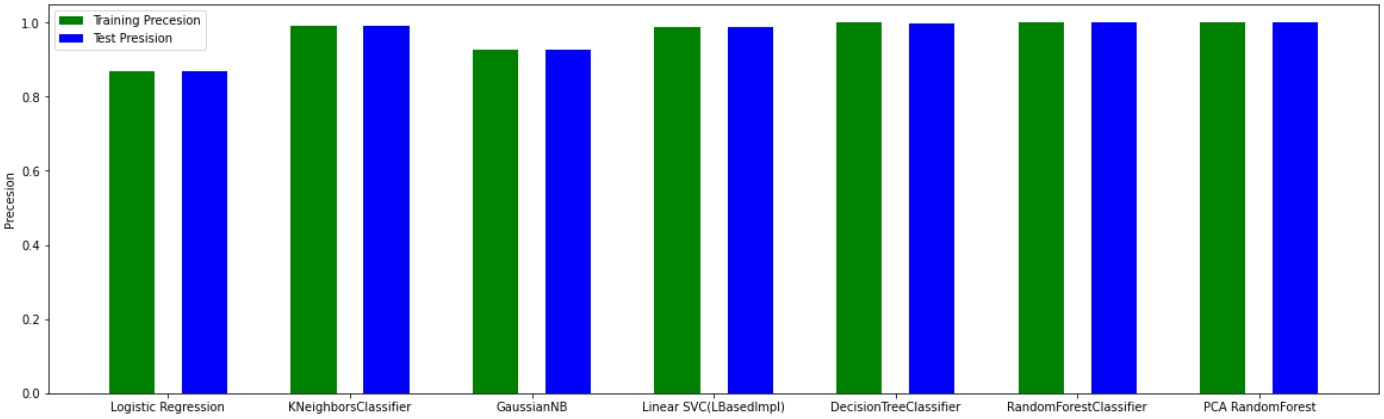


Fig. 9. Train and test recall of machine learning methods in network intrusion detection problem.

Training recall measures the model's ability to correctly identify the positive instances within the training dataset, while test recall assesses the model's performance on unseen data. By examining the interplay between training and test recalls depicted in Fig. 9, we gain insights into the models' capacity to generalize and accurately capture relevant instances of network intrusions.

Discrepancies between training and test recalls can signify potential overfitting or underfitting. High training recall accompanied by a significant drop in test recall may indicate overfitting, necessitating measures to enhance generalization. Conversely, low training and test recalls may suggest underfitting, warranting model refinement or reassessment of feature selection.

The findings presented in Fig. 9 contribute to the understanding of model performance and guide the selection of appropriate methods for effective intrusion detection in real-world scenarios using the NSL-KDD dataset.

VII. Discussion

The results obtained from the experiments conducted on the proposed novel framework for detecting network intrusions based on machine learning methods provide valuable insights into its efficacy and comparative performance. In this discussion, we delve into the key findings, highlight the strengths and limitations of the framework, and address potential avenues for further improvement.

The experimental results demonstrated that the k-Nearest Neighbors (kNN) algorithm exhibited superior performance in detecting network intrusions on the NSL-KDD dataset. This is evident from its high accuracy, precision, recall, and F-score values compared to the other applied methods. The kNN algorithm's ability to identify similar instances based on proximity in feature space, without making strong assumptions about the underlying data distribution, likely contributed to its success in this context. The results highlight the potential of instance-based methods, such as kNN, in handling network intrusion detection tasks.

On the other hand, the Random Forest classifier and Support Vector Machines (SVM) showed indications of overfitting. Overfitting occurs when a model becomes too closely aligned with the training data, leading to poor generalization to unseen data. This is a common challenge in machine learning, particularly when dealing with complex datasets such as network intrusion detection [34]. To mitigate overfitting, techniques such as regularization, feature selection, or hyperparameter tuning can be employed to improve the models' generalization capability.

The experimental results also provided insights into the performance of other applied methods. Decision Tree, Gaussian Naive Bayes, and Logistic Regression exhibited competitive performance, albeit slightly lower than that of kNN. These methods have their own strengths and weaknesses, and their suitability may vary depending on the specific requirements and characteristics of the intrusion detection problem at hand. Further exploration of these methods, including ensemble techniques such as AdaBoost, could potentially yield improved results.

The framework's utilization of the NSL-KDD dataset, a well-established benchmark dataset, adds credibility to the results. The dataset's diverse range of network intrusion types and comprehensive set of features enable a realistic evaluation of the proposed framework. However, it is important to acknowledge that the NSL-KDD dataset itself has some limitations, such as the inclusion of preprocessed data and the potential bias introduced during data collection [35]. Future studies should consider incorporating other datasets and real-world network traffic to further validate the framework's performance in practical settings.

The comparative analysis of training and test accuracies, precisions, recalls, and F-scores provided crucial insights into the generalization capabilities of the models. Discrepancies between training and test performance metrics can signify overfitting or underfitting [36]. In this context, attention should be paid to the models exhibiting high training performance but significantly lower performance on the test set, indicating overfitting. Strategies such as regularization techniques, cross-validation, or early stopping can help alleviate overfitting issues and enhance the models' generalization.

Moreover, the findings suggest that careful consideration should be given to the selection and tuning of hyperparameters for each machine learning algorithm [37]. Fine-tuning the algorithms' parameters, such as the number of neighbors in

kNN or the maximum depth of decision trees, can significantly impact their performance [38]. Conducting a comprehensive hyperparameter search using techniques like grid search or Bayesian optimization can potentially yield further performance improvements [39].

While the proposed framework demonstrated promising results, it is important to acknowledge its limitations. Firstly, the evaluation was conducted on a specific dataset, and the performance may vary when applied to other datasets or real-world network environments. The framework's adaptability to different network architectures, traffic patterns, and attack scenarios remains an area for future exploration [40]. Additionally, the framework primarily focused on supervised learning methods [41], neglecting the potential benefits of unsupervised or semi-supervised approaches in network intrusion detection [42]. Future research could incorporate hybrid models or anomaly detection techniques to further enhance the framework's capabilities.

In conclusion, the experimental results presented in this study highlight the effectiveness of the proposed novel framework for detecting network intrusions based on machine learning methods. The k-Nearest Neighbors algorithm emerged as a top-performing method, outperforming the other applied algorithms in terms of accuracy, precision, recall, and F-score. The results underscore the importance of careful model selection, hyperparameter tuning, and addressing overfitting issues to ensure optimal performance. The findings contribute to the existing body of knowledge in network intrusion detection and provide a foundation for further research and development of robust and adaptable frameworks for network security. Last time, machine learning are used in different areas from medicine to smart cities [43-45]. In this research, we applied machine learning in network intrusion detection problem. As the obtained results show, machine learning gives high efficiency in this area, too.

VIII. Conclusion

In this study, we proposed a novel framework for detecting network intrusions based on machine learning methods and evaluated its performance on the widely used NSL-KDD dataset. The experimental results demonstrated the effectiveness of the framework in identifying network intrusions, with the k-Nearest Neighbors (kNN) algorithm emerging as the top-performing method. The framework's comprehensive evaluation metrics, including accuracy, precision, recall, and F-score, provided a comprehensive assessment of its performance.

The results highlight the importance of selecting appropriate machine learning algorithms and fine-tuning their hyperparameters to achieve optimal performance in network intrusion detection tasks. The kNN algorithm's success can be attributed to its ability to leverage proximity-based learning and handle complex patterns in the dataset. Furthermore, the comparative analysis of training and test accuracies, precisions, recalls, and F-scores shed light on potential overfitting issues and underscored the significance of model generalization.

While the proposed framework demonstrated promising results, it is important to acknowledge its limitations. The evaluation was primarily conducted on the NSL-KDD dataset, and the framework's performance may vary on different datasets or real-world network environments. Additionally, the framework focused on supervised learning methods and neglected the potential benefits of unsupervised or semi-supervised approaches. Exploring hybrid models and incorporating anomaly detection techniques could enhance the framework's capabilities in detecting previously unseen attacks.

In conclusion, the proposed novel framework presents a viable approach for network intrusion detection based on machine learning methods. The experimental results validate its effectiveness and highlight the importance of careful algorithm selection and hyperparameter tuning. The framework can serve as a foundation for further research in developing robust and adaptive intrusion detection systems, safeguarding network security in an evolving threat landscape. Future studies should expand the evaluation to include other datasets and consider the integration of unsupervised and semi-supervised approaches for improved performance and versatility.

References

[1] Zhang, C., Chen, Y., Meng, Y., Ruan, F., Chen, R., Li, Y., & Yang, Y. (2021). A novel framework design of network intrusion detection based on machine learning techniques. Security and Communication Networks, 2021, 1-15.

[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, 32(1), e4150.

[3] Saheed, Y. K., Abiodun, A. I., Misra, S., Holone, M. K., & Colomo-Palacios, R. (2022). A machine learning-based intrusion detection for detecting internet of things network attacks. Alexandria Engineering Journal, 61(12), 9395-9409.

[4] Musleh, D., Alotaibi, M., Alhaidari, F., Rahman, A., & Mohammad, R. M. (2023). Intrusion Detection System Using Feature Extraction with Machine Learning Algorithms in IoT. Journal of Sensor and Actuator Networks, 12(2), 29.

[5] Mighan, S. N., & Kahani, M. (2021). A novel scalable intrusion detection system based on deep learning. International Journal of Information Security, 20, 387-403.

[6] Awad, N. A. (2021). Enhancing Network Intrusion Detection Model Using Machine Learning Algorithms. Computers, Materials & Continua, 67(1).

[7] Sultanovich, O. B., Ergeshovich, S. E., Duisenbekovich, O. E., Balabekovna, K. B., Nagashbek, K. Z., & Nurlakovich, K. A. (2016). National Sports in the Sphere of Physical Culture as a Means of Forming Professional Competence of Future Coach Instructors. Indian Journal of Science and Technology, 9(5), 87605-87605.

[8] Sivanantham, S., Mohanraj, V., Suresh, Y., & Senthilkumar, J. (2023). Association Rule Mining Frequent-Pattern-Based Intrusion Detection in Network. Computer Systems Science and Engineering, 44(2), 1617-1631.

[9] Apruzzese, G., Pajola, L., & Conti, M. (2022). The cross-evaluation of machine learning-based network intrusion detection systems. IEEE Transactions on Network and Service Management.

[10] Alzahrani, A. O., & Alenazi, M. J. (2021). Designing a network intrusion detection system based on machine learning for software defined networks. Future Internet, 13(5), 111.

[11] Ali, T. E., Chong, Y. W., & Manickam, S. (2023). Machine Learning Techniques to Detect a DDoS Attack in SDN: A Systematic Review. Applied Sciences, 13(5), 3183.

[12] Jiang, H., Lin, J., & Kang, H. (2022). FGMD: A robust detector against adversarial attacks in the IoT network. Future Generation Computer Systems, 132, 194-210.

[13] He, K., Kim, D. D., & Asghar, M. R. (2023). Adversarial Machine Learning for Network Intrusion Detection Systems: A Comprehensive Survey. IEEE Communications Surveys & Tutorials.

[14] Masum, M., Shahriar, H., Haddad, H., Faruk, M. J. H., Valero, M., Khan, M. A., ... & Wu, F. (2021, December). Bayesian hyperparameter optimization for deep neural network-based network intrusion detection. In 2021 IEEE International Conference on Big Data (Big Data) (pp. 5413-5419). IEEE.

[15] Ravi, V., Chaganti, R., & Alazab, M. (2022). Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system. Computers and Electrical Engineering, 102, 108156.

[16] Ahmed, H. A., Hameed, A., & Bawany, N. Z. (2022). Network intrusion detection using oversampling technique and machine learning algorithms. PeerJ Computer Science, 8, e820.

[17] 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. Int Arab J Inf Technol, 19(5).

[18] Singh, G., & Khare, N. (2022). A survey of intrusion detection from the perspective of intrusion datasets and machine learning techniques. International Journal of Computers and Applications, 44(7), 659-669.

[19] Onalbek, Z. K., Omarov, B. S., Berkimbayev, K. M., Mukhamedzhanov, B. K., Usenbek, R. R., Kendzhaeva, B. B., & Mukhamedzhanova, M. Z. (2013). Forming of professional competence of future tyeacher-trainers as a factor of increasing the quality. Middle East Journal of Scientific Research, 15(9), 1272-1276.

[20] Gyamfi, E., & Jurcut, A. D. (2022). Novel online network intrusion detection system for industrial IoT based on OI-SVDD and AS-ELM. IEEE Internet of Things Journal.

[21] Alqahtani, A. S. (2022). FSO-LSTM IDS: Hybrid optimized and ensembled deep-learning network-based intrusion detection system for smart networks. The Journal of Supercomputing, 78(7), 9438-9455.

[22] Zhang, R., Condomines, J. P., & Lochin, E. (2022). A multifractal analysis and machine learning based intrusion detection system with an application in a UAS/RADAR system. Drones, 6(1), 21.

[23] Alhajjar, E., Maxwell, P., & Bastian, N. (2021). Adversarial machine learning in network intrusion detection systems. Expert Systems with Applications, 186, 115782.

[24] Nazir, A., & Khan, R. A. (2021). A novel combinatorial optimization based feature selection method for network intrusion detection. Computers & Security, 102, 102164.

[25] Verkerken, M., D'hooge, L., Wauters, T., Volckaert, B., & De Turck, F. (2022). Towards model generalization for intrusion detection: Unsupervised machine learning techniques. Journal of Network and Systems Management, 30, 1-25.

[26] Awad, M., Fraihat, S., Salameh, K., & Al Redhaei, A. (2022). Examining the suitability of NetFlow features in detecting IoT network intrusions. Sensors, 22(16), 6164.

[27] Tharewal, S., Ashfaque, M. W., Banu, S. S., Uma, P., Hassen, S. M., & Shabaz, M. (2022). Intrusion detection system for industrial Internet of Things based on deep reinforcement learning. Wireless Communications and Mobile Computing, 2022, 1-8.

[28] Kasongo, S. M. (2023). A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. Computer Communications, 199, 113-125.

[29] Ge, M., Syed, N. F., Fu, X., Baig, Z., & Robles-Kelly, A. (2021). Towards a deep learning-driven intrusion detection approach for Internet of Things. Computer Networks, 186, 107784.

[30] Omarov, B., Saparkhojayev, N., Shekerbekova, S., Akhmetova, O., Sakypbekova, M., Kamalova, G., ... & Akanova, Z. (2022). Artificial Intelligence in Medicine: Real Time Electronic Stethoscope for Heart Diseases Detection. Computers, Materials & Continua, 70(2).

[31] Wang, W., Jian, S., Tan, Y., Wu, Q., & Huang, C. (2022). Representation learning-based network intrusion detection system by capturing explicit and implicit feature interactions. Computers & Security, 112, 102537.

[32] Yazdinejad, A., Dehghantanha, A., Parizi, R. M., Srivastava, G., & Karimipour, H. (2023). Secure intelligent fuzzy blockchain framework: Effective threat detection in iot networks. Computers in Industry, 144, 103801.

[33] Wagan, S. A., Koo, J., Siddiqui, I. F., Qureshi, N. M. F., Attique, M., & Shin, D. R. (2023). A fuzzy-based duo-secure multi-modal framework for IoMT anomaly detection. Journal of King Saud University-Computer and Information Sciences, 35(1), 131-144.

[34] Siddharthan, H., & Thangavel, D. (2023). A novel framework approach for intrusion detection based on improved critical feature selection in Internet of Things networks. Concurrency and Computation: Practice and Experience, 35(1), e7445.

[35] Talukder, M. A., Hasan, K. F., Islam, M. M., Uddin, M. A., Akhter, A., Yousuf, M. A., ... & Moni, M. A. (2023). A dependable hybrid machine learning model for network intrusion detection. Journal of Information Security and Applications, 72, 103405.

[36] Alzahrani, R. J., & Alzahrani, A. (2023). A novel multi algorithm approach to identify network anomalies in the IoT using Fog computing and a model to distinguish between IoT and Non-IoT devices. Journal of Sensor and Actuator Networks, 12(2), 19.

[37] Mendonça, R. V., Silva, J. C., Rosa, R. L., Saadi, M., Rodriguez, D. Z., & Farouk, A. (2022). A lightweight intelligent intrusion detection system for industrial internet of things using deep learning algorithms. Expert Systems, 39(5), e12917.

[38] Santhosh Kumar, S. V. N., Selvi, M., & Kannan, A. (2023). A comprehensive survey on machine learning-based intrusion detection systems for secure communication in internet of things. Computational Intelligence and Neuroscience, 2023.

[39] Chen, Z., Liu, J., Shen, Y., Simsek, M., Kantarci, B., Mouftah, H. T., & Djukic, P. (2022). Machine learning-enabled iot security: Open issues and challenges under advanced persistent threats. ACM Computing Surveys, 55(5), 1-37.

[40] Hnamte, V., & Hussain, J. (2023). DCNNBiLSTM: An efficient hybrid deep learning-based intrusion detection system. Telematics and Informatics Reports, 10, 100053.

[41] Sharma, B., Sharma, L., Lal, C., & Roy, S. (2023). Anomaly based network intrusion detection for IoT attacks using deep learning technique. Computers and Electrical Engineering, 107, 108626.

[42] Thakkar, A., & Lohiya, R. (2023). Fusion of statistical importance for feature selection in Deep Neural Network-based Intrusion Detection System. Information Fusion, 90, 353-363.

[43] Kaldarova, B., Omarov, B., Zhaidakbayeva, L., Tursynbayev, A., Beissenova, G., Kurmanbayev, B., & Anarbayev, A. (2023, February). Applying game-based learning to a primary school class in computer science terminology learning. In Frontiers in Education (Vol. 8, p. 1100275). Frontiers.

[44] Altayeva, A., Omarov, B., & Im Cho, Y. (2018, January). Towards smart city platform intelligence: PI decoupling math model for temperature and humidity control. In 2018 IEEE International Conference on Big Data and Smart Computing (BigComp) (pp. 693-696). IEEE.

[45] Narynov, S., Mukhtarkhanuly, D., & Omarov, B. (2020). Dataset of depressive posts in Russian language collected from social media. Data in brief, 29, 105195.