UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

Application of Machine Learning Techniques for Multiclass Classification of Intrusions in Communication Networks

Paula Domenica Campaña Donoso

Ingeniería en Ciencias de la Computación

Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Ingeniera en Ciencias de la Computación

Quito, 13 de diciembre de 2024

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

HOJA DE CALIFICACIÓN
DE TRABAJO DE FIN DE CARRERA

Application of Machine Learning Techniques for Multiclass Classification of Intrusions in Communication Networks

Paula Domenica Campaña Donoso

Nombre del profesor, Título académico **Ricardo Flores, Ph. D**

Quito, 13 de diciembre de 2024

© DERECHOS DE AUTOR

Nombres y apellidos: Paula Domenica Campaña Donoso

Código: 00215572

Cédula de identidad: 1720583622

Lugar y fecha: Quito, 13 de diciembre de 2024

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en http://bit.ly/COPETheses.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on http://bit.ly/COPETheses.

RESUMEN

En el mundo hiperconectado actual, donde gran parte de la vida de las personas está documentada en línea, el riesgo de ser víctima de ciberataques aumenta cada día. Por este motivo, la implementación de capas de seguridad adicionales se ha convertido en una cuestión crítica que exige atención urgente. Este estudio explora el potencial de los algoritmos de aprendizaje automático como solución a este problema. Estos algoritmos no solo se adaptan cuando aumenta el número de casos de ataque, sino que también detectan estos eventos de forma autónoma, lo que elimina la necesidad de supervisión continua. De este modo, proporcionan una capa de seguridad efectiva para identificar y clasificar potenciales amenazas. En este trabajo, se seleccionaron tres algoritmos de aprendizaje automático, cada uno diseñado para la clasificación de ataques multi clase. El objetivo principal consiste en determinar cuál de los algoritmos obtuvo los mejores resultados al detectar ataques con la base de datos CICIDS2017. Para este propósito, se eligieron el algoritmo Gaussiano NB, el algoritmo de árboles extendidos y el clasificador multilayer perceptron para validar la hipótesis de investigación. Los modelos se evaluaron en tres escenarios, en los que se varió el número de características. Los resultados revelaron que el modelo de árboles extendidos fue el que mejor desempeño tuvo en la clasificación de ataques multi clase. Para evaluar la que tan significativos son estos resultados y validar la hipótesis, se realizó una prueba de Wilcoxon probabilística. El valor p resultante estuvo cerca de 0, lo que indica que los resultados eran estadísticamente significativos y no se debían a una coincidencia.

Palabras clave: Aprendizaje automático, Intrusiones, Multiclase, Extended Trees, GaussianNB, MLPClassifier

ABSTRACT

In today's hyperconnected world, where much of people's lives are documented online, the risk of falling victim to cybercriminal attacks grows daily. Consequently, the implementation of additional security layers has become a critical concern that demands urgent attention. This study explores the potential of machine learning algorithms as a solution to this problem. These algorithms not only adapt as the number of attack cases increases but also autonomously detect such events, eliminating the need for continuous human supervision. By doing so, they provide an effective security layer for identifying and classifying potential threats. In this work, three machine learning algorithms were selected, each designed for multiclass attack classification. The primary goal consists of determine which algorithm performs best at detecting attacks with the dataset, CICIDS2017. For this purpose, GaussianNB, Extended Trees, and MLPClassifier were chosen to validate the research hypothesis. The models were evaluated under three scenarios, varying the number of features. The results revealed that Extended Trees outperformed the others in multiclass attack classification. To assess how significant these results are and validate the hypothesis, a Wilcoxon probabilistic test was conducted. The resulting p-value was close to 0, indicating that the findings were statistically significant and not due to chance.

Key words: Machine Learning, Intrusion, Multiclass, Extended Trees, GaussianNB, MLPClassifier

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCTION

People are becoming increasingly hyperconnected, relying on various applications and platforms for their daily activities. This growing connectivity has exposed sensitive data to significant risks. One of the most pressing challenges arising from the internet use is the constant threat of cyberattacks targeting databases, networks, and other systems. Cybercriminals often aim to access this information to disrupt its use or compromise its integrity. Although mechanisms have been developed to protect sensitive data, they are not foolproof and have inherent limitations. This reality has driven the search for more effective alternatives, with machine learning emerging as a promising solution for anomaly detection and attack classification.

To address these challenges, various studies have been conducted to identify the most effective methods for anomaly detection. The primary goals of these studies are to prevent attacks, establish stronger layers of protection, and improve cybersecurity resilience. Key concepts in this area include datasets, which serve as the foundation for experimentation; algorithms, which are analyzed and compared for their detection capabilities; and evaluation parameters, which determine the effectiveness of these algorithms in multiclass anomaly classification. Additionally, this research emphasizes understanding multiclass classification and anomalies, as these concepts are integral to achieving accurate and reliable detection.

Over the years, traditional solutions like firewalls and antivirus systems have been employed to prevent unwanted elements. Additional strategies include limiting access with VPNs to secure private networks, regularly backing up database information, frequently updating browsers and operating systems, training company personnel to monitor data, checking emails for phishing attempts, and improving password security to create stronger protection layers [22][23].

However, these solutions share a common limitation: they are not automated. Each requires human intervention for monitoring and maintenance, consuming valuable time that could be allocated to other tasks, such as solving problems or advancing projects. To address this challenge, a promising alternative is the implementation of anomaly detection using machine learning algorithms. These algorithms, which consist of a set of instructions to analyze, investigate, and evaluate large datasets, provide efficient solutions to complex problems. Machine learning models can be trained to detect potential attacks, offering an additional layer of security without requiring constant human oversight. For instance, distributed denial-of-service (DDoS) attacks are designed to render machines or networks inaccessible to users by overwhelming them with traffic. A conventional approach to defending against such attacks is to implement a firewall to prevent the traffic from affecting the user. However, this method is not entirely effective in preventing these attacks. In contrast, machine learning models leverage historical data to identify and anticipate the patterns of these attacks, thereby providing a more robust defense than traditional solutions.

In this study, four primary algorithms are assessed. These include the Random Forest Classifier, which assists in feature selection [2], as well as the GaussianNB, Extended Trees, and MLPClassifier models. The objective of this assessment is to evaluate the performance of these three distinct models and identify the most effective method for intrusion detection. This is done with the aim of enhancing communication network security and strengthening data protection. In order to evaluate the impact of dimensionality reduction on the results of the chosen metrics, a reduction in the number of features was applied. The Extended Trees Classifier exhibited the most accurate performance among the evaluated models across three scenarios: achieving 99% accuracy with 78 features, 98% with 20 features, and 97% with 10 features. This model consistently demonstrated superior performance compared to the other models, proving to be the most effective for this task.

RELATED WORKS

To identify the best datasets for this experiment, the most suitable algorithms for comparison, and the most effective parameters for evaluation, research was conducted by reviewing previous works from scientific articles and theses. For instance, Arizaga et al. [1] focused their investigation on the effectiveness of the Random Forest algorithm in detecting a specific type of attack: Distributed Denial of Service (DDoS). Their study evaluated how well this model distinguished attacks from benign data, specifically targeting DDoS detection. Similarly, Álvarez [2] concentrated on another type of attack, DoS Hulk, aiming to compare various machine learning models, such as DBScan and SVM, to assess their effectiveness in detecting this attack type within a telecommunication network.

Rodríguez [3], on the other hand, used the same dataset as Arizaga but approached it differently by grouping all the attacks into four main categories. He applied multiple models to identify the most effective one for detecting these grouped attacks. Furthermore, Rodríguez used Random Forest to visualize decision trees and measure accuracy, comparing binary classification (attack vs. no attack) to multiclass classification (identifying specific attack types), offering a distinct perspective compared to the previous studies.

Kumar et al. [4] adopted a unique approach by analyzing the performance of an unsupervised machine learning algorithm, MeanShift, with the same dataset used by the earlier authors. Their primary goal was to evaluate the efficiency of this model for anomaly detection, contributing a new perspective to this area of study. Finally, De Lima et al. [5] focused on the detection of two closely related yet prevalent attack types: DoS and DDoS. Their research assessed the effectiveness of machine learning models in identifying these attacks within a communication network, offering another valuable perspective.

All these authors conducted experiments using datasets that capture the communication network traffic containing various types of attacks. Each study aimed to identify the most effective mechanisms for detecting and diagnosing attacks present in the traffic, with an emphasis on classifying specific attack types. Additionally, a baseline algorithm was often employed as a reference point to compare against other models, with efficiency evaluated through various parameters. These efforts provided meaningful insights to help determine the most suitable algorithms for this task.

It was identified that the most commonly used datasets for the study of multiclass anomaly classification are: CICIDS2017 (which has 78 features and 15 attack categories), CSE-CIC-IDS2018 (with 79 features and 15 attack categories), and NSL-KDD (with 41 features and 40 attack categories). The latter is an updated version of the KDDCUP 99 dataset, which was previously widely used for anomaly detection. Each of these datasets has been designed to represent the traffic of a telecommunications network, with the exception of the KDDCUP 99 dataset, now known as NSL-KDD, which aims to represent the traffic of a network in a military environment. Nevertheless, all these databases are the most widely used because they represent similar attacks, allowing this experiment to be carried out appropriately.

The most common attacks present in these datasets are: DoS (Denial of Service), DDoS (Distributed Denial of Service), Botnet, brute force attacks, phishing attacks, R2L (Remote to Local), and U2R (User to Root). Although the presence of these attacks varies across datasets, they all represent the most common ways of compromising communication networks. The importance of these attacks lies in the fact that they illustrate the various techniques used to compromise network security.

In the scientific articles by Arizaga, et al. [1], Álvarez [2], Rodríguez [3], Kumar, et al. [4], and De Lima, et al. [5], not only is the rationale behind the selection of these datasets or

the different types of attacks analyzed, but the various machine learning algorithms used to identify the best multiclass anomaly classifier are also explored. Based on the selection of these elements, experiments were conducted in which several machine learning algorithms were applied to the selected datasets to detect the anomalies present and classify the types of attacks. In this way, the efficiency of each algorithm could be compared. The results of these studies concluded that the most effective algorithms for comparison are: Extended Trees, which allows differentiation between normal and malicious traffic, as well as identifying specific types of attacks; Neural Networks, which facilitate the classification of attack types; and finally, Naive Bayes, a probabilistic algorithm that, besides being independent of predictors, allows calculating the probability of a specific attack. Although each of these algorithms functions differently, the experiments conducted in the revised articles have demonstrated that they are sufficiently effective in detecting attacks, providing significant evidence and results that will support the selection of the best multiclass anomaly classifier.

Finally, it is essential to evaluate the effectiveness of each selected machine learning algorithm. To do this, it is crucial to establish various metrics that allow the evaluation of different aspects of each algorithm, such as precision, accuracy, among others, to solidly support the selection of the best machine learning approach. It is not recommended to rely solely on one parameter like accuracy, since although it measures the precision of the model, in imbalanced datasets it may not provide a sufficiently robust metric to assess the model's effectiveness [14]. Moreover, in databases influenced by multiple factors, accuracy might not be the most appropriate parameter, as it may not accurately reflect anomaly detection. For this reason, in addition to accuracy, the F1-Score will be used, a metric that balances precision and recall to obtain a more complete result. By employing these two parameters, a more thorough evaluation will be achieved, allowing the identification of the best algorithm.

On the other hand, it is possible to obtain results that may not be reliable or fail to fully support the investigation and the experiments conducted. To address this concern and ensure the validity of the findings, an additional statistical test was implemented alongside the chosen metrics. This test plays a dual role: first, it helps verify the reliability of the obtained results by assessing their consistency and significance; second, it evaluates the differences between the performance of the models in a more quantifiable manner. By doing so, this test provides a clearer understanding of how each model performs relative to the others and offers a visual representation of these differences, adding another perspective to the evaluation process. Such an approach ensures that the conclusions drawn from the study are both robust and well-supported.

PROPOSAL

Given the critical need to enhance the security of communications networks, identifying an effective solution to detect and classify various types of attacks was paramount. To address this, an experimental process was undertaken to determine the most suitable algorithm for multiclass intrusion detection.

In line with existing standards for machine learning, the most relevant guidelines for this study were identified based on their alignment with the project's objectives. The ISO/IEC 22989:2022 standard, titled Information technology — Artificial intelligence — Artificial intelligence concepts and terminology [9], was selected for its comprehensive definitions of machine learning concepts and objectives. This standard provides a robust framework for comparing and selecting algorithms based on critical factors such as security, precision, and robustness. Additionally, the ISO/IEC TS 4213:2022 standard, titled Information technology — Artificial intelligence — Assessment of machine learning classification performance [8], was utilized. This standard focuses on evaluating the suitability of selected models by examining their performance in model implementation, database composition, and the outcomes of conducted experiments.

Using these ISO standards as a foundation, a step-by-step process was designed to ensure adherence to the outlined guidelines and achieve the primary objectives of this study. As illustrated in Figure 1,

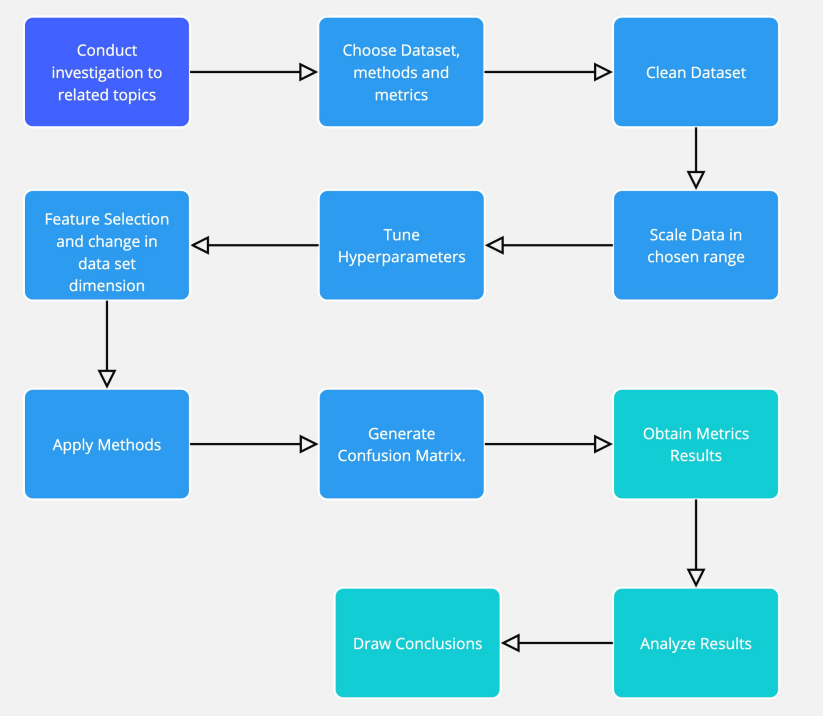


Figure 1. Process defined to conduct the research regarding the experiment to determine the best method for multiclass intrusion detection in communication networks[1]

the process began with a review of related articles and projects, which provided a deeper understanding of the phenomenon of intrusions and their various types, as well as an exploration of the methods applicable for detecting them.

A. Data Set Selection

The next stage involved selecting the dataset, methods, and metrics to be used in the experiment. For this study, the CICIDS2017 dataset was chosen, which contains 79 features: 78 of them are numerical values, and the last one is a label indicating whether the data is benign or represents an attack, and in this case, specifies the type of attack. The decision to use

[1] https://github.com/paulacd3005/ProyectoIntegrador.git

CICIDS2017 was based on its large data volume (approximately 2.8 million records) and the variety of attacks represented, which provides a solid basis for algorithm comparison.

B. Models Selection

Next, three main algorithms were selected for intrusion detection: Extended Trees, Naive Bayes, and Neural Networks. Specifically, within the Naive Bayes models, GaussianNB was chosen because, compared to other Naive Bayes variants, this model better meets the objective of efficient multiclass intrusion classification. According to Ige and Kiekintveld [15], GaussianNB is preferable in this context due to the Gaussian distribution present in the data, allowing effective analysis since the data in this dataset is continuous and statistically independent, as is typical in records of a communications network with multiple intrusions. Other types of Naive Bayes, such as Multinomial or Bernoulli, are not suited to this case, as the Multinomial variant is better suited to discrete data, such as text, while Bernoulli is focused on binary detection, which would limit its applicability in multiclass classification.

The equation that defines the operation of GaussianNB is as follows:

$$P(x_i|y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad (1)$$

where μ represents the mean of x_i within class y, σ is the standard deviation of x_i in that class, and x is the observed value of feature x_i. The first part of the formula normalizes the data, ensuring that the area under the curve equals 1. The second part, corresponding to the exponential component, describes the normal distribution of the data, thus facilitating the probability that a feature belongs to a specific class [17].

The second algorithm selected was MLPClassifier, a neural network that enables multiclass classification. The choice of a neural network is due to its effectiveness in processing data through multiple layers, where the weights of each data point influence the final classification, as shown in Figure 2.

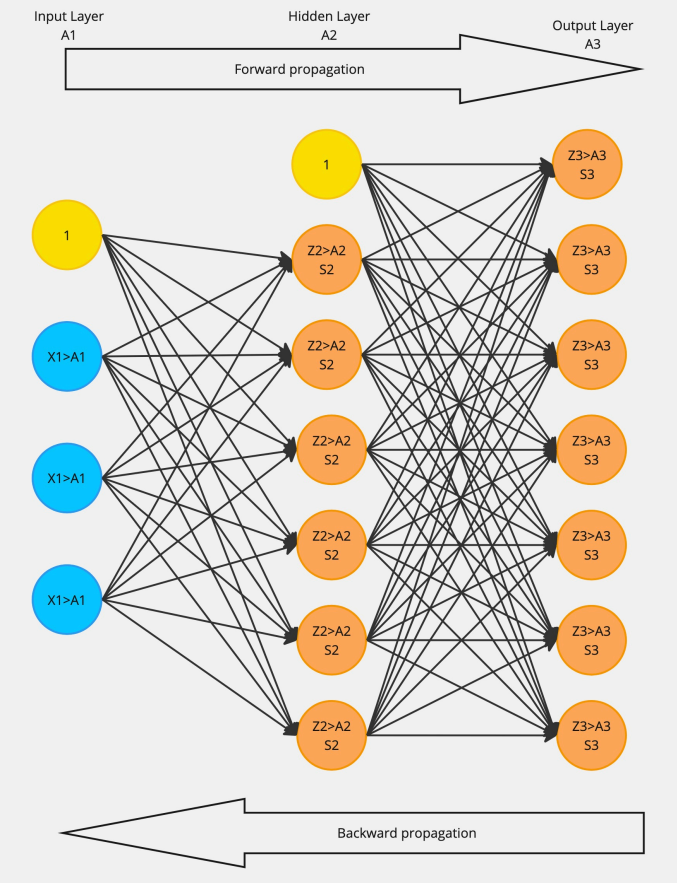


Figure 2. Example diagram of how neural networks work

According to Enslin and Neogi [11][13], neural networks like MLPClassifier do not require a deep structure (such as CNNs), which prevents overfitting in this specific experiment and facilitates accurate classification without the need for complex architectures. This is particularly useful in applications like intrusion detection, where efficient networks in terms of processing are desired [18].

The third algorithm selected was Extended Trees, chosen for its capability to perform multiclass detection and classify traffic as normal or malicious efficiently, leveraging data independence for rapid and accurate attack type detection, as discussed in the related works section.

To evaluate the effectiveness of these algorithms, four metrics were employed: Accuracy, Precision, Recall, and F1-Score. These metrics provide a comprehensive assessment

of the algorithms' performance in terms of efficiency and security. Among them, Precision and Recall are particularly relevant for binary classification, offering deeper insights into the models' ability to distinguish between benign and anomalous data. The chosen metrics are calculated using the following formulas [14]:

$$Accuracy: \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$Precision: \frac{TP}{TP + FP} \quad (3)$$

$$Recall: \frac{TP}{TP + FN} \quad (4)$$

$$F1 - Score: 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

C. Dataset Cleaning and Preparation

After selecting the dataset, algorithms, and evaluation parameters, a thorough cleaning of the dataset was conducted, where NaN and infinite values were replaced with the average value, as the data distribution is continuous and normal. This step ensures that the data remains within the expected range without introducing significant alterations. Regarding duplicate data, a partial (5%) random deletion was decided to preserve most of the data to ensure good model training and prevent deletion from affecting the results.

D. Normalization and Feature Selection

The data was then normalized to a range from 0 to 1, which avoids negative values and keeps values within a unified range, thus optimizing the performance of the models. Additionally, to reduce the dimensionality of the dataset and improve accuracy, the feature selection technique, specifically the Wrapper method, was applied. According to Sanki [16], there are three main methods for feature selection: Filters, Wrappers, and Embedded Methods.

Since this is a classification problem, the Wrappers method was chosen, allowing analysis optimization without a high computational cost. This method, known as Backward Feature Selection, selects the most relevant features by iterating through each feature and was implemented through Random Forests. In this context, Random Forests generates multiple decision trees from random subsets of data, selecting the most relevant features for model training, thereby ensuring better performance in the analysis.

The Random Forest Classifier model was used along with hyperparameter tuning, where the use of 100 trees was set to balance model performance and reduce the training time required to predict the most important features. Additionally, a random_state value of 42 was used to ensure replicability of results in each model execution, thereby maintaining consistency in data evaluation and training.

To compare the effectiveness of feature selection, tests were conducted using the 10 and 20 most important features. This approach was inspired by the work of Álvarez [2] and Rodríguez [3], who used 20 and 12 features, respectively, in their comparisons. By comparing these reduced feature sets with the original 78 features, the impact of dimensionality reduction on the final metrics could be identified. The objective of this process was to analyze how individual features influence anomaly detection and the classification of different anomaly types. Consequently, this dimensionality reduction was implemented to create two new scenarios, enabling an evaluation of each model's performance based on the metrics obtained.

Confusion matrices, which can be found in the GitHub repository, were created to evaluate the accuracy of Random Forests in selecting the most important features and to confirm that attack types were correctly classified, aligning with the goal of improving security in communication networks. Following this, hyperparameter tuning was performed on Extended Trees and MLPClassifier to optimize their performance by identifying suitable parameter combinations for each model.

For Extended Trees, the hyperparameters adjusted included n_estimators, which varied randomly between 100 and 500, and the maximum depth, set to a random value between 5 and 30. The minimum samples split ranged from 2 to 10, and the minimum number of leaf nodes was randomly generated between 1 and 10. Additionally, the maximum features analyzed were either the square root or the base-2 logarithm of the total features.

Similarly, MLPClassifier underwent hyperparameter tuning to refine its effectiveness. The hidden layer sizes tested included three configurations: (50, 50, 50), (50, 100, 50), and (100). The activation functions considered were tanh and relu, while the solvers evaluated were SGD and Adam. For the alpha parameter, values ranged uniformly between 0.0001 and 0.05. The learning rate was set to either constant or adaptive, and the maximum number of iterations tested were 200, 300, and 500.

In contrast, GaussianNB did not require hyperparameter tuning due to its simplicity and inherent effectiveness in handling continuous data.

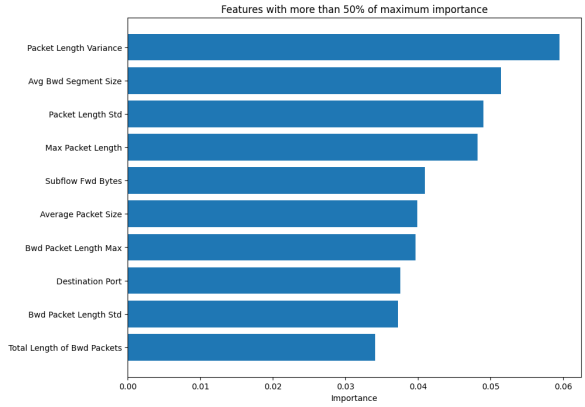
Finally, tests were conducted using the full dataset as well as datasets reduced to 10 and 20 features. For each scenario, the models were applied with the tuned hyperparameters, followed by a 3-fold cross-validation process with 5 iterations. 5 iterations were initially selected as a cost-effective approach that allows for an understanding of each model's functionality and performance. The best 5 combinations, based on performance measurements, were selected for further analysis. This methodology enabled a comparison of each algorithm’s performance across different scenarios, providing insights into how feature selection affects multiclass intrusion detection and identifying the algorithms that deliver the best results for this type of analysis. These experiments are expected to yield results that support drawing meaningful conclusions about the most effective algorithm among those tested.

Building on this, hyperparameter tuning and k-fold cross-validation were implemented to further refine the analysis. This procedure divided the selected dataset into training and

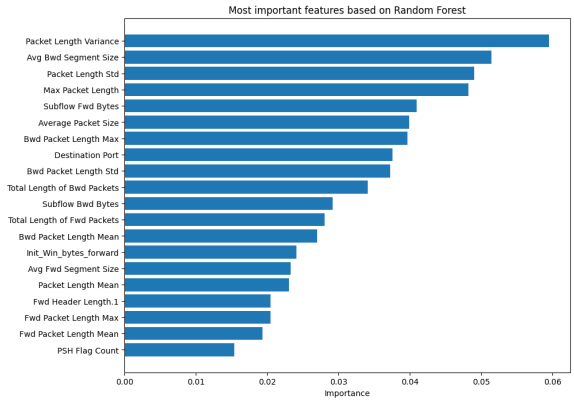
testing subsets according to the specified folds, with 80% of the data allocated for training and 20% for testing. This approach ensured that the models were trained on a substantial portion of the data and validated on the remaining subset to assess their performance. Following this step, the selected models—GaussianNB, Extended Trees Classifier, and MLPClassifier—were executed. Results for each scenario were then generated, specifically examining the performance of the models using 78 features, 20 features, and 10 features.

RESULTS AND DISCUSSION

As shown in Figure 3, a bar chart illustrates the importance levels of the features used to distinguish between benign data and attacks, as well as to identify the type of attack. The importance level is measured by how effective the features are at differentiating between anomalies and non-anomalies. Regarding the Random Forest Classifier, the importance of each feature is evaluated based on the ability to reduce impurity levels, thereby contributing to greater accuracy in detecting anomalies, identifying the types of anomalies, and distinguishing benign data.



(a) 10 most important features based on Random Forest



(b) 20 most important features based on Random Forest

Figure 3. Bar charts representing the top 20 and 10 most important features for attack detection in the database

Starting with the results obtained in the first scenario, where 78 features were used, Table 1 shows that one of the algorithms outperformed the others in the classification process based on the reference metrics: Accuracy, Recall, Precision, and F1-Score. Each model performs a multiclass classification to detect each type of attack. However, two of the four metrics, Recall and Precision, are calculated based on the model's ability to determine whether the detected data is benign or an attack. Consequently, the models group all attacks, and if they detect an attack, it is classified as an "attack" rather than its specific type. This effectively performs a binary classification of "normal" or "attack." It is also important to note that

MLPClassifier utilized probabilities to classify data between normal and various attack types, while Extended Trees, derived from decision tree models, employed a ranking process to identify attacks.

Table 1. Metrics for the Dataset with 78 features

Algorithm	Accuracy	Precision	Recall	F1-Score
Extended Trees	0.994082	0.994093	0.994082	0.993586
Naive Bayes	0.704162	0.971618	0.704162	0.806574
MLPClassifier	0.975917	0.976663	0.975917	0.975490

The machine learning algorithm that achieved the best results was the Extended Trees Classifier, attaining approximately 99% across all metrics used in the binary classification process. This classification focused on distinguishing between anomalies and benign data as an initial step. Although this algorithm is based on randomness rather than on nodes within a neural network or probabilities, it proved to be the most effective in classification. This demonstrates that the random selection of trees from data subsets yields better results compared to the other models. The second-best algorithm was the MLPClassifier, achieving results of approximately 97% across all metrics, indicating good performance in binary atack classification. However, it does not surpass the Extended Trees Classifier. Finally, GaussianNB, represented as Naive Bayes in the table, was the algorithm with the poorest performance based on the metrics. Its results showed greater variation, and although it reached 97% in Precision, this value is lower than that of the MLPClassifier, highlighting that this probabilistic model is not the most suitable for multiclass attack classification.

Continuing with the results obtained in the second scenario, where only the top 20 numerical features were used, Table 2 once again demonstrates the superiority of one machine learning algorithm over the others, evaluated using the same metrics as before.

Table 2. Metrics for the Dataset with 20 features

Algorithm	Accuracy	Precision	Recall	F1-Score
Extended Trees	0.987113	0.987822	0.987113	0.986522
Naive Bayes	0.532547	0.964787	0.532547	0.665886
MLPClassifier	0.950006	0.954186	0.950006	0.949139

In this case, where a reduced dimensionality of the dataset is used, the machine learning algorithm with the best performance remains the Extended Trees Classifier. This model achieved results of approximately 98% across all metrics, maintaining its superiority compared to the other models. However, a slight decrease in its performance across each metric is observed, which could be attributed to several factors, one of them being the reduced number of features used. This is because working with fewer features to distinguish between benign data and attacks reduces the specificity of the classification. This highlights the importance of having a larger number of features, as they enable a higher level of precision in data classification.

The second-best model continues to be the MLPClassifier; however, it also shows a decrease in its metrics, with results around 95% and 94%. This reduction is significant, as it indicates that the same factor affects not only the first algorithm but also the second. Finally, Naive Bayes once again ranks as the worst-performing model. A notable result in its metrics is its Accuracy, which drops to just 53%, compared to the 70% achieved when using all features. This demonstrates that as the number of features decreases, multiclass attack classification becomes less precise in a communication network with diverse characteristics.

Finally, when analyzing the third scenario, where only the top 10 numerical features are used, Table 3 shows that the same model highlighted in the previous scenarios once again demonstrates its superiority, evaluated using the same metrics as before.

Table 3. Metrics for the Dataset with 10 features

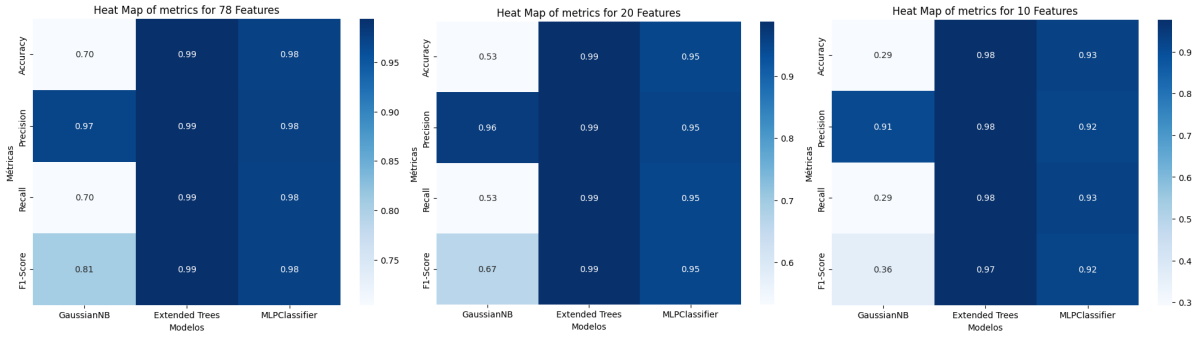
Algorithm	Accuracy	Precision	Recall	F1-Score
Extended Trees	0.976392	0.976892	0.976392	0.974495
Naive Bayes	0.293437	0.912316	0.293437	0.362544
MLPClassifier	0.931442	0.924971	0.931442	0.924322

In this final scenario, an even smaller number of features is used to analyze how dimensionality reduction influences the multiclass classification of anomalies in this dataset. Upon reviewing the results, the same performance order among the models is maintained, with the Extended Trees Classifier performing the best, achieving values of approximately 97%. The MLPClassifier ranks second, with results ranging between 92% and 93%, depending on the metric. Finally, Naive Bayes once again ranks as the worst-performing model, showing significantly low values, such as an accuracy of 29%.

Although the Extended Trees Classifier continues to achieve values above 95% across all metrics, a consistent decline in its results is observed. This suggests that the common factor in all three scenarios—namely, the reduction in features—is one of the most influential elements affecting metric outcomes. This trend is also reflected in the other models, some of which are more affected than others, but all experience a decline in performance as the number of features decreases.

This highlights that reducing the number of features may lead to the loss of key elements needed to distinguish between benign traffic and attacks. Therefore, this study not only identifies a model that outperforms the others but also determines a scenario with overall better performance. This underscores the importance of each feature when implementing a security layer using machine learning models, such as those selected in this study.

To facilitate the comparison between the three models and their respective metrics, heatmaps were created, providing a more visual representation of their performance. As shown in Figure 4, three heatmaps are presented, one for each scenario evaluated.



(c) 78 features (b) 20 features (a) 10 features

Figure 4. Heat Map for each scenario (78, 20, and 10 features) and comparison between models and metrics

Each of these heatmaps illustrates the results for each metric and model. The darker the blue color within the map, the higher the represented value. To improve visualization, the values are displayed with only two decimal places, reflecting the corresponding percentage. Despite this simplification, the heatmaps clearly highlight the differences between the results obtained for each metric across the models. Additionally, it is evident how, in each scenario, the metrics progressively decrease. Although the colors may appear similar, this is due to the adjustment to the new range of results obtained in each situation as a result of the reduced number of features.

Once these results were obtained, it became necessary to determine whether they occurred randomly or if the models genuinely produced significant results that demonstrate the superiority of one over another. For this reason, the Wilcoxon statistical test, was implemented. This analysis, also known as the Wilcoxon signed-rank test, is used to compare how distant two sets of values are from one another, with the objective of validating or rejecting the proposed hypothesis.

This test is non-parametric, meaning it analyzes data that do not necessarily follow a normal distribution. Furthermore, it can handle various types of data, whether normal, ordinal, or of other categories, as it does not rely on any prior distribution for its analysis. This flexibility makes it a suitable tool for evaluating any type of data [19][20][21]. In essence, this test is used to determine whether a hypothesis should be accepted or rejected. The formula employed in this statistical test is as follows:

$$W = \sum_{i=1}^{N_r} \left[sgn\left(x_{2,i} - x_{1,i}\right) \cdot R_i \right] \qquad (6)$$

The W represents the value of the statistical test. Then, a summation is included where N_r indicates the sample size, excluding the pair of data being evaluated. A sign function, represented as sgn, is used to determine the sign of the values of the data pairs. These pairs are represented by x_1 and x_2, which correspond to the rank values of each distribution and are compared to calculate the distance between them. Finally, R_i represents the rank evaluated in this statistical test.

The result of this formula provides the value of the distance between the data pairs, allowing an analysis of how separated or close these values are in order to assess the significance of the results. Next, it is necessary to calculate a p-value. This can be done using the W-value with the corresponding distribution or by employing the Z-value, which is obtained as follows:

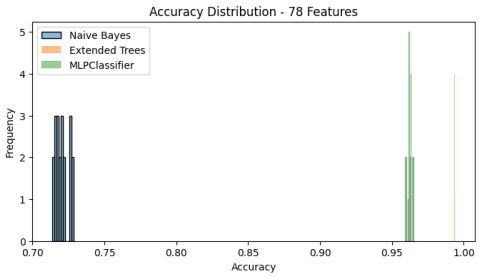
$$Z = \frac{W - \mu w}{\sigma w} \qquad (7)$$

The Z-value is obtained from the result of the Wilcoxon test by subtracting the mean of the W-value and dividing the result by its standard deviation. This calculation standardizes the W-value, facilitating the interpretation of the results in terms of statistical significance.

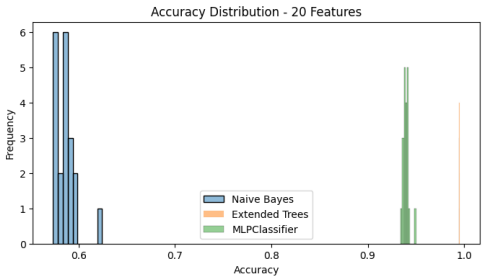
Once the Z-value is obtained, it is compared with the corresponding distribution to calculate the p-value, which determines whether the hypothesis should be accepted or rejected. If the p-value is less than 0.05, the results are considered statistically significant and unlikely to have occurred by chance. Conversely, if p is greater than or equal to 0.05, the results lack significance and could have been obtained randomly, leading to the rejection of the hypothesis.

In this work, the calculated results yielded a p-value close to 0. However, this value does not represent an exact 0; rather, it indicates that the distances between the results are so substantial that the p-value is extremely small. When calculated and rounded, it is expressed as 0.

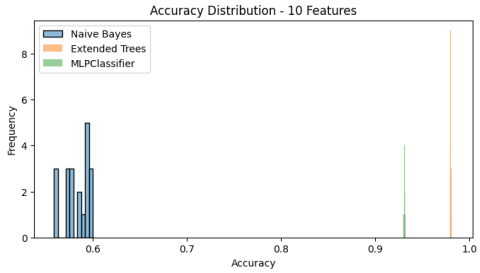
To confirm this, 20 samples were generated by applying the models and corresponding metrics. These samples were stored in an array and graphically represented in histograms, which can be seen in Figure 5.



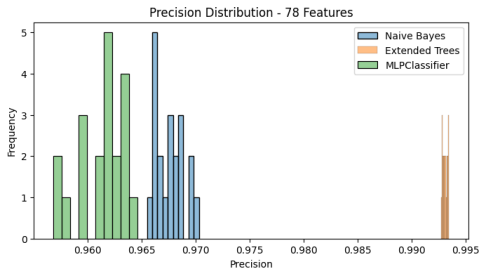
(a) Accuracy – 78 Features



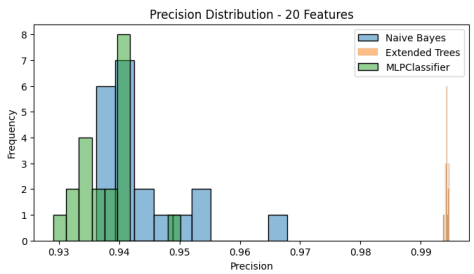
(b) Accuracy – 20 Features



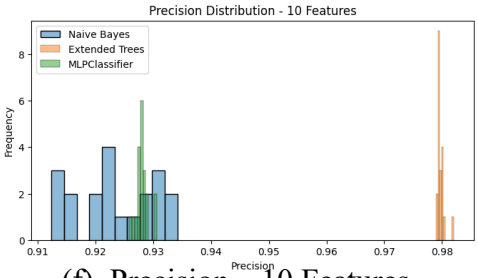
(c) Accuracy – 10 Features



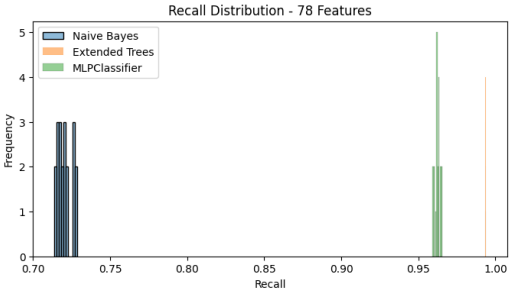
(d) Precision – 78 Features



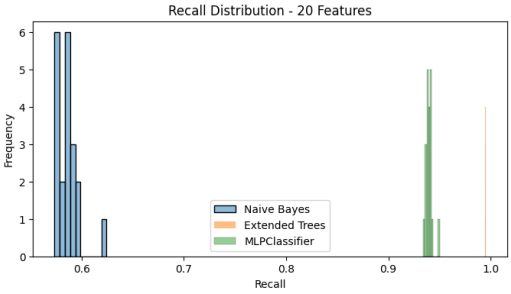
(e) Precision – 20 Features



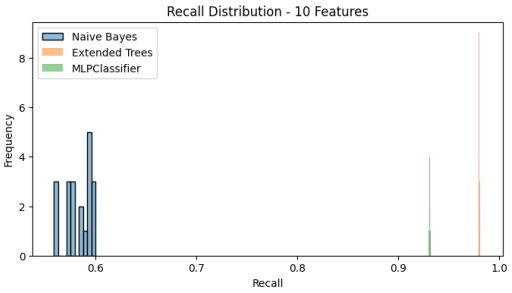
(f) Precision – 10 Features



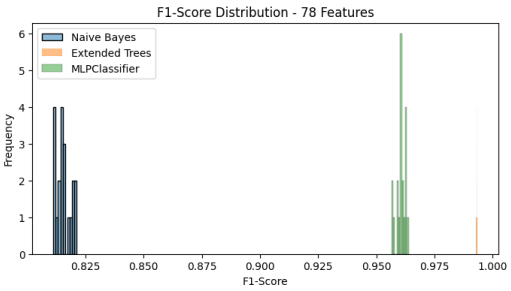
(g) Recall – 78 Features



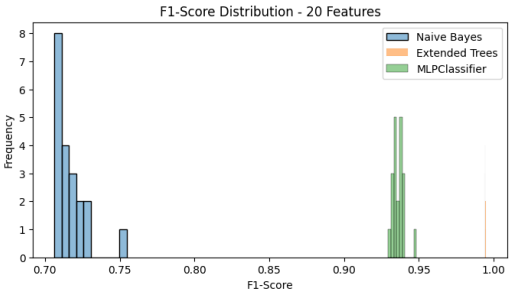
(h) Recall – 20 Features



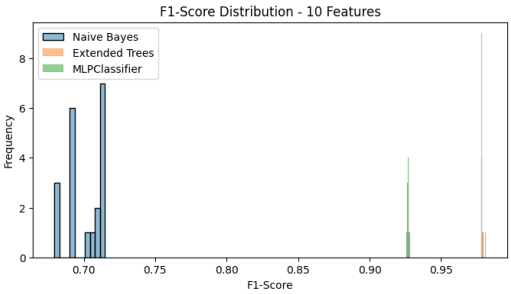
(i) Recall – 10 Features



(j) F1-Score – 78 Features



(k) F1-Score – 20 Features



(l) F1-Score – 10 Features

Figure 5. Histograms representing the distance between models for each metric in each scenario within a set of 20 samples. Justification for a p-value close to 0

Upon analyzing these graphs, the significant distance between the models was clearly observed, which justifies the p-value being so small that it rounds to 0. Subsequently, the

Wilcoxon statistical test was applied to these samples, yielding consistent results: p-values below 0.05. The highest p-value, averaged across the 20 samples, was 0.0296 for the Precision metric when comparing the GaussianNB and MLPClassifier models. Although this value is 0.0296, it remains below 0.05, reinforcing that the results were not obtained randomly and are statistically significant. This confirms that the p-value is less than 0.05, indicating that the results are statistically significant and not random. Furthermore, it demonstrates that the models were correctly trained and that there is a notable difference in their effectiveness. In particular, it validates that the Extended Trees Classifier is the best model among the three evaluated, making it a strong candidate for adding a layer of security to a communication network.

CONCLUSIONS AND FUTURE WORKS

Upon completing this work, a wide range of factors that influence the results and future work opportunities can be identified. As mentioned in the related work section, there was a great diversity of datasets that could have been used for this study. However, the CICIDS2017 database was chosen. This decision was based on how this dataset represents the most common types of attacks. Additionally, being a communication network, it allowed the project to achieve its objective: determining the most effective machine learning model for detecting potential attacks in a typical environment.

All of this was done with the goal of finding the best possible solution to create a new layer of security that benefits people living in a hyperconnected world, where much of our activities rely on tools that utilize the Internet.

Furthermore, as discussed in the related work section, different machine learning models were selected for comparison.

To measure the efficiency of these three models, a classification method was necessary. For this reason, comparison metrics were chosen to evaluate and determine the best performance for each model. As mentioned in the related work and proposal sections, the metrics selected to assess the models' effectiveness were Accuracy, Precision, Recall, and F1-Score. These metrics evaluate the accuracy of the models from different perspectives, providing a more comprehensive view of their performance. This is important, as relying on only a subset of these metrics could introduce limitations when determining how effective a model truly is.

Based on these three models and the results obtained for each metric, it was established that the best model is the Extended Trees Classifier. This is not only due to its high scores, which demonstrate its efficiency, but also because, even when the number of features used is

reduced, the decline in metric performance is not as significant as it is with the other two models. This represents a promising solution to the problem of attacks, as it is a machine learning model trained to detect anomalies and can be implemented as an additional security layer, offering better protection for users.

As this project has the potential to expand and improve, there are numerous elements that could be implemented to modify, extend, or explore new objectives in the search for solutions. As technology continues to evolve, the availability and variety of datasets, models, and metrics will also increase and adapt, creating opportunities to employ new approaches. These could include exploring other types of Decision Trees, DBScan, SVM, or comparing one-class classification with multi-class classification. Additionally, methods like Isolation Forest or Local Outlier Factor could be considered. New metrics could also be introduced, such as the time it takes to run the models, loss graphs comparing training and validation results, or precision-recall curves for a more visual evaluation of model performance. Furthermore, combining datasets could allow for more exhaustive and comprehensive research.

This could help identify models that offer a more robust and effective security layer. Ultimately, this project is not intended to be limited solely to the elements used but aims to lay the groundwork for a research area that integrates data analysis, communication networks, machine learning models, security, and other disciplines. Its purpose is to expand knowledge and improve the tools people use daily, driving positive technological change.

Moreover, it seeks to ensure a secure environment where information is protected and where the levels of cyberattacks are controlled and, as much as possible, reduced. At the same time, it strives to continuously improve protection strategies against such attacks, which are not always predictable or avoidable.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the Department of Computer Science at USFQ for providing the necessary resources (A100 computer and server) to execute the code developed for the presented project.

REFERENCES

[1] J. A. Gamboa, J. C. Arroyave, and E. A. Unamuno, “Detección de ataque de ddos utilizando machine learning – algoritmo de random forest,” Artículo Original, 2022.

[2] D. Álvarez Polo, “Detección de ataques de intrusión con algoritmos de machine learning,” Tesis presentada como requisito parcial para optar al título de Ingeniero de Sistemas y Computación, Universidad de los Andes, 2023, [Online]. Available: https://repositorio.uniandes.edu.co/server/api/core/bitstreams/de635c47-f374-41dc-b4f0-ac5dc0b2cc97/content.

[3] J. M. R. Rama, “Aplicación de ataques de intrusión de machine learning a la detecci´on de ataques,” Trabajo Fin de M´aster, 2018, [Online]. Available: https://openaccess.uoc.edu/bitstream/10609/81126/11/jmrodriguez85TFM0618memoria.pdf.

[4] A. Kumar, W. Glisson, and R. Benton, “Network attack detection using an unsupervised machine learning algorithm,” Aisel, 2020, [Online]. Available: https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1762&context=hicss-53.

[5] F. S. D. L. Filho, F. Silveira, A. D. M. B. Junior, G. Vargas-Solar, and L. Silveira, “Smart detection: An online approach for dos/ddos attack detection using machine learning,” Security and Communication Networks, vol. 2019, p. 1–15, Oct. 2019.

[6] R. F. Moyano et al., “Standard latent space dimension for network intrusion detection systems datasets,” Research Article, 2023.

[7] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy, vol. 1, Jan. 2018, p. 108–116.

[8] “Iso/iec ts 4213:2022(en), information technology — artificial intelligence — assessment of machine learning classification performance,” 2022, [Online]. Available: https://www.iso.org/obp/ui/en/#iso:std:iso-iec:ts:4213:ed-1:v1:en.

[9] “Iso/iec 22989:2022(en), information technology — artificial intelligence — artificial intelligence concepts and terminology,” 2022, [Online]. Available: https://www.iso.org/obp/ui/en/#iso:std:iso-iec:22989:ed-1:v1:en:sec:5.17.

[10] S. Mehrotra. (2023, Jun.) Unleashing the power of extended isolation forest: An anomaly detection breakthrough. [Online]. Available: https://medium.com/@sukritimehrotra/unleashing-the-power-of-extended-isolation-forest-an-anomaly-detection-breakthrough-19d36aad8681.

[11] S. Enslin. (2022, Jan.) The complete guide to neural network multi-class classification from scratch. [Online]. Available: https://towardsdatascience.com/the-complete-guide-to-neural-networks-multinomial-classification-4fe88bde7839.

[12] T. Omoniyi. (2024, May) Distilbert for multiclass text classification using transformers. [Online]. Available: https://medium.com/@kiddojazz/distilbert-for-multiclass-text-classification-using-transformers-d6374e6678ba.

[13] S. Neogi. (2021, Dec.) Exploring multi-class classification using deep learning. [Online]. Available: https://medium.com/@srijaneogi31/exploring-multi-class-classification-using-deep-learning-cd3134290887.

[14] "Is accuracy a good measure of model performance? — fiddler ai," 2023, [Online]. Available: https://www.fiddler.ai/model-accuracy-vs-model-performance/is-accuracy-a-good-measure-of-model-performance.

[15] T. Ige and C. Kiekintveld, "Performance comparison and imple-mentation of bayesian variants for network intrusion detection," arXiv, Jan. 2023.

[16] Sanki. (2023, Jan.) Feature selection methods - sanki - medium. [Online]. Available: https://medium.com/@sanket.ai/feature-selection-methods-4f14dd6c4087.

[17] Kashishdafe. (2024, Mar.) Gaussian naive bayes: Understanding the basics and applications. [Online]. Available: https://medium.com/@kashishdafe0410/gaussian-naive-bayes-understanding-the-basics-and-applications-52098087b963.

[18] A. Nair. (2024, Aug.) Mlp classifier – a beginner's guide to sklearn mlp classifier. [Online]. Available: https://analyticsindiamag.com/ai-mysteries/a-beginners-guide-to-scikit-learns-mlpclassifier/.

[19] J. W. Pratt, "Remarks on zeros and ties in the Wilcoxon signed rank procedures," Journal of the American Statistical Association, vol. 54, no. 287, pp. 655–667, 1959. [Online].Available: http://www.jstor.org/stable/2282543

[20] DATAtab, "T-Test, Chi-Square, ANOVA, Regression, Correlation..." 2024. [Online]. Available: https://datatab.net/tutorial/wilcoxon-test

[21] R. J. Freund, W. J. Wilson, and D. L. Mohr, "Nonparametric methods," in Elsevier eBooks. Elsevier, 2010, pp. 689–719.

[22] "Cyber attacks — cyber threat solutions — core security." [Online]. Available: https://www.coresecurity.com/cyber-attacks

[23] D. Schrader. (2024, oct) How to prevent cyber attacks: Strategies and best practices. [Online]. Available: https://blog.netwrix.com/how-to-prevent-cyber-attacks