SCIENTIFIC CULTURE

www.sci-cult.com

SCIENTIFIC CULTURE, Vol. 12, No. 1.1, (2026), pp. 2979-2991
Open Access. Online & Print

DOI: 10.5281/zenodo.121126217

DATA MINING FOR THE EARLY DETECTION OF CYBERATTACKS ON ENTERPRISE NETWORKS

Brandon Morales[1]*, Enmanuel Panduro[2], Dikxon Luque[3], Teodoro Andrade4, Carlos Chiri[4]

Universidad San Ignacio de Loyola (USIL), Perú
[1]Universidad Nacional Mayor de San Marcos (UNMSM), Perú
[2]Universidad Nacional Mayor de San Marcos (UNMSM), Perú
[3]Universidad Nacional Mayor de San Marcos (UNMSM), Perú
[4]Universidad Nacional Mayor de San Marcos (UNMSM), Perú

Received: 15/09/2025
Accepted: 10/01/2026

Corresponding Author: Brandon Morales (brandon.moralesf@usil.pe)

ABSTRACT

The early detection of cyberattacks is critical to protecting enterprise networks. This paper proposes a method that uses data mining and machine learning techniques to identify harmful traffic on computer networks. The UNSW-NB15 dataset was used as a reference for testing this method. The CRISP-DM methodology was applied, ranging from understanding the data to evaluating the model. Univariate and bivariate exploratory analyses were carried out to select relevant characteristics for the study. Joint learning algorithms, such as Random Forest, Extra Trees, AdaBoost, and XGBoost, were used. Results show that models using the bagging method, particularly Random Forest, perform much better than boosting-based models in metrics such as accuracy (0.98), recall (0.99), and F1-score (0.98) in the attack category. It is concluded that collective learning approaches are an effective, understandable, and low-computational-cost alternative for automatically detecting intrusions in corporate networks. This study highlights the feasibility of integrating robust data analytics approaches into advanced cybersecurity systems.

KEYWORDS: Deep Learning, computer vision, artificial intelligence, data mining, intrusion detection.

1. INTRODUCTION

Over the years, an increasing number of sectors have adopted digital platforms for customer service and connection. This has made cybersecurity a fundamental element of computer networks. Hackers continuously look for ways to breach systems to steal valuable information and misuse it for profit. Without proper measures in place, such attacks can compromise the integrity and image of companies (Biswas, 2018; Boutaba et al., 2018). Concerns about cybersecurity are growing and are driving the development of robust intrusion detection systems that can learn from data and apply it in real-time situations. These systems are essential for safeguarding networks against harmful connections (Wang, 2019). They identify and report irregular behavior on the network.

Intrusion identification systems (IDSs) fall into two categories: misuse-based and anomaly-based (Aydin, Zaim, & Ceylan, 2009). IDSs that rely on misuse compare attack signatures from new connections within a network to previously recorded attack signatures to determine if the new connection is harmful. However, these types of IDSs are only effective in identifying known attacks. Additionally, the emergence of novel "zero-day" attacks, which have not been observed previously (Boutaba et al., 2018), poses a risk. In this scenario, anomaly-based intrusion detection systems come into operation. The fundamental concept of an anomaly-oriented IDS is to examine normal network behavior and report deviations from anticipated behavior. Anomaly IDSs can be developed using various machine learning techniques. There are multiple available datasets that can be useful for analyzing network behavior. Intrusion detection systems are based on machine learning. However, twenty years after its release, it has become obsolete. Disadvantages of this dataset include its age, notable asymmetry in the target variables, and redundancy. The KDD'99 dataset has served as a reference for developing a system.

In this study, the UNSW-NB15 dataset (Description of the UNSW-NB15 Dataset, 2018) is used. This dataset is an optimized version of the KDD Cup dataset. This dataset enables the examination of network properties under normal conditions and in attack situations. The UNSW-NB15 dataset is more current and balanced, and it is becoming the new benchmark for developing effective intrusion detection models. A significant issue with the KDD'99 dataset was the class imbalance, as the number of samples for the U2R and R2L attack categories was much lower than for other targets. This imbalance negatively affects the classifier's effectiveness. The UNSW-NB15 dataset corrects the limitations of the KDD'99 dataset by offering a better balance between classes and less redundancy.

The UNSW-NB15 dataset contains ten classes: Normal, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shell Code, and Worms. In this article, the binary version of the dataset is considered, where 0 corresponds to the normal class and 1 corresponds to the aggregate attack class [3]. The rest of the article is organized as follows: Section 2 presents the existing literature on the topic. This section summarizes some of the best posts. Section 3 documents the work done on the UNSW-NB15 dataset. The subsections describe the dataset, the required data preprocessing techniques, and the exploratory data analysis performed with state-of-the-art graphs. Additionally, the results of implementing various co-learning algorithms in the dataset are shown in table form. Finally, Section 4 concludes the article with final thoughts and future discussion topics.

State of the Art

Among the reviewed studies, there are classic contributions and mixed alternatives. One study examined the automation of intrusion detection systems using machine learning. This study explains how machine learning can detect anomalies in a network. Generally, tags are binary: 0 for normal and 1 for an attack. Some datasets classify attacks into subcategories. However, this method is expensive, time-consuming, and less accurate. The study also proposed several alternative models: Bayesian models, artificial neural networks (ANNs), Markov models, fuzzy clustering, and k-means clustering (Wagh, Pachghare, & Kolhe, 2013). These approaches are complemented by preprocessing, training/testing, and evaluation techniques using confusion matrices.

In their study of intrusion detection in work networks based on machine learning, Vipin Das and his team found that they applied the theory of approximate sets (RST) together with SVM. This approach achieved 98.7% accuracy with 29 characteristics selected from the original 41 in the KDD set (Das et al., 2010). This vastly outperformed principal component analysis (PCA), which selected 27 characteristics and achieved only 84.32% accuracy, highlighting the importance of accurate feature selection.

In another study, Bhati et al. (2020) used a subspace discriminant classifier to improve the accuracy of the minority classes R2L and U2R of the KDD set. They achieved accuracies of 99.71% and 99.58%, respectively, which is higher than 97.8%. These results demonstrate the impact of well-

targeted, comprehensive approaches.

In his study, Othman describes an intrusion detection model that uses machine learning. He found that implementing the Spark-Chi-SVM model, which combines ChiSqSelector reduction and SVM classification with SGD, reached an AUROC of 99.55%. This result is superior to logistic regression (92.77%) and simple SVM (94.36%). This validates the power of Spark for computational optimization (Othman et al., 2018).

In their study on intrusion detection using hybrid data, Ren et al. proposed DO_IDS, a hybrid system that cleans data with iForest and optimizes it with AG and Random Forest. DO_IDS was evaluated on UNSW-NB15 and achieved an accuracy of 92.8%. The authors recommended improving training time as a future goal.

In their study on instruction detection models in cybersecurity, Sarker and his team developed IntruDtree, a tree-based classifier optimized for various characteristics. IntruDtree obtained 98% accuracy, surpassing SVM, KNN, and logistic regression. It proved to be robust against undetected data (Sarker et al., 2020).

Models based on deep learning have gained ground as an effective solution to the complexity and volume of cybersecurity data (Banoori & Hegde, 2023). These models feature a hybrid convolutional neural network (CNN)-recurrent neural network (RNN) architecture that achieves over 99% accuracy in the NSL-KDD and CICIDS2017 datasets. This highlights their ability to combine spatial and sequential analysis (Wang et al., 2023). Additionally, they propose a multi-branched CNN model that integrates a hybrid feature selection approach, achieving 99.2% accuracy and efficiency in classifying malicious traffic. Similarly, Zhang et al. (2019) used autoencoders for unsupervised detection, reducing the need for tags and facilitating the identification of new threats. While these approaches outperform classical models in accuracy and sensitivity, they require more computing power and produce more difficult-to-interpret results, which can limit their practical adoption.

Other studies on intrusion detection continue to adopt traditional models, such as classic machine learning models: SVM, Random Forest, and Naive Bayes, due to their interpretability and lower cost of computer equipment (Alghamdi et al., 2022). G. A. Cordero et al. (2019) achieved an accuracy of 97.68% by combining multi-agent systems with Random Forest. Arora and Kalia (2023) demonstrate that AdaBoost is competitive in trusted network environments with 96.75% accuracy and a low false positive rate (Meenal Gaur et al., 2020). However, these models usually require intensive preprocessing and manual feature selection, and they are sensitive to imbalanced data and new attack classes.

Integrating multiple paradigms has produced more robust solutions, such as hybrid or multi-agent models. For example, the study led by Syeda Mehak Raza implemented DL-CAD, combining multistage feature selection (Chi2 + PSO) with deep neural networks and achieving 99.69% accuracy in CICIDS2017 (Syeda Mehak Raza et al., 2023). Similarly, another study by Norah Abdullah Alghamdi allows for the detection of known and unknown attacks through cooperative agent and classifier structures (Alghamdi et al., 2020). These models' key advantage is their robustness against diverse scenarios; however, they require extensive validation and careful design of interoperability between modules.

Study development

Objective of the study and understanding of the business

This study aims to enhance the automatic detection and classification of cyberattacks using the UNSW-NB15 dataset by employing: 1) exploratory data analysis (EDA) to understand the behavior of dataset attributes, and 2) ensemble learning techniques to optimize the performance of the classification model.

We will also examine the machine learning concepts employed in cybersecurity. Using the UNSW-NB15 dataset (Verma and Ranga, 2017), exploratory data analysis will be performed and ensemble learning algorithms will be implemented. The EDA process uses data visualization techniques to analyze and find patterns in the data, providing a better understanding of its characteristics. EDA collects information about the distribution of the data, the correlation between features, and the anomalies present in the dataset. This information will enable more informed decisions regarding which features to include in the set during training and which relevant attributes to extract to improve model accuracy.

Defining the Model

The proposal's integral development is presented herein, constituting the primary focal point of this research. The CRISP-DM (Cross Industry Standard Process for Data Mining) methodology is employed, encompassing the subsequent stages:

Comprehension of the business domain.

Data compression.

Data preparation.

Modeling.

Evaluation.

Deployment. The methodology enables the

structured development of data-driven systems in an orderly and replicable manner.

Figure 1. CRISP-DM Methodology

Understanding the data

The dataset employed in this study is referred to as the UNSW-NB15. This set is a contemporary and superior alternative to the traditional KDD cup dataset, as the latter has been rendered obsolete and is rife with flaws. The UNSW-NB15 demonstrates superior balance between classes in intrusion identification when compared to other datasets. Despite its lower volume compared to several other datasets, its level of redundancy is lower, making it suitable for training a model with high effectiveness. The set encompasses ten categories: one representing normal conditions and nine corresponding to various types of attacks. Additionally, a binary version of this dataset exists, where 0 indicates normal conditions and 1 signifies an attack. In the present study, the analysis will be constrained to the binary version of this set. The set contains a total of 49 attributes, a factor that facilitates precise classification of classes. The quantity of training data is 175,341, whereas the test data totals 82,332 (Description of the UNSW-NB15 dataset, 2018).

Data preparation

Data preprocessing

Data preprocessing is machine learning's first step in model building. The actual data is inconsistent, so preprocessing prepares the disorganized data for visualization. Here are the findings we obtained by applying data preprocessing techniques:

- Data cleansing. No missing values were found in the dataset. Most of the features are continuous and can be useful for classification.
- Data transformation. After reviewing the description of each characteristic, the data has a high degree of bias and requires normalization before visualization. Some features were categorical, so we coded them so that we could include them during training. While the coded variables did not contribute to prediction accuracy and only increased the dimensionality of the dataset, we considered removing these categorical features from the training data.
- Data reduction. We removed outliers from the dataset before moving on to data visualization to better understand the distribution of features. However, for the final training, we included the 175,341 samples of the 33 selected characteristics.

To learn more about each of the variables in the dataset, see (NUSW_NB15 features. csv).

Modelling

Data Visualization

Count plots were produced to illustrate the number of regular and attack samples in the dataset. For all graphical visualizations, the seaborn library in Python was used. Figure 2 shows the count graph, which makes it easy to visualize the total number of regular and attack samples in the dataset (0 represents the regular class and 1 represents the attack class).

Number of attack samples = 119341

Number of Normal Samples = 56000

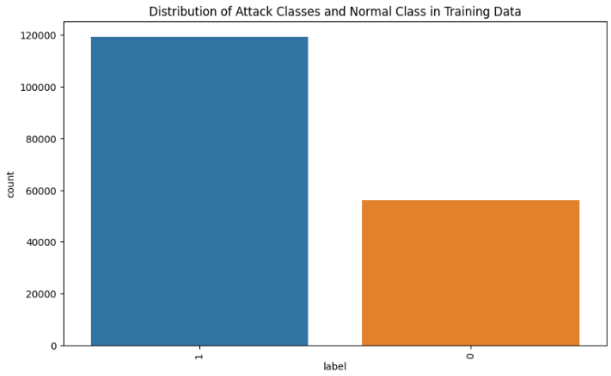


Figure 2. Counting Graph

Univariate Analysis

The process entails a meticulous examination of each feature within the dataset to identify potential irregularities, as well as any existing asymmetries and other pertinent data. This comprehensive approach is instrumental in selecting the most suitable characteristics for the training process (Doshi, 2019). The creation of box plots for each attribute will allow for the identification of numerous anomalous values in the dataset. Consequently, we address these outliers and reintroduce the box plots. This facilitated a more profound comprehension of the attributes and their pertinence to the ranking. A box plot is a graphical

representation of the median, represented by the horizontal line on the box, minimum and maximum values, indicated by whisker lines, and anomalous values, marked by dark dots.

As illustrated in the following figure, the box plots represent the most relevant features. The horizontal axis of the figure corresponds to the features, while the vertical axis shows the corresponding normalized values.

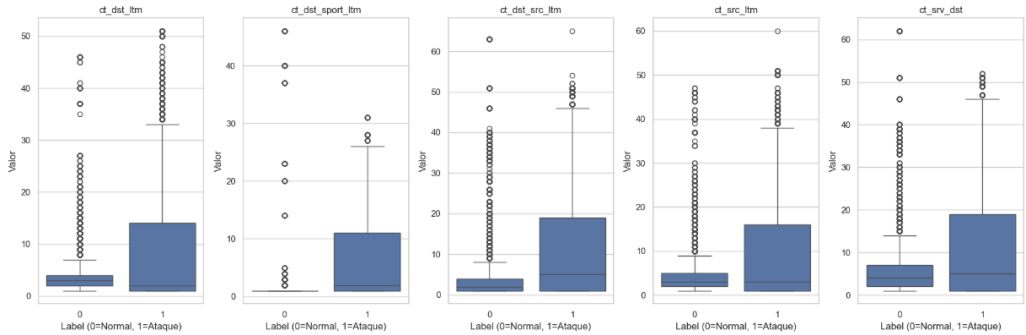


Figure 3. Box and whisker diagram for meaningful values Dataset Features

The graph suggests that the median value for the characteristic ct_srv_dst is considerably higher for the normal class compared to the attack class. Furthermore, the range of values observed for the normal class exceeds that of the attack class. In order to facilitate the process of sorting, it is imperative that the values of both labels and classes are situated within distinct ranges. It is evident that the outliers for the characteristic of the ... ct_dst_sport_itm are considerably less pronounced in comparison to other features. It is generally advantageous to maintain a minimal number of outliers. The feature ct_dst_sport_itm has been demonstrated to be advantageous for ranking purposes, as the values for the normal class are largely consistent, with only a few outliers. In contrast, the values for the attack class are distributed over a different range.

A collaboration has been initiated with Violin Plots to investigate the distribution and probability density of the data. A violin chart is a combination of a box chart and a density chart (Doshi, 2019). Violin charts provide a visual representation of the distribution of data for each label, as well as its density. The following chart delineates the salient features of the violin.

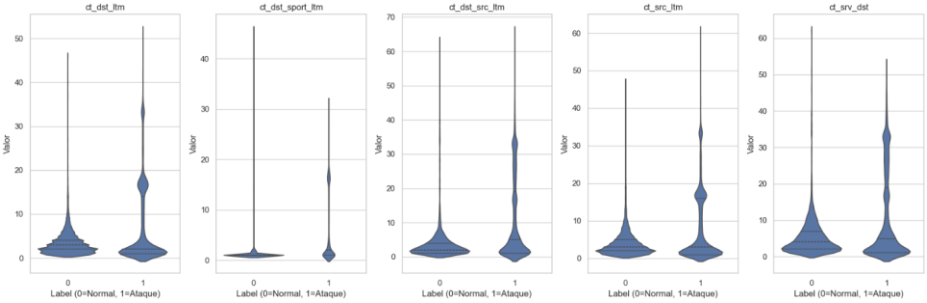


Figure 4. Violin chart for significant features of the Dataset

As illustrated in Figure 4, the density distribution of each label is found to be virtually identical for each characteristic, a conclusion that can be drawn from the inverted density graphs. The distribution of data is analogous to that depicted by box plots.

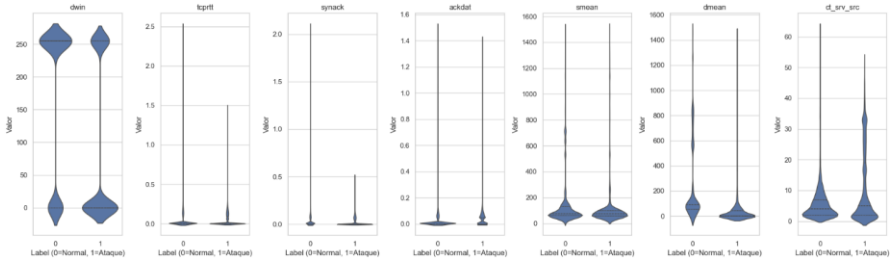


Figure 5. Violin chart for some similar features in the Dataset

As illustrated in Figure 5, the violin graphs for the synack, tcprtt, and confirmed features exhibit a high degree of similarity, suggesting a strong correlation between the features. It is preferable for there to be a lower correlation between them.

The Point Plots for each feature have been plotted against the range of values for all features after scaling. The point plots provide insight into the change in the average values for both labels across the different features. The dot plot is a graphical representation of an estimate of the central tendency of a numerical variable. This estimate is determined by the position of the dots on the scatter plot. The uncertainty surrounding the estimate is indicated by error bars. The dot plot displays the mean values and the standard deviation of the data (doshi, 2019). As illustrated in Figure 6, the values of the various characteristics of the normal class exhibit significant variability. In contrast, the values of the attack class demonstrate minimal variation, with a tendency to approximate zero. This observation suggests that the values of these characteristics reside in distinct ranges.

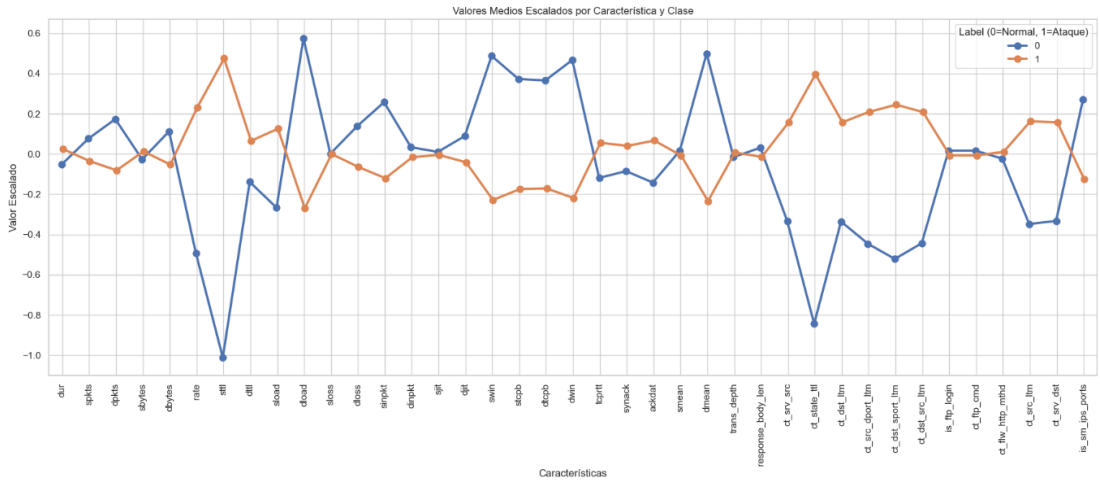


Figure 6. Point plots for all relevant features in the Dataset

Bivariate Analysis

Bivariate analysis is defined as the analysis of one characteristic in relation to another. This type of analysis facilitates the identification of correlated features and enables the examination of the distribution of data from one feature to another.

A heat map has been plotted for all the relevant features of the dataset, and several interesting observations have been obtained (Chapaneri, 2019). The value 1 indicates a high correlation, while -1 signifies a negative correlation. A value of 0 indicates that the two characteristics exhibit the lowest degree of correlation.

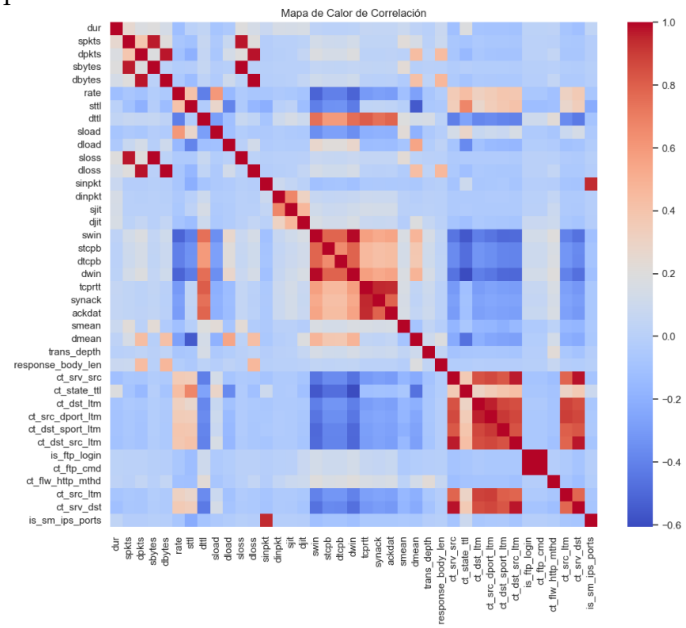


Figure 7. Heatmap for all functions

This study employs a rigorous, systematic approach, meticulously analyzing the interplay between characteristics and the distribution of values. As illustrated in Figure 8, the joint graphs of two features with positive correlation demonstrate a linear distribution of the values of the confirmed tcprtty features, thereby validating our understanding of violin graphs. This suggests that the confirmed tcprtty features exhibit a correlation that can be discerned simply by observing the graph shape. Furthermore, the heat map reveals a correlation value of 1 for tcprtt and ackdat, indicating a positive correlation between these two features.

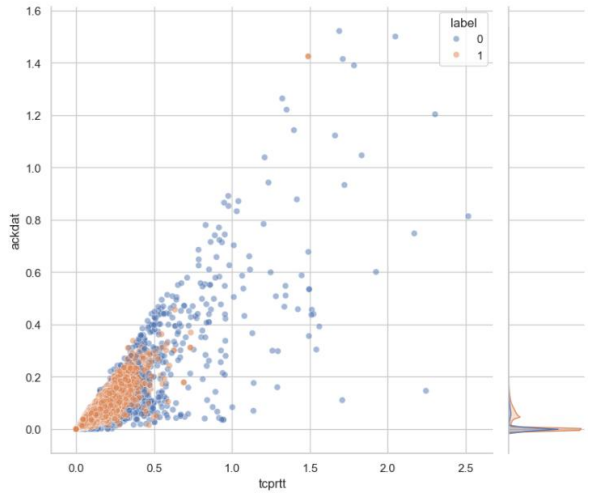


Figure 8. Joint tracing between tcprtt and ackdat features

Following a thorough examination of the data through exploratory analysis, the characteristics exhibiting the highest number of anomalies are identified and eliminated, such as a high number of outliers or data with significant skewness. It has been determined that these features will not be useful for the classification process.

	Característica	Estado	Prioridad
0	dur	Seleccionada	1
1	spkts	Seleccionada	2
2	dpkts	Seleccionada	3
3	sbytes	Seleccionada	4
4	dbytes	Seleccionada	5
5	rate	Seleccionada	6
6	sttl	Seleccionada	7
7	dttl	Seleccionada	8
8	sload	Seleccionada	9
9	dload	Seleccionada	10
10	sloss	Seleccionada	11
11	dloss	Seleccionada	12
12	sinpkt	Seleccionada	13
13	dinpkt	Seleccionada	14
14	sjit	Seleccionada	15
15	djit	Seleccionada	16
16	swin	Seleccionada	17
17	stcpb	Seleccionada	18
18	dtcpb	Seleccionada	19
19	dwin	Seleccionada	20
20	tcprtt	Seleccionada	21
21	synack	Seleccionada	22
22	ackdat	Seleccionada	23
23	smean	Seleccionada	24
24	dmean	Seleccionada	25
25	ct_srv_src	Seleccionada	26
26	ct_state_ttl	Seleccionada	27
27	ct_dst_ltm	Seleccionada	28
28	ct_src_dport_ltm	Seleccionada	29
29	ct_dst_sport_ltm	Seleccionada	30
30	ct_dst_src_ltm	Seleccionada	31
31	ct_src_ltm	Seleccionada	32
32	ct_srv_dst	Seleccionada	33
33	id	Eliminada	-
34	proto	Eliminada	-
35	service	Eliminada	-
36	state	Eliminada	-
37	trans_depth	Eliminada	-
38	response_body_len	Eliminada	-
39	ct_flw_http_mthd	Eliminada	-
40	is_ftp_login	Eliminada	-
41	ct_ftp_cmd	Eliminada	-
42	attack_cat	Eliminada	-
43	label	Eliminada	-
44	is_sm_ips_ports	Eliminada	-
45	srcip	Eliminada	-
46	sport	Eliminada	-
47	ct_src_ ltm	Eliminada	-

Figure 9. Prioritize Features in Order

According to Figure 9, included 33 characteristics out of a total of 49 in the final training in order of priority. These characteristics are: 'dur', 'spkts', 'dpkts', 'sbytes', 'dbytes', 'rate', 'sttl', 'dttl', 'sload', 'dload', 'sloss', 'dloss', 'sinpkt', 'dinpkt', 'sjit', 'djit', 'swin', 'stcpb', 'dtcpb', 'dwin', 'tcprtt', 'synack', 'ackdat', 'smean', 'dmean', 'ct_srv_src', 'ct_state_ttl', 'ct_dst_ltm', 'ct_src_dport_ltm', 'ct_dst_sport_ltm', 'ct_dst_src_ltm', 'ct_src_ltm', 'ct_srv_dst'.

Evaluation

Implementation and evaluation of set algorithms

Four of the most robust co-learning algorithms have been implemented in the dataset, and the accuracy, precision score, retrieval, and f1 score of each of these classification models have been recorded. These evaluation parameters are derived from the confounding matrix, which quantifies true positives, true negatives, false positives, and false negatives of a ranking algorithm calculated on a validation dataset [4] (Liu and Lang, 2019). As illustrated in Table 2, the representation of a confusion matrix is presented.

Table 2. Representing a Confusion Matrix

Real/predicho	Negativo	Positivo
Negative	True Negative (TN)	False Positive (FP)
Positive	False Negative (FN)	True Positive (TP)

The formulas for the evaluation parameters used are: (ghoneim, 2019)

Acurracy: Indicates how many samples are truly positive among the total predictive positives.

Acurracy = TP/(TP+FP)

Recall=Indicates how many samples are truly positive among the actual positive samples

Recall = TP/(TP+FN)

F1-score = Can provide better insight than accuracy in

The case of class imbalance.

F1-score = 2*[(precision*recall) /(precision + recall)]

1. What is Ensemble Learning and the types of ensemble methods?

In the context of joint learning, smaller models are aggregated to construct a master model. Upon the entry of a new data point, each base model generates a prediction for the input tuple. The class that garners the most votes is designated as the output of the master or aggregate model. It has been demonstrated that base models are capable of utilizing any classification algorithm to make predictions (Gupta and Rani, 2020).

Machine learning set methods can be categorized into two distinct types: The process of bagging and boosting is a critical component of the overall strategy. Bagging is a process that involves generating multiple base classifiers in parallel to reduce the variance of the estimate. Ultimately, the predictions are combined in a weighted manner, with greater weight assigned to models that demonstrate higher accuracy than the most accurate model. In this method, multiple subsets of the original dataset are formed by random sampling. Each of these datasets is then fed into a machine learning model, known as a "classifier," which is trained to make predictions. The mean of these predictions is then considered in regression problems or in the maximum vote count in the case of classification models, and aggregated as a result of the master model.

The process of boosting involves the generation of successive models, with each subsequent model aiming to rectify the deficiencies identified by its predecessor. Examples that have been misclassified in previous rounds receive a higher weighting. The final step in the model's development process involves the aggregation of the predictions from each base classifier. These predictions are then combined through a weighted majority vote or a weighted sum, resulting in a final prediction. This prediction is then utilized as the model outside the set.

Brief explanation of the set algorithms used

Random forest : The Random Forest follows the Bagging aggregation approach to form the ensemble model. The original data subsets are formed by randomly selecting samples from the original data, with replacement. The only difference between bagging and Random Forest is that decision trees are formed randomly to minimize the correlation between them.

Extra Trees: The extra trees, also known as extremely random trees, are similar to the random forest; the difference is that the extra trees use the entire training set to train the base classifiers instead of subsets of the original dataset and are split for each base DT; they are chosen randomly rather than greedily chosen.

AdaBoost: The AdaBoost classifier follows the reinforcement set method. A sequence of classifiers is formed where greater weight is given to the instances more difficult to classify.

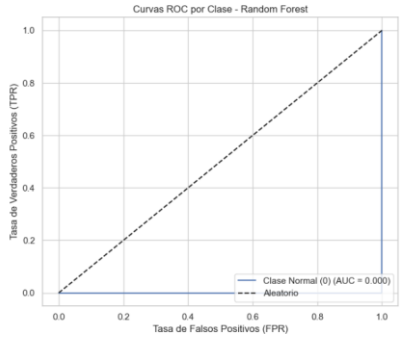
XGBoost: XGBoost is short for extreme gradient potentiation and is an implementation of gradient-powered trees. Improves the execution speed and performance of a model.

Table 3. List of evaluation parameters for each model

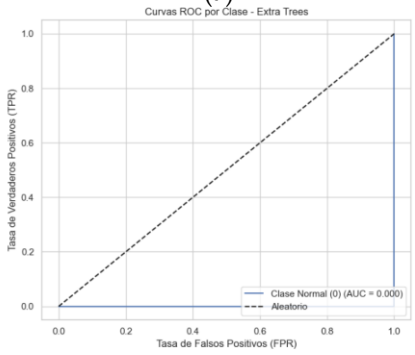
Metric	Class	Random Forest	Extra Trees	AdaBoost	XGBoost
Accuracy	Normal	0.98	0.97	0.93	0.95
Precision	Normal	0.99	0.98	0.91	0.94
Recall	Normal	0.96	0.95	0.88	0.92
F1-Measure	Normal	0.97	0.96	0.89	0.93
Accuracy	Attack	0.98	0.97	0.93	0.95
Precision	Attack	0.97	0.96	0.90	0.93
Recall	Attack	0.99	0.98	0.87	0.91
F1-Measure	Attack	0.98	0.97	0.88	0.92

As illustrated in Table 3, the Random Forest classifier and the Extra Tree classifier produce nearly identical outcomes, and the AdaBoost and XGBoost booster algorithms also yield comparable results. The Bagging sorter has been demonstrated to enhance the efficiency of sorting labels. The Random Forest sorter has been demonstrated to achieve the highest level of accuracy, with a recorded percentage of 0.98%. As illustrated in Figure 10, the ROC curves of the Random Forest and Extra Tree classifiers are nearly indistinguishable, a similarity that is also evident between the ROC curves of the AdaBoost and XGBoost classifiers. A comparative analysis reveals that the bagging methods of joint learning demonstrate superior performance for our dataset in comparison to the reinforcement method. Furthermore, we can observe

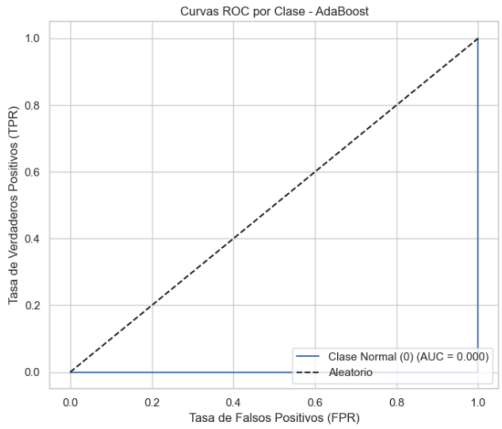
The area under the curve (AUC) for the attack class is greater than that for the normal class in all four graphs. This is due to the fact that there were more examples of training and testing for the attack class objective than for the normal class objective.



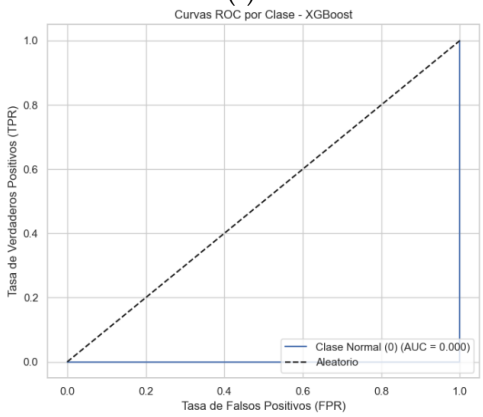
(a)



(b)



(c)



(d)

Figure 10. (a) and (b) are the ROC curves of the normal and attack classes for the Random Forest Classifier and the Extra Tree Classifier respectively. (c) and (d) are the ROC curves of the normal and attack classes for AdaBoost and XGBoost classifiers respectively.

Results

In this study, a total of 49 attributes were enumerated. Following the selection process, the training dataset contained 175,341 data points (UNSW_NB15_training-set.csv), while the test data amounted to 82,332 data points (UNSW_NB15_testing-set.csv) of the UNSW-NB15, 2018 dataset. The implementation of the proposed model, based on ensemble learning techniques, yielded robust results in the binary classification of network traffic as normal or malicious, using the

UNSW-NB15 dataset. Four algorithms were evaluated: The following algorithms were considered: Random Forest, Extra Trees, AdaBoost, and XGBoost. The precision, recall, and F1-score metrics for both classes (normal and attack) were extracted from the respective confusion matrices. Bagging-based models (Random Forest and Extra Trees) exhibited superior overall performance in comparison to boosting models (AdaBoost and XGBoost). The Random Forest classifier demonstrated the highest overall accuracy (86.99%) and exhibited a notable equilibrium between accuracy (97% for the normal class and 82% for the attack class) and recall (73% and 98%, respectively), suggesting a high attack detection capability without a substantial compromise to the false positive rate.

With regard to the discriminative performance evaluated using the ROC curves, a higher area under the curve (AUC) was observed for the "attack" class in all models. This is attributable to the greater number of representative samples in the training set. The ROC curves for Random Forest and Extra Trees exhibited a high degree of overlap, thereby confirming their similar behavior and high sensitivity. Furthermore, exploratory data analysis (EDA) revealed pertinent patterns. Distributions of features such as ct_srv_dst and ct_dst_sport_itm exhibited marked disparities between classes, thereby substantiating their status as pivotal attributes for detection. Conversely, a high correlation was identified between attributes such as tcprtt and ackdat. These attributes were controlled to avoid redundancy in the final model.

Conclusions

The findings of this study demonstrate that ensemble learning techniques, particularly bagging models such as Random Forest, offer an effective and reliable solution for the early detection of cyberattacks on enterprise networks. The implementation of the CRISP-DM methodology enabled the integration of exploratory analysis, feature selection, and multi-criteria evaluation to validate the effectiveness of the model in a realistic environment.

The utilization of box plots facilitated the comprehension of the distribution of data for each characteristic in relation to the labels. Consequently, the violin diagrams, in conjunction with the analysis of the data through box plots, facilitated the examination of the frequency distribution of each characteristic. Through a comparative analysis of the violin diagrams, insights were gleaned regarding the correlation between the features under consideration. Furthermore, violin diagrams with analogous shapes demonstrate a positive correlation. The utilization of dot plots facilitated the observation that the mean value of each characteristic for attack tags is approximately zero and exhibits significant variability for normal class tags. Univariate and bivariate analyses facilitated the visualization of the data and enabled the determination of which feature was more significant than others.

The researchers implemented several ensemble learning algorithms and determined that the random forest classifier exhibited the highest level of accuracy, with a value of 86.9%, followed by Extra Trees and the XGBoost classifier. AdaBoost obtained the lowest accuracy, although the performance of all classifiers varies very little and offers practically similar results. The empirical evidence suggests that it is feasible to attain high levels of precision and sensitivity through cost-effective, readily interpretable techniques in computer equipment, obviating the necessity of employing complex architectures such as deep neural networks. A notable strength of the proposed approach is its capacity to adapt to imbalances in data, a capability that is particularly advantageous in business contexts characterized by operational constraints. This work presents a replicable and extensible framework for institutions aiming to deploy intelligent early intrusion detection systems. The framework emphasizes the qualities of robustness, interpretability, and efficiency.

Discussion

The findings of the present study are consistent with the recent literature supporting the use of ensemble learning in cybersecurity environments. In contrast to models based on deep learning (DL), which, although they achieve accuracies close to 99% (Banoori & Hegde, 2023; Wang et al., 2023), may present limitations such as high computational demand and complex interpretability (Smith et al., 2024). The techniques employed in this study are designed to achieve an optimal balance between precision, efficiency, and model comprehension.

Conversely, the implemented algorithms demonstrate superior performance in comparison to conventional models such as Naive Bayes or logistic regression, particularly in scenarios where the classes are imbalanced. Despite not attaining the 90% global accuracy threshold, the Random Forest model exhibited consistent and balanced performance, thereby substantiating its viability as a pragmatic approach for initial threat detection.

It is also noteworthy that the UNSW-NB15 dataset, which was utilized in this study, signifies a substantial enhancement over the conventional

KDD'99 dataset. This enhancement is primarily attributed to the UNSW-NB15 dataset's provision of a more equitable distribution, exhibiting reduced redundancy and enhanced representativeness of contemporary attack patterns. This approach enabled the development of training models that exhibited superior generalizing capabilities and heightened resilience when confronted with real data.

Finally, although boosting models such as XGBoost have demonstrated competitive performance, their marginal gain does not justify their superior complexity compared to bagging models. This finding aligns with research conducted by Alghamdi et al. (2022) and underscores the importance of efficiency and simplicity in real industrial contexts, as evidenced by the work of Meenal Gaur et al. (2020). In subsequent research, it would be prudent to investigate hybrid approaches that integrate interpretability and power, such as multi-agent models or modular architectures. Furthermore, the incorporation of automated dimensionality reduction techniques (e.g., RFE or Chi2-PSO) to enhance the efficiency and scalability of the system is recommended.

References

A. Kareem, M. A. Mohammed, A. A. Jaber, A. A. A. Geman, et al. (2023). An efficient anomaly detection based on optimal deep belief network in big data. *International Journal of Intelligent Engineering and Systems*. https://doi.org/10.22266/ijies2023.0430.43

Aggarwala, P., & Kumar Sharma, S. (2015). KDD dataset attribute analysis: class-by-class for intrusion detection. 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015).

Ahmed, M. (2018). Collective anomaly detection techniques for network traffic analysis. *Annals of Data Science*, 5(4), 497–512.

Aydin, M., Zaim, A. H., & Ceylan, K. G. (2009). Design of a hybrid intrusion detection system for computer network security. *Computers & Electrical Engineering*, 35(3), 517–526.

Bhati, B. S., Rai, C., Balamurugan, B., & Al-Turjman, F. (2020). An intrusion detection scheme based on the set of discriminant classifiers. *Computers and Electrical Engineering*, 86.

Bhati, S., Rai, H., Balamurugan, B., & Al-Turjman, F. (2020). Ensemble approach for effective intrusion detection using discriminant classifiers. *IEEE Access*, 8, 140431–140444.

Biswas, S. K. (2018). Intrusion detection using machine learning: a comparative study. *International Journal of Pure and Applied Mathematics*, 118, 101–114.

Boutaba, R., Salahuddin, M. A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., & Caicedo, O. M. (2018). Comprehensive study on machine learning for networks: evolution, applications and research opportunities. *Journal of Internet Services and Applications*.

Chapaneri, R. (2019, March 14). Exploratory analysis of the UNSW-NB15 dataset for the detection of malicious networks. *Medium*. https://medium.com/@radhikachapaneri/exploratoryanalysis-of-unsw-nb15-dataset-for-detecting-maliciousnetwork-traffic-6b3e6743e907

Das, A., Abraham, A., & Das, S. (2010). Intrusion detection system using rough set and support vector machine. *International Journal of Network Security*, 10(1), 1–10.

Das, V., Pathak, V., Sharma, S., Sreevathsan, Srikanth, M. V., & Kumar, G. (2010). Network intrusion detection system based on machine learning algorithms. *International Journal of Computer Science and Information Technologies*, 2.

Data from the 1999 KDD Cup. (1999, October 28). http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Description of the UNSW-NB15 dataset. (2018, November 14). ADFA-NB15 Data https://www.unsw.adfa.edu.au/unsw-canberra-Ciberseguridad/Conjuntos

Divekar, A., Parekh, M., Savla, V., Mishra, R., & Shirole, M. (2018). Reference Datasets for Anomaly-Based Network Intrusion Detection: Alternatives to KDD CUP 99. In *IEEE International Conference on Computing, Communication and Security (ICCCS)*.

Doshi, S. (2019, February 3). Data analysis using data visualization with Seaborn. *Towards Data Science*. https://towardsdatascience.com/analice-los-datos-a-traves-de-visualizacion-de-datos-usando-seaborn-255e1cd3948e

Fayaz M., Alazab M., Rauf H. T. (2024). An Explainable Ensemble Learning Model for Early Detection of Attacks in IoT-Enabled Industrial Systems. *Computer Science proceeded*. https://doi.org/10.1016/j.procs.2023.12.056

G. A. Cordero, C. Catania, C. García Garino (2019). Deep autoencoding models for unsupervised anomaly detection in cybersecurity: A survey. *Computers & Security*. https://doi.org/10.1016/j.cose.2019.02.001

Ghoneim, S. (2019, April 2). Accuracy, recall, precision, F-score, and specificity – which one to optimize? *Towards Data Science*. https://towardsdatascience.com/precisión-recuperación-precisión-puntuación-especificidad-que-optimizar-en-867d3f11124

Gupta, D., & Rani, R. (2020). Improved malware detection through big data and co-learning. *Computers & Electrical Engineering*, 86.

Liu, H., & Lang, B. (2019). Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey.

Meenal Gaur, Mohammad A. Tawhid, Kandarpa Kumar Sarma, Rituparna Chaki, Nilanjan Dey (2020). A survey on deep learning techniques for cyber security in the Internet of Things. *Computers & Security*. https://doi.org/10.1016/j.cose.2020.101860

Mohamed H. Behiry, M. Aly (2022). Cyberattack detection in wireless sensor networks using a hybrid feature reduction technique with AI and machine learning methods. *Journal of Big Data*. https://doi.org/10.1186/s40537-022-00809-3

Norah Abdullah Alghamdi, Areej Mohsin, A. Elngar, Amal Farouk, Mohamed Mostafa Morsy, Mohammed Nabil (2022). A Multiagent and Machine Learning-based Hybrid NIDS for Known and Unknown Cyberattacks. *International Journal of Intelligent Engineering and Systems*. https://doi.org/10.22266/ijies2022.0630.21

Othman, S. M., Ba-Alwi, F. M., Alsohybe, N. T., & Al-Hashida, A. Y. (2018). Intrusion detection model using machine learning algorithm on Big Data environment. *Journal of Big Data*, 5(1), 34.

Othman, S. M., Ba-Alwi, F. M., Alsohybe, N. T., & Al-Hashida, A. Y. (2018). Intrusion detection model using a machine learning algorithm in a big data environment. *Journal of Big Data*, 5(1).

Rakesh Banoori, Nagaratna P. Hegde (2023). A hybrid CNN-RNN approach for anomaly-based intrusion detection system. *Journal of Big Data*. https://doi.org/10.1186/s40537-023-00870-w

Ramaiah, S. M., Pushpalatha, M. P. (2019). Anomaly-Based Intrusion Detection System Using Machine Learning in Software-Defined Networking. *Applied Sciences*. https://doi.org/10.3390/app9020364

Rana Imran Hussain, Mohammad Shorfuzzaman, Ibrahim Khalil, Omar Alfandi, Sagheer Abbas, et al. (2022). Network Anomaly Detection Based On Late Fusion Of Several Machine Learning Algorithms.

Ren, J., Guo, J., Qian, W., Yuan, H., Hao, X., & Jingjing, H. (2019). Development of an effective intrusion detection system through hybrid data optimization based on machine learning algorithms. *Security and Communication Networks*.

Ren, W., Zhou, Y., Liu, X., & Dai, Y. (2019). DO_IDS: A data optimization based intrusion detection system using machine learning algorithms. *Computers & Security*, 86, 147–161.

Sarker, I. H., Abushark, Y. B., Alsolami, F., & Khan, A. I. (2020). Intrudtree: A machine learning based cyber threat detection model. *Computers & Security*, 92, 101750.

Sarker, I. H., Abushark, Y. B., Alsolami, F., & Khan, A. I. (2020). IntruDTree: A machine learning-based cybersecurity intrusion detection model. *Symmetry*, 12(5), 1–17.

Sharma, N. V., & Yadav, N. S. (2021). An optimal intrusion detection system through recursive feature removal and a set of classifiers. *Microprocessors and Microsystems*, 104293.

Sovilj, M., et al. (2020). Cyber intrusion detection using deep generative models: Variational autoencoder with a Gaussian mixture model. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2020.113471

Syeda Mehak Raza, Sumit Kumar, Fawad Ali, Muhammad Awais, Hafeez Anwar, Ali Kashif Bashir (2023). DL-CAD: A Deep Learning-Based Cyber Attack Detection System Using Multistage Feature Selection Method. *Sensors*. https://doi.org/10.3390/s230707091

Tuğba Aytaç, Muhammed Ali Aydın, Abdül Halim Zaim (2020). Detection DDOS Attacks Using Machine Learning Methods. *Electric*. https://doi.org/10.5152/electrica.2020.20049

U. Aickelin, D. Green, D. M. Capel (2023). The Missing Link in Network Intrusion Detection: Taking AI/ML Research Efforts to Users. *IEEE Internet Computing*. https://doi.org/10.1109/MIC.2023.3236449

UNSW_NB15 features.csv. (n.d.). https://www.unsw.adfa.edu.au/unsw-canberra-ciberseguridad/ADFA-NB15-Conjuntos data/NUSW-NB15_features.csv

Verma, A., & Ranga, V. (2017). Statistical analysis of the CIDDS-001 dataset for network intrusion detection systems using remote machine learning. In *6th International Conference on Intelligent Computing and Communications*.

Wagh, S. K., Pachghare, V. K., & Kolhe, S. R. (2013). Survey on intrusion detection systems using machine learning techniques. *International Journal of Computer Applications*, 78(16).

Wagh, S., Pachghare, V., & Kolhe, S. (2013). Survey on intrusion detection system using machine learning techniques. *International Journal of Computer Applications*, 78(16), 7–11.

Wang, K. (2019). Network data management model based on the Naïve Bayes classifier and deep neural networks in heterogeneous wireless networks. *Computers & Electrical Engineering*, 75, 135–145.

Wenjie Wang, Xueqian Wang, Ming Li, Zhen Zhang, Hao Wu (2023). Hybrid feature selection and multi-branch CNN for network intrusion detection. *Data Mining and Knowledge Discovery (DMCAD, Springer)*. https://doi.org/10.1007/s10618-023-00967-3

Yunfei Guo, Chao Yan, Yong Zhang, Chuanjiang Jiang, Yulong Wang (2023). Traffic Management in IoT Backbone Networks: A Multi-Arm Bandit Based GNN Approach with SDN Orchestration. *Sensors*. https://doi.org/10.3390/s23167091