IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2023.0322000

A Comprehensive Survey on Ensemble Learning-based Intrusion Detection Approaches in Computer Networks

Thiago José Lucas[1], Inaê Soares de Figueiredo[1], Carlos Alexandre Carvalho Tojeiro[1], Alex Marino G. de Almeida[1], Rafał Scherer[2] (Member, IEEE), José Remo F. Brega[1], João Paulo Papa[1] (Senior Member, IEEE), Kelton Augusto Pontara da Costa[1] (Senior Member, IEEE)

[1]Department of Computing, São Paulo State University, Bauru, Brazil. (e-mail: {t.lucas,inae,ctojeiro,amarino,remo.brega,joao.papa,kelton.costa}@unesp.br)
[2]Department of Computing, Czestochowa University of Technology, Częstochowa, Poland. (e-mail: rafal.scherer@unesp.br)

Corresponding author: Thiago José Lucas (e-mail: t.lucas@unesp.br).

The authors are grateful to Brazilian FAPESP grants 2021/05516-1, and 2021/10320-9, and Unesp Edital n-06:2023 PROPe. The project financed under the program of the Polish Minister of Science and Higher Education under the name "Regional Initiative of Excellence" in the years 2019 - 2023, project number 020/RID/2018/19, the amount of financing PLN 12,000,000.

ABSTRACT Machine learning algorithms present a robust alternative for building Intrusion Detection Systems due to their ability to recognize attacks in computer network traffic by recognizing patterns in large amounts of data. Typically, classifiers are trained for this task. Together, ensemble learning algorithms have increased the performance of these detectors, reducing classification errors and allowing computer networks to be more protected. This research presents a comprehensive Systematic Review of the Literature where works related to intrusion detection with ensemble learning were obtained from the most relevant scientific bases. We offer 188 works, several compilations of datasets, classifiers, and ensemble algorithms, and document the experiments that stood out in their performance. A characteristic of this research is its originality. We found two surveys in the literature specifically focusing on the relationship between ensemble techniques and intrusion detection [1] [2]. We present for the last eight years covered by this survey a timeline-based view of the works studied to highlight evolutions and trends. The results obtained by our survey show a growing area, with excellent results in detecting attacks but with needs for improvement in pruning for choosing classifiers, which makes this work unprecedented for this context.

INDEX TERMS Cybersecurity, Machine Learning, Ensemble Learning, Intrusion Detection Systems

I. INTRODUCTION

INFORMATION can be considered the greatest asset of a company or corporation in the modern economic scenario, according to [3], and this fact highlights that owners should be concerned about safeguarding their data, especially about confidentiality, integrity, and availability. The fourth industrial revolution, or ''Industry 4.0'', a term created by the President of the World Economic Forum, Klaus Schwab [4], describes the paradigm shift in the way people live and relate to each other and, in particular, how companies and corporations need to know how to manage information in order to survive in a competitive market, reinforcing the need as mentioned earlier to protect data through the implementation of information security methods.

A cybersecurity report by [5] compiles projections about the global information security scenario, highlighting the main threats, estimates, and suggesting priorities for risk mitigation. According to [6] and [5], ahead of risks of legal uncertainty (19%), pandemics (22%), natural disasters (25%) and interruption of services (42%), the risks with information security incidents represent in 2022 a total of 44% for businesses globally.

Projections of [7] and [5] estimate a loss of 10.5 trillion dollars in 2025. For comparative purposes, in 2015, it was 3 trillion dollars, and in 2021, 6 trillion. Investment in cybersecurity tends to maintain the annual level of 1.75 trillion annually until 2025.

Also, according to [8] and [5], it is worth noting that prioritizing the protection against attacks based on people (social engineering), investing in limiting data loss and business interruption, and in particular, applying advanced analytics and intelligence technologies for security are essential for

the survival of companies in an environment as critical as information security.

The statistics mentioned above justify the efforts employed in elaborating this research.

Given the context presented, it is essential to apply intelligent methods to detect attacks to prevent damage. Intrusion Detection and Prevention Systems (IDS/IPS) based on machine learning can analyze large amounts of data coming from network traffic and, through the training step, create robust models capable of recognizing attack patterns.

Although not trivial, creating an IDS model is a challenge already known and debated in cybersecurity community. The challenge is increasingly related to reducing the computational cost of training and increasing detection results by reducing false-positive and false-negative alarms.

The general objectives of this work is to obtain state-of-the-art ensemble learning techniques applied to intrusion detection. The specific objectives were as follows:

- Conducting a comprehensive systematic review of the literature to obtain related works from the period between the years 2015 and 2022 (to work with years already ended, allowing to organize complete annual statistics and present a timeline-based view of the works);
- Extraction of the main characteristics for the works obtained so that it was possible to observe trends in the use of classifiers, datasets, and ensemble algorithms.

Table 1 compiles a brief comparison of this work with two other surveys of similar scope:

TABLE 1. Brief comparison of this proposal with two surveys of similar scope.

Survey	Published	Coverage	Organization	Focuses
[1]	2016	19	Supervised, unsupervised and Hybrid approaches	Distributed systems
[2]	2017	34	Homogeneous and heterogeneous classifiers	General
This	–	188	Timeline, per year	General

The motivation for preparing this survey is to provide the scientific community with a comprehensive view of the relationship between Ensemble Learning algorithms and Intrusion Detection Systems. Due to the scope of the research, we provide a timeline-based view that allows us to observe the evolution of the relationship mentioned above over the last few years.

The scarcity of comprehensive surveys in the specialized literature that address the evolution of Ensemble Learning algorithms in Intrusion Detection Systems (IDS) has been a significant gap in cybersecurity. This lack of longitudinal analysis has made it difficult to understand emerging trends, technological progress, and the relative effectiveness of different approaches to IDS over time. The present work is an essential contribution to resolving this substantial lack, as it provides a detailed and comprehensive chronological overview of ensemble algorithms in IDS, highlighting the variations in the ensemble methods employed and the data sets relevant information used over the years. By charting a historical narrative of approaches and identifying emerging trends, this survey offers researchers and practitioners valuable insight into the evolution of intrusion detection strategies, enabling critical insights to guide future research and improve the effectiveness of cybersecurity solutions.

The remainder of this manuscript is organized in such a way that Section II contains an explanation concerning the method used to produce this work. Section IV covers brief descriptions of the main components of an IDS. Section III contains detailed reports on each of the works listed in this review. In Section V we describe our interpretation of the area in light of the works listed, and finally, in Section VI we report the conclusion of this work.

II. METHODOLOGY

A systematic literature review aims to identify, analyze and interpret available evidence related to a particular research topic or phenomenon of interest. For the production of this review, the set of instructions related in the Scannavino et al. [9] study were followed. The main tasks listed can be seen in Figure 1.

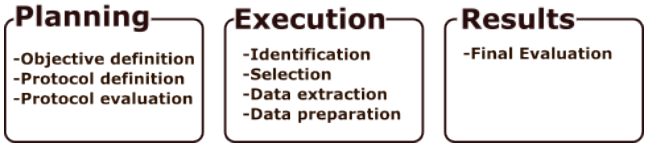


FIGURE 1. Systematic review procedure, adapted from Scannavino et al. [9]

The planning task consists of elaborating a protocol that is the central element of the work that will guide the execution. The development of a suitable protocol aims to eliminate biases that may be committed by Kitchenham et al. [10].

The main objectives of this study can be summarized as follows: obtaining the State of the Art (SOTA) in the research area relevant to IDS after year 2014, obtaining an overview of the main works related to the components and techniques used in elaborating an IDS, and reporting the main contributions.

A systematic literature review was carried out, where articles registered in the scientific bases IEEExplore, ACM Digital Library, Springer, Wiley, Hindawi and MDPI were collected by searching in the intervals between the years 2015 and 2022 for records that contained the terms ”ensemble“ and ”intrusion“ in both the title and the keywords.

The search protocol can be seen in Table 2. The acquisition process of related articles is shown in Figure 2; it is possible to notice that only for the Springer repository the search criterion did not take into account the occurrences of keywords in the abstract, this was since if we added the abstract as a search criterion the number of articles returned exceeded

TABLE 2. Research protocol applied to perform this work.

General Info	
Title	A survey on Ensemble Learning applied to intrusion detection in computers networks.
Objectives	Extract SOTA of Ensemble Learning applied to intrusion detection.
Research Question	
Keywords	ensemble; intrusion
Search string	Q1 ->((((TITLE OR KEYWORDS) == ("ensemble" AND "intrusion")) AND PUBLICATION_YEAR == ({2015, ..., 2021)) Q2 ->(((TITLE == ("ensemble" AND "intrusion")) AND PUBLICATION_YEAR == ({2015, ..., 2022))
Sources Selection Criteria	Digital repositories of relevant scientific articles withinr computer science academia.
List of sources	IEEE, ACM, MDPI, Springer, Hindawi, WILEY
Selection and Evaluation of Studies	
Study selection strategy and assesment of quality	We adopted the procedure described in Keshav et al. [11] to identify studies concerning the research topic of interest to this study.
Inclusion criteria	Considering the first step of Keshav et al. [11], all articles that denoted covering the area of security and computer networks were included in the scope of this study.
Exclusion criteria	Considering the first step of Keshav et al. [11], all articles that did not cover the area of security and computer networks were excluded from the scope of this study.
Data synthesis and presentation of results	
Data Extraction Strategy	Analysis of essential topics was based on analyzing relevant terms extracted from the abstracts of all articles, with the help of wordcloud formulation. Among the relevant terms, it was possible to note terms such as MODEL, DATASET, ACCURACY, DETECT, and LEARNING. These keywords guided the compilation of studies.
Summarization strategy	The summarization of the results was based on the relevant terms contained in the abstracts, which can be viewed in the wordcloud as noted in Figure 4. With the selection of these relevant terms, we segregated the studies by using Datasets, Ensemble Models, and consequently the Base Classifiers, already making up the rest of the article reading process according to Keshav et al. [11].

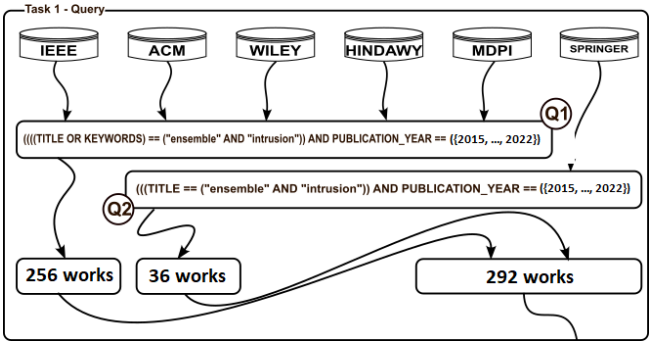


FIGURE 2. Acquisition Queries.

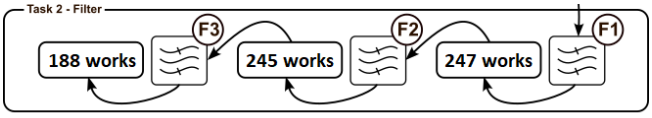


FIGURE 3. Filtering Process.

five thousand articles. Thus, the acquisition of articles was restricted to the title for the Springer repository.

Once the phase of acquiring articles related to the area of interest was completed, which totaled 292 articles segregated by repositories. Then the filtering phase of the list of articles took place as illustrated in Figure 3. The refinement was performed using the reading technique proposed by Keshav et al. [11], and as a criterion for the inclusion of articles, we selected all articles that denoted covering the area of security

FIGURE 4. Wordcloud obtained by all abstracts.

and computer networks included in the scope of this study. As exclusion criteria, we drop all studies out of inclusion criteria. At the end of this stage, we have 188 selected articles related to the topic of interest.

Once the selection of articles is finalized, and in order to obtain insights into the research area of interest, we use a word cloud that provides data and information to management and support decisions [12]. Exploration of essential topics was based on analyzing relevant terms extracted from the abstracts of all articles, with the help of word cloud formulation, as noted in 4.

The summary of the results was based on the relevant terms contained in the abstracts, which can be viewed in the word cloud presented in Figure 4. With the selection of these relevant terms, we divided the studies by used Datasets, Ensemble Models, and consequently the Base Classifiers, already making up the rest of the article reading process according to Keshav et al. [11]. Still regarding the total volume

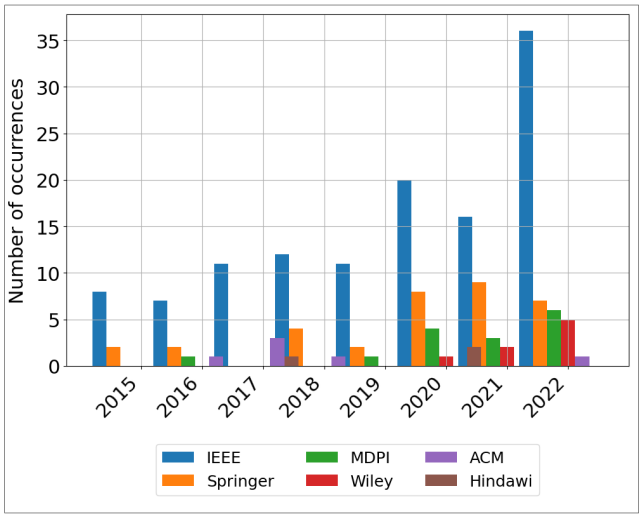


FIGURE 5. Distribution of works obtained by publisher per year.

of articles, it can be seen in Figure 5 that the IEEE and Springer repositories represent 84% of the total number of articles listed in this study.

In this section, we described the elaboration of the research protocol and the tasks performed to complete the research protocol. At the end of the tasks, we obtained 188 articles related to the subject of Intrusion Detection Systems using Ensemble techniques. With the articles in hand, we created a word cloud that gave us insights into this work's subjects and relevant components. In the next section, we produce a careful and synthesized article-by-article review of those selected by the research protocol section. This research was initially part of a compilation of publications written for a Ph.D. thesis, available at [13]. However, the original thesis contained works obtained between 2015 and 2021. For this document, we added works referring to the year 2022, as there was a considerable increase in the number of publications related to the topic.

III. LITERATURE REVIEW

In this section, we discusses article by article those chosen after the selection process, as presented in Table 2. To conduct a chronological development line, we segregated the reports by year of publication in which we described the major findings for each manuscript.

A. WORKS PUBLISHED IN 2015

Mehra et al. [14] carried out an experiment using the KDD-Cup'99 dataset, which was initially processed by the C4.5 algorithm to generate Decisions Trees that, in a second step, were grouped by the AdaBoost ensemble learning algorithm. Finally, Snort integration was carried out to perform adequate tests in real-time. The authors highlight the achievement of a Detection Rate of 89.56%, while the value of the False Alarm Rate was 0.1% for the tests performed.

Haq et al. [15] was used Ensemble learning techniques for the feature selection process and the alarm classification. A union and intersection logic between the features obtained by the Best First, Genetic Search, and Rank Search algorithms allowed an adequate choice of the best features in the NSL-KDD dataset. The Ensemble was performed through Majority Voting among the Bayesian Network, Naive Bayes, random forest, and J48 classifiers in the classification process. The proposed model obtained a True Positive Rate of 98.0% against a False Positive Rate of 0.021%.

Milliken et al. [16] proposed a Stacking Ensemble model. The authors used the ISCX 2012 dataset as a basis for the tests and the OneR, Conjunctive, and Naive Bayes classifiers in a Stacking model to perform the Ensemble. The experiments showed that the analysis of entropy between pairs of features from different subsets could contribute to the generation of a subset more suitable for attack classification. The experiments obtained an F-Score of 92% as the best result.

Amudha et al. [17] proposed a combination of Core Vector Machine as a classifier and Bagging as an Ensemble technique. Other tests were performed by the authors and are documented in their publication, such as the application of AdaBoost to Ensemble and Naive Bayes, decision tree, and random forest as classifiers. However, the best results in the KDD-Cup'99 dataset pointed to an accuracy of 98.7% for DoS detection, 98.78% for probe attacks, 98.16% for R2L, and 98.41% for U2R using the combination of Bagging and Core Vector Machine.

Sornsuwit et al. [18] presented an expert model for detecting R2L and U2R attacks. It combines Naive Bayes, multilayer perceptron, decision tree, support vectors machine, and k-nearest neighbors classifiers using the Ensemble AdaBoost technique. The best features in the KDD-Cup'99 dataset were selected through correlation analysis. The best results obtained were a sensitivity of 76% for Naive Bayes and multilayer perceptron and a specificity of 99.05% for a combination of different Naive Bayes classifiers.

Gaikwad et al. [19] presented a model that uses Bagging as an Ensemble technique for reptree classifiers. The authors selected the best features in the NSL-KDD dataset using BIRCH Hierarchical Clustering. The results obtained were 99.67% accurate, while a False Positive Rate of only 0.3% was observed. Although they did not specify which classifier was used as a base, Sreenath et al. [20] also modeled an intrusion detector using Bagging on the same NSL-KDD, and the measurement accuracy in detection was 97.85%.

Through the one-vs-all Ensemble technique, Ye et al. [21] combined three types of classifiers to improve the performance in intrusion detection for the proposed model. The authors used decision tree, neural network, and support vectors machines algorithms as classifiers. The dataset used in the tests was the NSL-KDD and the feature selection technique used was not specified. The accuracy obtained in the classification was 97.35%.

Robinson et al. [22] presented two models that are experts in detecting DDoS attacks. Using random forests as

classifiers, the authors modeled two Ensembles: one with AdaBoost and the other with bagging to compare them. The bagging-based model obtained the best results when observing the values obtained for False Negative Rate, False Positive Rate, and Precision. The only evaluation metric where AdaBoost was superior was Recall. All tests were performed on the LLS-DDoS1.0, CAIDA 2007, and CAIDA Conficker datasets.

As it is a standard protocol for remote access on Unix and Linux servers, the SSH protocol is widely used for dictionary attacks to gain remote privileges to control vulnerable servers. This is what Gonzalez et al. [23] claims in their experiment, where they tested different Ensemble methods (bagging, boosting, AdaBoost, MultiboostingAB, and Rotation Forest) to classify alerts from the Euskalert honeypot. The best results were with Bagging, where it was possible to measure a True Positive Rate of 99.93%.

Tama et al. [24] proposed an experiment with particle swarm optimization and tree-based classifiers for network intrusion detection. PSO is used for feature selection, and the classifiers (C4.5, RF, and CART) were tested in multiple ensemble formats. The NSL-KDD dataset was used. The best accuracy (99.80%) and false-positive rate (0.02) were obtained with PSO-50 and the Average Probability ensemble scheme.

B. WORKS PUBLISHED IN 2016

An Ensemble with C4.5, reptree, and RTree classifiers combined with Bagging was presented by Al-Jarrah et al. [25] In the pre-processing, the authors chose the best features based on correlation for the ISOT dataset, and the tests performed pointed to an accuracy of 99.97%.

Alotaibi et al. [26] presented an intrusion detector for wireless networks. Using decision trees as classifiers, the authors modeled an Ensemble technique by majority voting where the most voted label among Bagging, random forest, and an extra tree is elected. The tests were performed on the AWID dataset, and the results showed an accuracy of 96.32% and a precision/recall of 96%.

Folino et al. [27] highlights two problems that Machine Learning methods usually present when dealing with alert classification situations: unbalanced classes and speed of streams for real-time detection. To create a model capable of dealing with these problems more effectively, the authors modeled an Ensemble using boosting to combine the following classifiers: J48, JRIP rule learner, Naïve Bayes tree, Naïve Bayes, OneR, logistic model trees, logistic regression, decision stumps, and k-nearest neighbor. The best results in tests performed with the ISCX IDS 2012 dataset obtained a precision of 88.28%, a recall of 80.79%, and an area under the precision/recall curve of 89%.

Using an Ensemble proposal to implement rotation forests, Tama et al. [28] experimented with 20 classifiers to create a model for intrusion detection in wireless networks. Several combinations were tried in the GPRS dataset, and the best results for detection in WEP/WPA and WPA2 bases (which are protocols for wireless networks) were obtained with the best first decision trees grouped by the Ensemble rotation forest algorithm. The area under the precision/recall curve calculated in the tests was 96.01%.

Mehetrey et al. [29] presented an intrusion detector for cloud computing that distributes the task of capturing packets among the network hosts and, by sampling, unites in a central controller the Ensemble routine that was implemented using Bagging. The classifier implemented by the authors was the C-Fuzzy decision trees. In tests with the KDD-Cup'99 dataset, the proposed model obtained an accuracy of 99.47%.

A combination of two feature selection methods, PCA and LDA, combined with the SVM classification algorithm was proposed by Aburomman et al. [30] The pairs of standard features obtained by the PCA and LDA methods applied to the KDD-Cup'99 dataset are processed by a weighted majority voting Ensemble for ten SVM classifiers. The authors obtained an accuracy of 92.1% in the classification of alerts combined with a false positive rate of 1.9% and a false negative rate of 10.8%.

Kiranmai et al. [31] presented an expert intrusion detection system for cloud computing focusing on DDoS attacks. With its dataset. The proposed Ensemble method was called Consensus Cluster Plus, which united hierarchical cluster classifiers. The authors did not present results that could measure the quality of the presented system.

Wang et al. [32] presented an Ensemble using Random Committee and voting. Combining Bayesian network and random tree classifiers, the method obtained excellent results when observed by the area under the precision/recall curve in the KDD-Cup'99 dataset: 99.9% for Probe and R2L; 100% for DoS, and 99.5% for U2R.

DAREnsemble is an IDS architecture developed by Gaikwad et al. [33] that uses rule learners and decision trees, implemented with the voting role combination method. Compared to other ensemble models, such as Bagging and AdaBoost, DAREnsemble presents better results when using Average Probability as a combination rule, reaching 99.88% accuracy. For training and testing, the NSL-KDD dataset was utilized.

Working with KDD99 and ISCX IDS datasets, Folino et al. [34] proposed a distributed intrusion detection framework using ensemble learning and non-trainable functions, CAGE-MetaCombiner. On the KDD99 dataset, the model performs better than the comparison model for minority classes and almost as well on the majority classes. The model also performs better on the ISCX dataset than the non-specialized algorithm with a specialized ensemble.

A system called SCDNN is proposed by Ma et al. [35], and it combines spectral clustering and deep neural network algorithms. SC separates features by similarity, facilitating the identification of unknown attack types. The KDD-Cup'99 and NSL-KDD are divided into six subsets for testing the model. Analysis of performance is done by dataset, and results vary for each class and dataset. The system outperforms the comparison models in many instances.

C. WORKS PUBLISHED IN 2017

Miller et al. [36] performed several experiments on the NSL-KDD dataset to compare the following Ensemble techniques: majority vote, weighted vote, naive Bayes combination, Bagging, boosting, rotation forest, and random forest, all using Naive Bayes classifiers. The best results were obtained with the naive Bayes combination, where it was possible to measure 84.13% of correctly classified alerts while 15.86% of erroneously classified alerts were observed.

Kamarudin et al. [37] proposed an expert intrusion detection system for web attacks. The Ensemble LogitBoost technique increased the ability of random forest classifiers to detect attacks. Performing tests on the NSL-KDD and UNSW-NB15 datasets, the authors performed the selection of features combining wrapper and filter techniques. The observed detection rates were 89.75% and 99.10% for the datasets mentioned above, respectively.

Rajasekaran et al. [38] presented a sequential ensemble for alert classification. In the first layer, an EMSVM classifier is applied, and later (considering that only data labeled as benign in the current step go to the next step), the other packets are classified by k-NN and by SMO in the order presented. The selection of the best features is made with an Intelligent Agent-based Attribute Selection Algorithm (IAASA). The results obtained in the KDD-Cup'99 dataset were approximately 97% accurate for Probe, 97.5% for DoS, 99% for U2R, and 86% for R2L.

Chen et al. [39] presented an unsupervised intrusion detector that creates an Ensemble by voting between four clustering algorithms: DBSCAN, One-SVM, Agglomerative Clustering, and Expectation Maximization. The results obtained in the experiments with the NSL-KDD dataset were a detection rate of 91.03% and a false positive rate of 2.26%.

After selecting the best features in the NSL-KDD dataset using PSO and correlation-based selection, Tama et al. [40] implemented three Ensemble techniques for SVM classifiers: majority voting between AdaBoost and RSM. The proposed model obtained an accuracy of 85.01% and a false positive rate of 12.6%.

Primartha et al. [41] created majority voting ensembles for random forest classifiers to detect intrusion in three different datasets: NSL-KDD, UNSW-NB15, and GPRS. For the datasets above, the observed accuracy was 99.57%, 95.5%, and 91.8%.

Jabbar et al. [42] proposed an Ensemble-based intrusion detection system of clusters. Using the Gure KDD dataset and the ADTree, k -Means, and k-NN classifiers through weighted majority voting for binary classification, the tests obtained an accuracy of 99.93% and a detection rate of 99.8%. Aravind et al. [43] performed a similar experiment were using several distance metrics. The authors obtained 90% accuracy using k-means in the UNSW-NB15 dataset. In the same vein, Ruoti et al. [44] proposed a detector that addresses the application of Ensemble methods through voting using several unsupervised classifiers in the NSL-KDD dataset; however, the authors did not disclose the results obtained.

Belouch et al. [45] performed a comparison between three different Ensemble techniques: booting, Bagging, and Stacking. The authors combined four different classifiers: decision tree, Naive Bayes, multilayer perceptron, and reptree. All experiments were performed on the UNSW-NB15 dataset. The publication documents all the results, and it is possible to observe that the best Ensemble in terms of accuracy was a Stacking of reptree followed by a Stacking of decision trees.

Niranjan et al. [46] presented an intrusion detector that combines a selection of the best features using an information gain ratio and a voting scheme between Bagging, Random Committee, and decision tree. Considering that the first two use random trees as classifiers, the authors performed tests on the NSL-KDD and KDD-Cup'99 datasets, were both pointed to a valid positive rate of 100% and a false positive rate of 0%.

Garg et al. [47] presented their Ensemble model using Very Fast decision trees classifiers and selecting the best features with Hoeffding bound. The presented model obtained a detection rate of 98.58%. All tests were performed on the KDD-Cup'99 dataset.

Three different datasets were used to test an Ensemble of support vectors machines proposed by Reddy et al. [48] Using Rough set theory to obtain the best features, the authors obtained 99.95%, 100%, and 99.98% accuracy for the KDD-Cup'99, HTTP CSIC 2010, and UNB ISCX datasets, respectively.

Tim et al. [49] created an intrusion detection model that combines several Ensemble methods using the decision tree as the base classifier. Each cluster uses a technique: Bagged tree (decision trees based on Bagging), AdaBoost, logitboost, gentleboost (derivation of AdaBoost), and RUSboost (boosting algorithm for class balancing). The best detection rates were obtained with bagged tree and Gentleboost, obtaining 99.1% when tested on the UNSB-NB15 dataset.

Seeking to detect Neptune-type attacks (a kind of DoS) on the NSL-KDD dataset, Mkuzangwe et al. [50] created a binary classifier capable of detecting intrusion by differentiating normal from abnormal connections. The authors implemented an Ensemble via AdaBoost joining decision stump classifiers which allowed them to obtain an accuracy of 87.83%. The feature selection method used in the experiment was the Information Gain Ratio.

Ludwig et al. [51] presented an intrusion detector that uses a neural network to create an Ensemble containing an autoencoder, a deep belief neural network, a deep neural network, and an Extreme Learning Machine. The presented detector is capable of performing binary classification. The author performed the experiments on the NSL-KDD dataset and obtained a detection rate of 97.9% in addition to an accuracy of 92.4%

Salunkhe et al. [52] propose an IDS focused on enhancing the detection rate by treating minority classes. KDD-Cup'99 dataset is utilized to evaluate the system's performance on multi-class classification, using different base classifiers: Lo-

gistic regression, J48, and Naive Bayes. All classifiers perform better when applied with the proposed method.

D. WORKS PUBLISHED IN 2018

An Ensemble of SVM classifiers using Bagging as clustering techniques was proposed by Tengl et al. [53] The intrusion detector obtained a subset through PCA applied to the NSL-KDD dataset. The authors divided the classification task between three detectors according to three computer network protocols present in the dataset mentioned above. A genetic algorithm collaborates in the process of choosing the weights in order to potentiate a particular classifier to the detriment of another in the cluster. The accuracy obtained in the tests was 88.28%.

The purpose of the intrusion detector proposed by Yuan et al. [54] is real-time data processing. The authors point out that if statistical properties of data alterations are observed throughout monitoring, particularly the variance between attacks, it may be possible to build a more robust classification system. In this sense, the authors proposed an Ensemble-based on the concept of incremental drift learning. The classifiers used were Naive Bayes, Stochastic Gradient Descent, and multilayer perceptron. In tests performed on the KDD-Cup'99 dataset, the observed accuracy was 94.91%.

Sun et al. [55] presented an intrusion detection model that combines three classic Ensemble techniques: Bagging, boosting, and Stacking. The base classifiers used were SVM and k-NN. With tests performed on the NSL-KDD dataset after generating a subset through PCA, it was possible to measure the best accuracy results for the classification of regular packages with 99.41% and that of probe packages with 93.13%.

Gao et al. [56] presented a semi-supervised classification model. Consisting of two steps, the classifier uses an Ensemble through weighted voting composed of decision trees and neural networks trained in a supervised way, that is, with access to data labels. In a second moment, there is the removal of redundant and noisy data in the dataset sample without labels so that the resulting subset is processed by the same Ensemble of the first moment. The authors used a subset obtained by applying PCA to the NSL-KDD dataset, and the accuracy obtained was 84.54%.

Gautam et al. [57] proposed an intrusion detection model that groups, through Bagging, three classifiers, namely Naive Bayes, Adaptive Boost, and PART. The dataset used by the authors was the KDD-Cup'99, from which the best features were obtained by observing analyzes based on entropy and filtering. The results obtained point to an accuracy of 99.97%.

Vaca et al. [58] modeled an intrusion detection system for wireless networks. Using decision trees for classification, the authors tested four different Ensemble techniques: Bagging, random forests, extra-trees, and XGBoost. After pre-processing the AWID dataset, the resulting subset was better processed by combining decision trees and random forests, which showed an accuracy of 95.87% in multi-class detection.

Shen et al. [59] presented an intrusion detector based on an Extreme Learning Machine and optimized with a bat algorithm (inspired by the behavior of bats), whose function was to eliminate the worst classifier among the four different subsets that were generated through bootstrap selection (more details on bootstrapping are covered under "Bagging" in Section [sec:machinelearning]) in the original dataset. The Ensemble was applied by majority voting among the three classifiers not eliminated by the meta-heuristic optimization algorithm. The authors performed tests on the KDD-Cup'99, NSL-KDD, and Kyoto datasets. The measured accuracies were 98.94%, 97.46%, and 99.19% for the datasets mentioned above.

Mirza et al. [60] proposed an Ensemble by majority voting. The label most voted by three different classifiers, namely logistic regression, neural network, and decision trees, is attributed to the analyzed data. The subset processed by the proposed method was obtained after applying the PCA method to the KDD-Cup'99 dataset. The accuracy obtained in the tests was 96.14%. Another proposal of Ensemble by majority voting was presented by Marir et al. [61] where the authors reduce the dimensionality of the datasets using a deep belief network and classify the data through a grouping of support vector machines. This model was tested in four different datasets: NSL-KDD, KDD-Cup'99, UNSW-NB15, and CICIDS 2017, where the areas under the precision/recall curves were, respectively, 98.56%, 98.44%, 98%, and 96.30%.

Abdullah et al. [62] proposed a multi-class intrusion detector with k-SVM, a combination of k-means with support vector machines. The model presented a detection rate of 99.7%. The best features in the NSL-KDD dataset are obtained after applying Genetic Linear Discriminant Analysis in the pre-processing stage. The Ensemble was implemented through voting.

Pham et al. [63] compared AdaBoost and bagging to build a more effective intrusion detection system. The authors used different classifiers in the tests with the NSL-KDD dataset: decision tree, random forest, decision stump, reptree, and random tree. The authors used two methods to select the best features: information gain ratio and leave-one-out. The best results were obtained with the features selected through information gain ratio and classified with grouping via bagging decision tree classifiers, obtaining an accuracy of 84.25%.

Zhang et al. [64] modeled an intrusion detector that uses a sparse autoencoder network to reduce the dimensionality of the NSL-KDD dataset. An ensemble of neural networks using XGBoost is responsible for classifying the packets. The authors also used the SMOTE technique to create synthetic packages seeking to increase the data of classes with fewer populations. The classification layers were organized in the form of a binary tree. For DoS, Probe, U2R, and R2L attacks, the observed F1-score was 98.86%, 86.02%, 77.89%, and 98.14%, respectively. For regular packages, it was 98.98%.

Zwane et al. [65] compares several simple classifiers with the Ensemble bagging, AdaBoost, and random forest techniques. Using the UNSW-NB15 dataset, the authors obtained

the best result in an Ensemble of random forests reaching 98.1% of the area under the ROC curve. The other classifiers used in the tests were multilayer perceptron, decision tree, Bayesian network, and support vector machine.

Muallem et al. [66] presented an intrusion detector tested on Darpa DDoS, Darpa DDoS Malware, and DoS DNS datasets. The grouping of holding trees classifiers using the Accuracy Updated Ensemble was responsible for obtaining an accuracy of 100%.

Jabbar et al. [67] propose a novel ensemble classifier for intrusion detection based on Naive Bayes and ADTree (a combination of decision tree and boosting), with a majority voting ensemble. NSL-KDD dataset is used in the experiments. Interquartile Range (IQR) is used to remove outliers from the dataset. The average accuracy achieved was 97.11%.

Parvat et al. [68] propose a 2 phase NIDS which applies binary classifiers to solve multi-class classification in a One-vs-All strategy. A decision tree is used for the aggregation of predicted values. Features are selected from the NSL-KDD dataset using domain knowledge. Accuracies of 99.99% and 99.89% were observed for binary and five-class classification, respectively.

Kaur et al. [69] present an ensemble technique using supervised and unsupervised learning for detecting network anomalies and misuse. The experiments were performed in three different datasets, the NSL-KDD, Kyoto 2006+, and KDD-Cup'99. Seven different combinations of base classifiers were tested in a clustering ensemble approach, and K-means + J48 and K-means + RF showed the best performances for all datasets.

Aiming to identify the critical features for the construction of an intrusion detection model with high accuracy, Thaseen et al. [70] propose the use of Chi-Square feature selection and SVM, modified Naive Bayes, and LPBoost classifiers with the majority voting for an IDS. The model's analysis is done with the NSL-KDD dataset. It can detect DoS and R2L attacks with 99% accuracy, probe attacks with 98% accuracy, and u2R with 100% accuracy.

E. WORKS PUBLISHED IN 2019

Moustafa et al. [71] modeled an intrusion detection system with AdaBoost. The authors' focus was on creating a specific model for IoT environments. The selection of features was performed by applying the correlation coefficient between the data from the UNSW-NB15 and NIMS datasets. The Ensemble created was composed of three classifiers: naive Bayes, decision tree, and neural network. The binary classifier obtained an accuracy of 98.97% and 98.29% for the tests on the datasets in the order they were mentioned.

Labonne et al. [72] proposed a cascade intrusion detection system. It is a sequence of neural networks that detect a type of attack in the NSL-KDD and KDD-Cup'99 datasets. Sequentially, the subset resulting from each detection is then processed by the subsequent specialist detector so that, at the end of the cascade processing, it is possible to detect each one of the classes present in the datasets. The measured accuracies were 95.27% for the KDD-Cup'99 dataset and 97.77% for the NSL-KDD.

Liang et al. [73] proposed an intrusion detector that integrates an unsupervised classifier using k-means with a supervised classifier using support vector machines. The proposed Ensemble is given sequentially, where the model initially groups the data with k-means and then classifies them using support vector machines. In tests performed with the NSL-KDD dataset, it was possible to observe an accuracy of 99.45% and a detection rate of 99.04%.

The classifier presented by Gao et al. [74] can balance the size of training samples in order to build an adaptive model of intrusion detection. The authors implemented an Ensemble through weighted voting between decision tree, random forest, k-NN, logistic regression, support vector machines, and deep neural network classifiers. PCA was used to reduce the dimensionality of the NSL-KDD dataset. The accuracy observed in the tests was 84.23%.

Salo et al. [75] proposed an Ensemble for selecting the best features in the Kyoto dataset. The standard features (and) among the information gain, correlation, and significance methods were used to generate a subset processed by the k-means, support vector machine, k-NN, random forests, and quadratic discriminant analysis classifiers. The best results were obtained with k-NN, where accuracy of 82.28% was observed.

Montalbo et al. [76] compared several Ensemble methods: AdaBoost, gradient boost, random forest, and extra tree. With tests performed on a subset generated by selecting the best features and observing the standard deviation between the data from the NSL-KDD dataset, the authors obtained the same results of accuracy, precision, and Recall for the Ensemble methods all of them being 99.9%.

An intrusion detection system focused on DDoS attacks was modeled by Das et al. [77] The authors created an Ensemble through majority voting among the perceptron, support vector machine, k-NN, and decision tree multilayer classifiers. Using the NSL-KDD dataset in tests, it was possible to obtain an accuracy of 99.77% and a false positive rate of 0.23%.

He et al. [78] proposed an Ensemble model for feature selection. According to the authors, the model consists of applying an odd number of feature selection algorithms and, through simple voting, generating a subset containing the model's choices. The experiments used three datasets: KDD-Cup'99, CICIDS 2017, and UNSW-NB15. Four different classifiers were used to observe the improvement in the results before and after the Ensemble application: decision tree, k-NN, multilayer perceptron, and support vector machine. The best result obtained pointed to an accuracy of 99.4%. Another approach along the same lines was taken by Binbusayyis et al. [79] where Ensemble selected the best features using ReliefF, information gain ratio, consistency, and correlation. The authors used random forests for classification. In tests with the KDD-Cup'99, NSL-KDD, UNSW-NB15, and CICIDS 2017 datasets, the accuracy results were 99.97%,

99.89%, 95.87%, and 99.88%, respectively.

The intrusion detection system presented by Lu et al. [80] consists of layering packets classified according to TCP, UDP, and ICMP protocols. For each group of packages, binary classifiers arranged in cascade are applied so that, at the end of the process, it is possible to perform a complete multi-class classification. The authors performed tests with the NSL-KDD dataset using the support vector machine, decision tree, Bayes network, reptree, k-NN, BFTree, SimpleCart, and Naive Bayes classifiers. Compared to Bagging, boosting, and Stacking, the results were superior, reaching 95.33% for detecting DoS attacks, 91.14% for probe attacks, 55.21% for R2L, and 1.42 for U2R, in addition to an accuracy of 98.02% for detecting benign packets.

Illy et al. [81] was motivated to create an IoT intrusion detection system with the lowest possible false-positive rate. Using several possible combinations between k-NN and decision tree classifiers and the Ensemble bagging, boosting, random forest, and voting methods, the authors presented several configuration scenarios and their respective results in tests with the NSL-KDD dataset. The best accuracy values for binary classification were 85.81% (decision tree + Bagging) and for multi-class, 83.83% (voting between k-NN, random forest, Bagging, and decision tree boosting).

Sharma et al. [82] developed a three-stage NIDS with feature selection, weighted extreme learning machines as classifiers, and softmax aggregation. Feature selection is made by extra trees classifier. The model performs multi-class classification. When tested on UNSW and KDD-Cup'99 datasets, it obtained 98.24% and 99.76% accuracy.

Tackling both intrusion detection and feature selection problems, Mousavi et al. [83] propose a method using gradually feature removal, ant colony algorithm, and ensemble DT that presents high accuracy (99.92%) with only 16 features of the KDD-Cup'99 dataset.

Hu et al. [84] propose a method for intrusion detection based on Dynamic Deep Forest. The system is composed of two distinct parts. The first, responsible for feature selection, is built with random forest and Extra Trees, and the other, for classification, uses XGBoost and LightGBM as base classifiers and linear regression for the final output. For training, KDD 99 dataset was chosen. Compared to XGBoost and DBN independently, the proposed model performs better in all metrics, presenting 0.917 precious, 0.862 recall, 0.831 F1 scores, and 0.028 false alarm.

In experiments with the NSL-KDD and UNSW-NB15 datasets, Hsu et al. [85] and Tama et al. [86] presented intrusion detection models using Ensembles either in classification or classification in feature selection. The first grouped, using Stacking in the detection process, the support vector machine, autoencoder, and random forest classifiers. The second used an Ensemble of majority voting for rotation forest and bagging in the classification process and applied a feature selector that makes an Ensemble of particle swarm optimization, ant colony, and genetic algorithm. The accuracy results of the first authors indicate 91.7% with NSL-KDD and 91.8% with UNSW-NB15, while the other authors obtained 85.79% and 91.27% in the same sequence.

F. WORKS PUBLISHED IN 2020

Mixing multi-objective genetic algorithms and neural networks, Kumar et al. [87] presents a new approach for network intrusion detection with a majority voting ensemble. The proposed model performs well in detecting both majority and minority classes and increases the detection rate on both the datasets used for development, reaching 97% accuracy on the ISCX-2012 dataset and 88% on NSL-KDD.

Lian et al. [88] proposes a stacking ensemble of Decision trees and Recursive Feature Elimination for selecting features and classifying intrusion data. Tests are made on the KDD-Cup'99 and NSL-KDD datasets. The method presents 0.9921 average accuracies on the KDD-Cup'99 dataset and 0.9923 average accuracies on the NSL-KDD dataset.

Rajagopal et al. [89] presents in their article a stacking ensemble model using logistic regression, K-nearest neighbor, Random forest, and SVM. The UNSW-NB15 dataset is used for testing binary and multi-class classification, along with the UGR'16 dataset. The best features of each dataset were identified with a combination of information gain and hashing. 97.19 overall accuracy was observed with the UGR'16 dataset and 94.00 with the UNSW-NB15.

Aryeh et al. [90] felts a lack of recent and complete datasets for training effective IDS systems and presented the creation of a realistic dataset (UMaT-OD-20), together with a Multi-layer Stack Ensemble model for intrusion detection. The dataset was created with the help of the ONDaSCA framework and used in the training of the proposed model. Five different machine learning algorithms were used in the creation of the IDS: KNN, decision tree, logistic regression, random forest, and Naive Bayes. The predictions made by these algorithms were learned and fed into another KNN and RF via ten-fold cross-validation, and the final results showed that the model had a better performance than other work referenced in the authors' research, with 97.93% accuracy and 0.22% false alarm rate.

In a differentiated approach to intrusion detection systems, Das et al. [91] proposed an IDS that utilizes natural language processing and supervised ensemble machine learning (NLPIDS) to analyze natural language HTTP requests and detect anomalous traffic in a network without having to learn existing attack methods. The researchers utilized the HTTP dataset CSIC 2010 to train the ML framework, which applies five different classifiers in the first stage (logistic regression, support vector machines, naive Bayes with gaussian function, decision tree, and neural networks) and then feed the obtained results into an ensemble classifier that applies majority voting, LR, NB, NN, DT, and SVM. The performance of each algorithm is then analyzed, and the best one is aggregated to the NLDS. The results of this experiment show a 99.96 detection rate, 0.07% false alarm rate, and 99.96 f1 score.

Concerned with detecting network intrusion but also with the time it takes between identifying them and taking a

measure to avoid any damage, Divakar et al. [92] propose an intrusion detection system using machine learning and analyze its performance not only in regard to accuracy but also training time. The used dataset is the UNSW-NB 15, and it is processed by an ensemble algorithm composed of 10 classifier methods, which are evaluated according to their performances in metrics like accuracy and training time. This ensemble selects the method with faster training time and higher accuracy. Through the experiments conducted, Extreme Gradient Boosting (XGB) showed the best overall results, with 95.54% accuracy and 190 seconds of training.

Fitni et al. [93] proposed an ensemble learning approach as a solution to the detection accuracy, detection time, false alarms, and unknown attacks problems faced by IDS. One of their approaches to reduce the detection time and improve the detection process was to work with the CIC-IDS2018 dataset, selecting only the relevant features according to statistical measurements (chi-squared and Spearman's rank correlation). The choice of classification algorithms to compose the ensemble learning model was made by evaluating their individual performances on the chosen dataset, and the final model was built using gradient boosting, decision trees, and logistic regression, obtaining the following results over the selected data: 98.8% accuracy, 98.8% precision, 97.9% f1 score, 97.1% recall and 00:10:54 execution time.

Khonde et al. [94] focused on the problem of detecting multiple attacks at the same time or detecting attacks that are a combination of other existing ones. The NSL-KDD dataset is processed for feature reduction, using gain ratio, information gain, and correlation coefficient method algorithms in sequence, reducing to 25 features. For the multi-attack signatures, the authors captured some scripted attack packets from the network. The dataset is used for training three classifiers for a bagging ensemble model: naive Bayes, decision tree, and random forest, and this ensemble model outperforms the individual methods (99% precision, 0.09 false alarm rate, and 98.76% accuracy.

Kyatham et al. [95] presented an authorial Ensemble model for two probability-based classifiers. Using MLP and k-NN, the authors performed experiments on the NSL-KDD and CICIDS-2017datasets. In multi-class detection for the datasets mentioned above, the results showed, respectively, the accuracy of 97.05% and 99.68%, a precision of 98% and 100%, a recall of 97% and 100%, and an F1-Score of 97% and 100%.

Also dealing with the problem of unbalanced data, Lin et al. [96] turn their attention to software-defined networks and develop an IDS based on online integrated learning. The ensemble method applied is bagging, and the results are better than those of AdaBoost and C4.5 algorithms, with 72.26% accuracy. But the method also shows great performance fluctuation, with a variation of 20.63%.

Using binary classification, Nandi et al. [97] proposed an ensemble-inspired IDS. To increase classification accuracy, the best 15 features of the NSL-KDD dataset were selected through an ensemble of feature selection techniques: information gain, gain ratio, and reliefF. For the classification step, the J48, decision table, and random forest algorithms were compared and obtained 99.39%, 98.83%, and 99.58% accuracy, respectively.

Using RF, DBSCAN, and RBM as base classifiers, Otoum et al. [98] proposed an ensemble model to improve the classification accuracy for machine learning IDS. Along with the classifiers, the researchers aimed to apply both DBCC and IBCC methods to maximize the system's accuracy. The proposed model works with a stream of data that is divided into two portions that are fed into two identical parallel ensemble methods (consisting of the previously mentioned classifiers). The outputs are combined by BCC-based algorithms, namely the DBCC method. For training, 38 features of the KDDTrain+ dataset were used. Results showed approximately 100% detection and a 1.0 accuracy rate.

Using the H2O python library, Rai et al. [99] presented comparisons between ensemble learning algorithms to evaluate the effect that feature selection has on their performance. Using a genetic algorithm, the author extracted the main 43 features from the NSL-KDD dataset after performing one-hot encoding and standardization. From the three ensemble methods implemented, DRF presented weaker performance with feature selection, while GBM and XGBoost showed improvements. For comparison, a DNN was also implemented, and though it had a worse performance than all ensemble-based algorithms, it also showed improvement with FS.

Rashid et al. [100] tested four different machine learning methods against their proposed bagging ensemble learning model, and theirs showed better performance, with 84.93% accuracy and 2.45% false alarm rate. The system developed by the authors had tree-based algorithms as base classifiers for a bagging ensemble model and was trained on the NSL-KDD dataset. From the dataset, the top 30 features were selected according to the information gain algorithm.

Also working with bagging ensemble learning, Shi et al. [101] proposed an approach using improved extreme trees as base classifiers and combining it with quadratic discrimination analysis through maximization for the final prediction results. The training was conducted on KDD-Cup'99 and UNSW-NB15 datasets, whose best features were selected by Extra-trees. The model outperforms other algorithms on both datasets, reaching 92.88% accuracy and 0.9538 F1 scores on the KDD and 92.45% accuracy and 0.9462 F1 scores on UNSW-NB15, which shows it also has good adaptability.

Yang et al. [102] proposes system uses Gradient Boosting decision tree-Paralleled quadratic ensemble and Gated Recurrent Unit as a way to work with both spatial and temporal data for intrusion detection. CICIDS-2017and CAS2018's (ICN traffic data) best features are selected with random forest and PCA algorithms. The model presents great classification capabilities, reaching 99.9% accuracy for all analyzed classes of the CICIDS-2017dataset.

Zhang et al. [103] apply a T-ensemble scheme to build a Bayesian CNN intrusion detection system in a way that predictions will only be made when the model reaches a

determined threshold of certainty with the output. The system's evaluation is made on the NSL-KDD and UNSW-NB15 datasets, with and without dropping predictions. Both binary and multi-class detection are performed. For binary classification, average accuracies of 99.33% and 99.68% are reached for NSL-KDD and UNSW-NB15 datasets, respectively.

Proposing to improve intrusion detection, Iwendi et al. [104] apply correlation-based feature selection (CFS) together with ensemble classifiers (i.e., Bagging and AdaBoost). J48, RF, and Reptree were tested as base classifiers. The system was tested on the KDD99 and NSL-KDD datasets for binary and multi-class classification, resulting in 0 false alarm rate and 99.9% detection rate for the first and 0.5 FAR and 98.6% DR for the second set. Some attacks have 0 classification accuracy due to small sample sizes.

Aiming to identify both known and zero-day attacks, Khraisat et al. [105] proposed a Hybrid IDS combining signature-based and anomaly-based detection. The SIDS is based on C5.0, and the AIDS uses One-class SVM, and both algorithms are combined through the stacking ensemble method for better performance. The proposed system includes online and offline phases for real-time intrusion detection. NSL-KDD and ADFA datasets are used for training and obtained 83.24% and 97.40% accuracy, respectively.

Mahfouz et al. [106] developed both a new dataset, GTCS, and an ensemble model addressing the issue of accuracy and FAR in IDS. The proposed GTCS is completely labeled and has 84 features. The model has an FS phase, where InfoGainAttributeEval is used. For classification, J48, IBK, and MLP classifiers' outputs are combined with majority voting. With 98.62% accuracy for detecting normal traffic, 98.87% for the botnet, 96.7% for brute force, 98.99% for DDoS, and 88.69% for infiltration, the model outperforms the algorithms used for comparison in the research.

Noticing a lack of comparisons between homogeneous and heterogeneous online ensemble methods, Martindale et al. [107] presented their research on the topic and propose three new heterogeneous models, comparing their performance for online training and how they managed concept drift. The proposed models are the combinations, in pairs, of three base classifiers: SVM, Hoeffding Adaptive Tree (HAT), and Adaptive random forest (ARF). For the final output, weighted majority voting was utilized. The dataset used is the KDD-Cup'99. HAT+ARF had the best performance (approximately 98%) but worst runtime (55.59 s), followed by SVM+ARF (97% accuracy and 44.39 s runtime) and then SVM+HAT, with worse accuracy but faster runtime (94% and 11.16 s).

Feng et al. [108] designed a two-layer integration model based on the stacking framework. The researchers tested different classifiers and ensemble methods to determine which had better performance and if the integration model improved the quality of the results. They also test how many base classifiers are ideal for the first layer of the proposed model. The best features of the KDD-Cup'99 dataset are selected with principal component analysis and used for training. The authors observed improvement with the use of their framework for all the tested algorithms.

Abirami et al. [109] proposed the Least Square support vectors machine intrusion detection system (LSSVM-IDS). The main focus of the model is to identify the most important features through Principal Component Analysis feature selection. The data is then applied to the ensemble phase, which is composed of random forest, linear SVM, Naive Bayes, and logistic regression combined through a stacking classifier. USB-NB15 dataset is utilized, and the model reaches 95% accuracy.

Applying genetic programming, Folino et al. [110] proposed a novel ensemble-based framework for online intrusion detection which takes into consideration the detection of concept drift (E2SC4ID). The framework is constantly updated as the base classifier, and ensemble models are incrementally discovered and replaced when concept drift is detected. An artificial dataset, named Hyperplane by the authors, is utilized along with the ISCX IDS dataset. The best results are obtained with Hyperplane, with a 0.928 AUC score. For the ISCX IDS, the highest score is 0.892.

Rajadurai et al. [111] devised a stacking ensemble model for intrusion detection in wireless networks. The model has a gradient boosting machine and random forest as base classifiers, and the NSL-KDD dataset was used in its development. Feature selection was also made with RF. The overall accuracy of the proposed NIDS was 91.06%.

Bhati et al. [112] uses an extra tree classifier on an ensemble model for network intrusion detection. The extra tree is an ensemble of decision tree classifiers. KDD-Cup'99 and NSL-KDD datasets are tested, and the model achieves 99.97% and 99.32% accuracy, respectively.

Sadiwala et al. [113] conducted a comparative experiment with RF ensemble algorithm and autoencoder to define which has better performance for intrusion detection. The experiments are performed on the NSL-KDD dataset. The AE presents much better accuracy than the RF, with 90.30% and 77.38% scores, respectively.

Wei et al. [114] created a support vector machine-based IDS which uses ensemble learning for performance optimization. The proposed model processes data with a data bag approach instead of a single flow representation. AdaBoost and bagging are combined with SVM for data processing, and the model reaches 90.58% recall, 88.95% precision, 11.24% FPR, and 0.8976 F-score on the Kioto 2006+ dataset.

Applying Deep Learning for intrusion detection, Alkadi et al. [115] implemented a Bidirectional Long Short-Term Memory neural network on the UNSW-NB15 dataset (although they also performed experiments on another dataset that goes beyond the scope of this dissertation). An Ensemble authored method called Deep Blockchain Framework was also implemented. The best accuracy obtained in the tests was 99.41%, with 60 nodes in the middle layer.

Lower et al. [116] performed a series of tests with various Ensemble algorithms on the NSL-KDD dataset. The authors implemented voting (between k-NN, decision tree, MLP, and logistic regression), Bagging, random forest, and AdaBoost

(all using decision trees as classifiers). The mean accuracy, precision, and F1-Score results were 81.2%, 72.5%, and 82% for voting; 85.4%, 77.1%, and 84.7% for Bagging, and the best values of accuracy, precision, and F1-Score for random forest and AdaBoost methods were 86.3%, 78.6% and 85.7% for the first and 86.5%, 78.5% and 85.8% for the latter.

Folino et al. [117] presented a framework for choosing ensemble methods based on accuracy to reduce the false-positive and false alarm rates. A complex combination of decision tree, JRIP Rule Learner, Naive Bayes Tree, Naive Bayes, One-R, Logistic Model Trees, logistic regression, Decision Stumps, and k-NN classifiers was presented in order to dynamically model Ensembles, that is, choosing the best combination among the above classifiers. The tests were performed on the CICIDS-2017dataset and the results presented by the authors document a decrease in computational cost in the data processing. The authors did not report results for intrusion detection.

Jiang et al. [118] proposed a combination of XGBoost with Particle Swarm Optimization. The authors implemented a multi-class intrusion detection system tested on the NSL-KDD dataset. The average precision (AP - averaged precision) obtained in the tests was 90%, 79%, 94%, 15%, and 49% to detect benign packages, Probe, DoS, U2R, and R2L, respectively. The work compares the results with other different Ensemble methods.

Olasehinde et al. [119] performed experiments on the UNSW-NB15 dataset to obtain the best meta-classifier (second layer classifier) hypothesis for an intrusion detection model using Stacking. Performing the classifications of the first layer with k-NN, Naive Bayes, and decision tree, the authors tested some possibilities of meta-classifiers and compared the results. Using three different subsets, the best results were using decision trees in the second layer, pointing to an average accuracy of 98.56%.

Tama et al. [120] presented an intrusion detection system that specializes in detecting web attacks. An Ensemble through Stacking was implemented; however, instead of using conventional classifiers, the authors used other Ensemble methods such as random forest, Gradient Boost Machine, and XGBoost to create the Stackings. Four different datasets were tested: CIC-IDS2017, HTTP-CSIC-2010, NSL-KDD, and UNSW- NB15. The average accuracy result reported by the authors was 96.07% for the proposed model, with the best result obtained in CICIDS-2017with 99.98%.

Mebawondu et al. [121] compare two different ensemble methods to define which performs best with the C4.5 decision tree classifier for network intrusion detection. AdaBoost improved the model's performance more than the bagging ensemble, but both presented better results than C4.5 by itself. The used dataset is UNSW-NB15. The highest accuracy reached with bagging is 97.70%, and with boosting, it is 97.75%.

G. WORKS PUBLISHED IN 2021

With the main focus on IoT networks, Kumar et al. [122] used fog computing to propose a network IDS based on a distributed architecture. The approach uses XGBoost, K-NN, and gaussian Naive Bayes as individual learners and RF for final classification. The system is tested on UNSW-NB15 and the IoT-based DS2OS dataset and performs better on the second one, reaching a detection rate of 99.99% for most of the attack types. The best features of each dataset are selected with mutual information-based feature selection.

In Srivastava et al. [123] the authors apply the Crow-Search algorithm for feature selection and then run the dataset through a Linear Regression, random forest, and XGBoost ensemble for multi-class classification. To ensure the functioning of the model, the UNSW-NB15 dataset is utilized. The proposed algorithm presents results between 62.00 and 100.00 for recall, 85.00 and 100.00 for precision, and 73.00 and 100.00 for accuracy, depending on the class.

The work from Bhati et al. [124] presents an IDS based on boosting ensemble model and using XGBoost. Training and testing were done on the KDD-Cup'99 dataset. The multi-class classification method detects five classes: DoS, Normal, Probe, R2L and U2R, presenting 99.98%, 99.95%, 99.01%, 94.60% and 53.84% detection rates, respectively.

Zhao et al. [125] developed DBN-LSSVM, using a deep belief network for feature reduction and a least-square vector machine for intrusion detection. The KDD-Cup'99 dataset is used for the development of the model. It reaches 99.12 accuracies, 98.54 detection rate, and 0.63 false alarm.

Combining decision tree, random forest, and gradient boosting decision tree, Zhao et al. [126] developed an ensemble model for network intrusion detection. For the ensemble, the authors apply the stacking method. The dataset used was UNSW-NB15. Results showed that the stacking ensemble reached 0.9874 AUC score, 0.9874 F1-score, and 0.0015 FAR.

In their paper, Acharya et al. [127] propose an ensemble machine learning assisted network intrusion detection (NIDS) model for anomalies detection in network flow data. The algorithm they present is enhanced with multiple stages of data preparation, such as data subsampling, normalization, and feature reduction, and for the intrusion detection phase, heterogeneous ensemble learning is applied. The heterogeneous ensemble classifier comprises the algorithms Naive Bayes, J48, k-NN, SVM, Boosting, Bagging, AdaBoost, and random forest, and it performs binary classification as well as multi-class classification. The utilized datasets were the NSL-KDD, KDD99, and UNSW-NB15, and the results presented a near 100% true positive rate (98.8%), with a 0.7% false-positive rate.

Analyzing the behavior of a network's users can be very useful for detecting malicious activity, Ahmed et al. [128] propose an algorithm TABIS to detect the risk level associated with each user and to perform intrusion detection. For selecting the best features of the KDD-Cup dataset used for training the model, an Ensemble Service-Centric Feature Selection

algorithm was used, and for the prediction step, a Sigmoid Recurrent neural network was applied. The model was tested for different numbers o services on the network, and the best results obtained were 93.8% accuracy, a false detection rate of 0.5%, and a time complexity of 14 seconds.

In Ainurrochman et al. [129], the authors utilized the CIDDS-002 dataset to evaluate the performance of an IDS system with two base classifiers (decision tree and Gaussian Naive Bayes) and four ensemble classifiers (random forest, AdaBoost, Bagging, and Stacking). The Bagging ensemble method, when applied in conjunction with the decision tree base classifier, produced the best results, with a mean accuracy of 99.71% and an average f1 score of 100%. The worst performing ensemble was the combination of Bagging and GNB (accuracy of 67.57% and f1 score of 61.80%).

The work Ennaji et al. [130] focused on comparing and evaluating the performance of combinations of five distinct ensemble learning models to maximize detection rates on an IDS. The first combination, MLP-RF-NB, presented the best results with 99.66% accuracy, 99.66% precision, 99.51% recall, and a 99.59 f1 score. They used the ten most important features of the NSL-KDD data set for their study.

The proposal from Khoei et al. [131] presents a comparative research on the performance of ensemble learning (bagging-based, boosting-based, stacking-based) and traditional machine learning techniques (K nearest neighbor, decision tree, Naive Bayes) for anomaly-based intrusion detection. They also compared the performance of ensemble learning paired with different feature selection algorithms. The stacking-based ensemble showed the best performance for both reflection-based and exploitation-based attacks using the CICDDoS 2019 dataset.

Experiments documented in Kiflay et al. [132] proposed an NIDS using ensemble learning and four different classifiers: random forest, AdaBoost, XGBoost, and Gradient Boosting decision tree, with a voting classifier. The proposed system is composed of three different units, and the NSL-KDD and UNSW-NB15 datasets were utilized.

Work documented in Li et al. [133] proposed a sustainable ensemble learning approach for intrusion detection using a modified voting mechanism and passing knowledge among models. The NSL-KDD dataset was used, and the five classes were identified with high accuracy.

In Li et al. [134], the authors evaluated three ensemble techniques (Bagging with random forest, Extreme Gradient Boosting, and Stacking with Generalized Linear Model) on their capabilities to handle imbalanced data for intrusion detection in Cyber-Physical systems. The ensemble models showed better performance compared to traditional decision tree and SVM algorithms, with the stacking methods presenting the best results. They used network packets from the Lemay* dataset for their evaluation. Also, considering the problem of imbalanced datasets, Lin et al. [135] proposes a semi-supervised ensemble approach for intrusion detection by adapting the existing Semi-Boost algorithm. Besides updates to the sampling method, the author suggests that Poisson's distribution be used to divide samples among the classifiers of the Bagging model and that different penalty factors be applied to each base classifier. The base classifiers are decision tree algorithms, and the dataset used for training the model is the NSL-KDD. The results are compared to those of the AdaBoost and C4.5 algorithms and present results worse than the first but better than the former, with an accuracy of 63.88% and precision of 55.41%, among other evaluation indicators.

The problem Nzuva et al. [136] set out to solve is the inability of IDS to detect intrusion in real-time. For that purpose, they trained Naive Bayes, Artificial neural network, K nearest Neighbor, support vectors machine, and C4.5 algorithms on the CICIDS-2017dataset. AdaBoost, bagging, and Stacking ensemble models were also trained with each of the mentioned methods, and all of them performed better than the individual classifiers. The best combination obtained was that of AdaBoost with C4.5 as the base classifier. After these results, the best performing ensemble was tuned for even better performance, achieving 99.06% accuracy, 98.9% Cohen's kappa value, and 100% F score.

To simplify the intrusion detection process, Parkar et al. [137] suggests a system that utilizes the same classifiers for feature selection and for the main classification process and concludes that it resulted in high accuracy rates. The training and testing of the proposed system were done on the NSL-KDD dataset and on additional data captured by the author to simulate real-time attacks. The algorithms used are random forest, K-nearest neighbor, and SVM, applied with the Wrapper Methods for Correlation-based Feature Selection on the FS phase and with a voting classifier for the classification phase. The RF algorithm had a better performance on its own than the other ones, including the ensemble model, presenting a 99.825% accuracy, 99.906% precision, 99.745% recall, and 99.792% F score.

Trying to propose a unique IDS, Seth et al. [138] built an ensemble of classifiers and an F1-score-based ranking system to define which one performed better in detecting each kind of network attack. RF and Principal Components Analysis were used to select the 24 best features of the CIC-IDS2018 dataset. To reach the best algorithms combination, seven classifiers were tested on the accuracy, prediction time, attack detection rate, and time efficiency, and the best two were used in the proposed model: LightGBM and HBGB reached an accuracy of 97.5%, and recall rate of 96.7%.

Using the NSL-KDD dataset, Sidharth et al. [139] propose an IDS and compare the results obtained by using boosting and stacking ensemble techniques. Only 10 of the dataset's features are used, the best according to random forest classifier and Recursive Feature Elimination feature selection. For the stacking ensemble method, AdaBoost and Extreme Gradient are the base classifiers, and logistic regression is the meta classifier, while for the boosting model, the weak learner is decision tree, and the meta classifier is AdaBoost. The boosting ensemble performed better than the stacking, with 80.10% accuracy, 0.824 precision, 0.823 recall, and 0.819 F1

score.

Working with multiple FS algorithms, Jaw et al. [140] propose HFS-KODE, an IDS system that relies on K-means, One-class SVM, DBSCAN, and Expectation-Maximization classifiers on an ensemble. The system presents 99.99%, 99.73%, and 99.997% accuracy for each of the used datasets, CIC-IDS2017, NSL-KDD, and UNSW-NB15. The authors present a lengthy and complete analysis of HFS-KODE and the comparisons between other existing IDS, showing that their Feature selection method can also enhance other algorithms.

In Lin et al. [141], the authors try to solve the problem of classification errors for minority classes by proposing the SMOTE technique for data balancing, paired with C4.5 feature selection, bagging ensemble learning with a random forest as base classifiers, and a majority voting meta-classifier. Both binary and multi-class classification were tested using UNSW-NB15 and CSE-CIC-IDS2018 datasets. For binary classification, accuracy was 99.65% and 99.98% for each dataset, respectively, and for multi-class classification, it was 87.35% and 96.53%.

The work by Wang et al. [142] observes that many times, in order to improve the adaptability of an ensemble intrusion detection model, the added complexity also increases training cost. Addressing the issue, they propose a snapshot ensemble based on group convolution (GCSE). Each snapshot is trained individually, and the final output is obtained through averaging. On both the NSL-KDD and UNSW-NB15 datasets, GCNSE outperforms the other models reproduced in the research, reaching 84.95% accuracy in the first and 79.59% accuracy in the second dataset.

The proposal Subasi et al. [143] by Subasi et al. Subasi et al. [143] is an IDS for a smart healthcare environment, using the Bagging ensemble method with a random forest classifier. The authors use the NSL-KDD dataset. The proposed method performs better than the other single and bagging ensembled classifiers, reaching an accuracy of 97.67%, 0.977 F-measure and AUC, and 0.921 kappa.

Experiments documented in Bertoni et al. [144] work with Optimum-path forest as a base classifier for a stacking-based intrusion detection system. For testing and training, the authors use the NSL-KDD dataset and the uneSPY developed by them. Three experiments were done for the purpose of comparing results: one with single classifiers, one with a homogeneous ensemble, and a third with heterogeneous stacking. It was observed that OFP performs worse as a base classifier for the proposed ensemble methods than as a single classifier.

To deal with Concept Drift, Mulimani et al. [145] developed the Adaptive Extreme Gradient Booster (AXGB), an IDS classifier that uses ensemble learning. For the experiment, the KDD-Cup'99 dataset was used as streaming data. The proposed model reaches 99.078% accuracy.

The authors of Psathas et al. [146] present a hybrid Intrusion Detection System (IDS) that combines 2D Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Multi-Layer Perceptron (MLP). The hybrid model is ensembled using a One-Versus-All strategy named COREM. The research utilizes a recently published dataset called Kitsune Network Attack for their experiments. COREM performs well on the training, validation, and test data, achieving overall accuracy rates of 92.66%, 90.64%, and 90.56%, respectively. However, the system encounters significant difficulties in detecting minority classes.

In Khonde et al. [147], the authors turn their attention to collaborative attacks, devising an ensemble learning-based IDS that should be capable of dealing with this sort of network invasion, along with other single attacks. For the development of the model, the researchers used the UNSW-NB15 dataset, incremented with some collaborative attacks signatures created by them, and reduced with information gain feature selection. The system is based on DT, SVM, and XGBoost classifiers, with a majority voting ensemble. The proposed approach has 97.96% accuracy, 0.07 false alarm rate, 0.85 recall, and 0.99 precision.

The work from Siddiqui et al. [148] points out the computational cost of implementing ensembles of autoencoders and proposes and compares four different methods to decrease the complexity of these types of models, which are then applied to NIDS. The methods differ on how they deactivate the AEs (criteria-based and random) and when that is done (post-training and in-training). IoT network datasets Kitsune and NBaIoT are used for the experiments. The proposed approach speeds the processing of samples 4.2 times.

The proposal from Singh et al. [149] is for a network-based intrusion detection system using boosted tree (adaptive boosting), bagged tree (bagging), subspace discriminant, and RUSBooted classifiers with a voting method. The CIDS 2017 dataset is used. The model outperformed many recent experiments and reduced convergence and execution time.

Experiments documented in Yousefnezhad et al. [150] utilize deep learning for feature selection, SVM (with RBF kernel) and k-NN for classification, and Dempster-Shafer method to combine outputs in their proposed IDS. Eight different combinations of the classifiers are tested, with different k and RBF values. The system's performance is evaluated on UNSW-NB15, CICIDS-2017, and NSL-KDD datasets. Sample selection from de datasets was made through the ensemble margin technique. 90.98%, 98.7%, and 99.80% were the detection accuracy of the model for each of the UNSW-NB15, CIC-IDS2017, and NSL-KDD datasets.

Work documented in Thinh et al. [151] proposes an ensemble of neural networks, along with a high-performing architecture connecting to explore parallel processing. The researchers also generated their own dataset for the development of the model. Among the tested configurations for the NN, the highest accuracy reached was that of 99.98%, with a 0.84% false alarm rate.

H. WORKS PUBLISHED IN 2022

The work presented by Chen et al. [152] introduced an Ensemble model that combines three machine learning classifiers based on Gradient Boosting Decision Tree (GBDT), XG-

Boost, CatBoost, and LightGBM. This model aims to address class imbalance and classify intrusion data in the datasets: NSL-KDD, CICIDS-2017, and ZYELL-NCTU NetTraffic-1.0. Additionally, a multi-layer neural network is incorporated at the output of the Ensemble classifier to predict specific types of attacks. The model evaluation employs a Composite Evaluation Criterion based on the cost matrix and macro F2-score. Furthermore, other evaluation measures, such as F1-Score, Precision, Recall, True Positive Rate, and True Negative Rate, are adopted. In conclusion, the authors assert that their proposed model has the potential to bridge the gap between simulated research and real-world applications. The use of architecture with Multilayer Perceptron (MLP) achieved better rates in the Composite endpoint with 72% compared to Ensemble, which reached 66%. This reflects the balance of accuracy, precision and recall rates, effective for application in the NSL-KDD dataset.

The proposal by Fu et al. [153] is based on an Ensemble feature selection using the ReliefF algorithm and resource clustering, combining four different distance metrics to address challenges of class imbalance and high dimensionality in datasets with the Fuzzy C-Means (FCM) technique. After normalization, the authors apply this approach to 122 features from the NSL-KDD dataset. SVM and KNN are used for classification, and the validation is performed using the metrics of Accuracy and F-Measure. The results show that, compared to traditional Feature Selection techniques, the resource clustering method with Feature Clustering (FCM) needs further research, but it managed to outperform the conventional methods in their proposal using the Ensemble feature selection method.

Hossen et al. [154], the proposal is based on utilizing Bagged Naïve Bayes Decision Tree, Decision Tree, and Random Forest with Ensemble to classify intrusion attacks in the NSL-KDD network attack dataset and compare the performance of ensemble classifiers with base classifiers. PreprocessingPre-processing involves removing missing values, converting categorical data to numerical, and applying data normalization. They applied Random Forest combined with Wrapper filter methods for feature selection, which selected 15 features out of 42. Evaluation of the results used Accuracy, Precision, F1 Score, Recall, and False Positive Rate. The tests concluded that the Bagged Naïve Bayes Decision Tree classifier performed the best among all classifiers, achieving an Accuracy with a detection rate of 99.77% for attacks.

The work from Gosai et al. [155] explores the application of Ensemble techniques to classify intrusions using Machine Learning models. The authors conduct experimental tests on the KDD-Cup'99 and CICIDS-2017 datasets, evaluating the results with metrics such as Accuracy, Precision, Recall, and F1-Score. Pre-processing involves handling missing values, NaNs, data normalization, and a 70% training and 30% testing split. Various Ensemble methods are employed, including Random Forest, Bagging, Ada Boosting, Gradient Boosting, and XG Boosting. The reported results demonstrate that the Ensemble techniques outperform other classifiers, achieving an impressive 99.96% and 99.87% Accuracy on the KDD-Cup'99 and CICIDS-2017 datasets, respectively. The work presented by Zheng et al. [156] aims to develop a two-level model for intrusion recognition in the CICIDS-2017 dataset. The first level involves binary classification to detect attacks, while the second level focuses on multiple categories to identify specific attack behaviors. The types use the CatBoost algorithm and Recursive Feature Elimination (RFE). The model is validated with Accuracy, Precision, Recall, and F1-Score, achieving a remarkable 99.91% accuracy.

In Li et al. [157], the proposal aims to improve Intrusion Detection Systems based on anomaly detection in the CICIDS-2017 dataset. The authors apply SMOTE for class balancing and employ ensemble with the bagging technique, resulting in a promising performance with XGBoost. Macro metrics for Precision, F1-Score, Recall, and ROC Curve achieve rates of 93.2%, 95.5%, 98%, and 99.4%, respectively. The authors highlight the improved performance with SMOTE compared to the imbalanced data.

The work by Sun et al. [158] proposes an Ensemble classification model with bagging for intrusion detection in e-learning platforms. Random Forest is trained, combining multiple tree models to form a stronger ensemble model using a public library platform dataset. The classification is validated with Accuracy, Precision, Recall, F1-Score, Confusion Matrix, and ROC Curve, resulting in 99% accuracy, 100% recall, 99% F1-Score, and 99% ROC Curve.

In Tayde et al. [159], the authors address the challenge of high computational time with datasets containing many features. They suggest choosing datasets with relevant features to achieve high accuracy and low computational cost. The CICIDS-2017 dataset is selected, and an Ensemble approach with Gain Ratio and Pearson Correlation is used to determine the top thirty features. Classification with Random Forest and the reduced set achieves an accuracy rate of 99.83%.

The work by Siddharthan et al. [160] focuses on detecting DDoS attacks in the MQTT network protocol used in IoT contexts. The authors simulate a network with Raspberry Pi devices and sensors to generate a dataset called SEN-MQTTSET, containing MQTT and TCP protocols infected with flooding attacks. Ensemble Multi-View is used for feature selection, and five algorithms (KNN, RF, NB, SVM, GB, and DT) are employed for attack detection. The results achieve 99% accuracy, and the best classifier is identified.

In Yang et al. [161], the authors utilize neural networks for intrusion classification and identification with real-time data from a Software-Defined Network (SDN). Cluster analysis is employed for feature reduction, and ROC Curve is used as the evaluation metric. A comparison with the IDS Suricata demonstrates 99% AUC values with low latency in the results.

The work by Wu et al. [162] addresses the limitations of intrusion detection systems in adapting to changing network data performance over time. They propose a dynamic incremental learning intrusion detection method using an ensemble with the RVM algorithm, similar to SVM but providing a probability distribution based on scores. The KDD-Cup'99

and CICIDS-2017 datasets are used for experiments, showing that the proposed method is effective, achieving a 99% detection rate for attacks.

The work presented by Das et al. [163] applies the Ensemble technique for feature selection with majority voting, known as EnFS, on the datasets NSL-KDD, CICIDS-2017, and UNSW-NB15 after cleaning and normalization. For data balancing, they selected the data type (benign or attack) with a minimum record count, choosing the same quantity for each class. Through this technique, they achieve a supervised structure named SupEnML for training with five individual models (Logistic Regression, Decision Tree, Naive Bayes, Neural Network, and Support Vector Machine). The dataset is divided into a (70-30) proportion, and an ensemble is applied with individual classifier hyperparameter tuning. The model's evaluation is based on Accuracy, Precision, False Positive Rate, Recall, F1-Score, and the ROC curve. The results show that the proposed model outperforms individual models by more than 85%. In the NSL-KDD dataset, the EnFS model achieves 100% accuracy outperforming all other results. The CICIDS-2017 set had the lowest rate of false positives, 0.006 seconds, only to the LASSO method with 0.005. The authors also conclude that the model showed the best F-1 and Precision metrics performance compared to all other experimental configurations used in the work.

In Long et al. [164], the authors propose an intrusion detection model using Autoencoders to reduce training time with satisfactory results. They use the NSL-KDD dataset and apply the One-Class SVM to create an unsupervised dataset with low dimensionality. The process involves three steps: feature selection with RFE to identify the most relevant features, applying the One-Class SVM, and then using the Autoencoder to address the sample imbalance. The outcome is an optimized training and testing time with accurate attack detection. The model's validation uses metrics such as the Confusion Matrix and the ROC curve. After the analyses, they conclude that when the abnormal data are almost equal to the normal ones in the data set, the result is bad, but when the data have a high imbalance, the detection result is good. Overall recall proved to be better than KitNet's main algorithm in Kitsune.

The work by Yao et al. [165] addresses the challenges of data imbalance and low accuracy. To tackle this, the authors employ an ensemble with the classifiers XGBoost, LightGBM, and Random Forest on the UNSW-NB15 dataset. They propose a two-layer soft-voting algorithm for binary and multiclass classification, utilizing SMOTE for data oversampling. To verify the quality of the results, they use the Confusion Matrix and the ROC curve. As a result, significant improvements are achieved in the detection rates of intrusions, such as DoS, ShellCode, Worms, and Reconnaissance attacks. The authors report that they improved the accuracy results with the soft-voting ensemble model by 0.45%, 0.11% and 0.12% compared to the RF bases algorithms, LightGBM and XGBoost, respectively. And with the application of SMOTE to deal with the unbalanced data of minority classes in the multiclass classification, it grows 141%, 16.2%, 8.8% and 2.6% in DoS detection, ShellCode, Worms and Reconnaissance, respectively.

The work presented by Thaker et al. [166] proposes an intelligent framework for intrusion detection in autonomous vehicles using Machine Learning with Ensemble. The study utilized the CICIDS-2017 dataset and attack traffic generated in a Controller Area Network (CAN) and captured using the Wireshark tool. Various classifiers are employed, including Decision Tree, Extra Tree, Random Forest, XGBoost, SVM, and KNN. Firstly, the authors perform pre-processing-processing by eliminating missing values, normalizing the data, and balancing the classes. Then, the classifiers are applied, and the results are evaluated using the Confusion Matrix. The best results are achieved with the XGBoost classifier, achieving an accuracy rate of 98.57%.

The work presented by Li et al. [157] involves generating a real intrusion dataset collected from routers and cameras, incorporating records from the KDD'99, UNSW-NB15, and CIC-IDS datasets. Afterward, they apply the technique of Bayesian Clustering, which combines principles from graph theory, probability theory, computer science, and statistics, enabling them to estimate patterns of similarity. Subsequently, they adjust the clustering using fuzzy clustering techniques, which provide an output indicating the probability of a point belonging to a specific cluster. For classification, they utilize a three-layer Denoising Autoencoder neural network with ReLU activation, an SVM with an RBF kernel, and k chosen as 5 in kNN. Additionally, fuzzy decision trees with a maximum height of 10 are incorporated in Random Forest. Finally, they validate the model using Precision, Recall, F1-Score, and the ROC Curve metrics. In Li et al. [167] proposes a Machine Learning Framework for detecting attacks in the Internet of Vehicles using an ensemble with three classifiers (XGBoost, LightGBM, and CatBoost) on the CICIDS-2017 dataset. The authors begin by classifying and training each attack class with the classifiers to determine the best classifier for each class individually, validating the results with the F1-Score metric. Subsequently, they employ the best classifier to recognize each type of attack in the dataset, conducting the final prediction. As a result, they achieve F1-Score scores of 99.99% for intrusion detection.

The proposal from Amarudin et al. [168] is based on a Machine Learning model that employs Ensemble with Bagging-SDN technique. In the proposal, Support Vector Machine (SVM), Decision Tree (DT), and Naive Bayes (NB) algorithms are selected to integrate the model applied to the UNSWNB15 dataset. The experiments documented in the study achieve an Accuracy of 80.79% compared to 75.89% achieved by a single classifier. Moreover, they attain improved True Positive and False Negative Rates using Bagging-SVM=6, Bagging-DT=10, and Bagging-NB=2 estimators.

The work presented by Data et al. [169] expresses that training the model using all the features of the dataset results in a computationally expensive model. To address this, they

propose an improved Ensemble model named AB-HT. This model combines AdaBoost and Hoeffding Tree, achieving F1-Score rates that are 18% higher than models trained without the Ensemble method. Additionally, they report that their model reduces training time due to the application of optimal hyperparameter adjustments on the CICIDS-2017 dataset. Still, the experimental results showed that the proposed model obtained higher training times with the AB-HT model than the batch learning model, and slower than the AdaBoost and Decision Tree models.

In Behravan et al. [170], the author's proposal aims to build a model for detecting five types of attacks on a vehicular network in the VeReMi dataset. Thus, they use Ensemble with the Stacking method in classifications performed with Neural Networks with the k-nearest neighbors (kNN), Decision Tree (DT), and Naive Bayes (NB) algorithms. For training, they select ten trained submodels, with 10% of each model containing two hidden layers, which are loaded for testing in the untrained part of the submodel. In conclusion, they state that Ensemble Staking results in better results, but they consider that data preparation is a main part of the process.

The proposal from Danso et al. [171] aims to develop an Ensemble-based model using data captured from the CICIDS-2017 and N-BaIoT datasets. To achieve this, the authors employ the classifiers Naive Bayes, Support Vector Machine, and k-Nearest Neighbors, applying the three Ensemble techniques (Bagging, Boosting, and Stacking). For feature selection, 30 characteristics are chosen for each dataset classification. In the results, the authors report that CICIDS-2017 outperforms the N-BaIoT dataset with 99.87%, 99.36%, and 99.34% accuracy, recall, and F1-Score, respectively, using the Stacking technique.

The work presented by Khan et al. [172] conducts intrusion attack classification using Deep Machine Learning techniques with Artificial Neural Networks and Ensemble Stack. For the execution, 4 Neural Networks with 6 layers each are employed, and their outputs feed into another Neural Network with an additional 3 layers that finalize the classification on the NF-UQ-NIDS-v2 dataset. The proposed model achieves an accuracy of 98.40%.

The work presented by O'Meara et al. [173] emphasizes the significance of measuring the False Negative Rate (FRN) in a Machine Learning model for intrusion detection. They propose a model using the InSDN dataset derived from a Software-Defined Networking structure containing 8 classes of attacks. To ensure balanced data, SMOTE is employed, and the dataset is prepared for binary classification. Feature selection uses Random Forest to extract important features, reducing the initial set of 80 features to 23. Various algorithms, including XGBoost, CatBoost, and Naive Bayes, are applied for classification. Data normalization is performed, and the dataset is split in a 70-30 ratio for training and testing. The results indicate that the Ensembles with Random Forest and XGBoost outperform other tested methods, demonstrating the best FRN. Also, when comparing the performance of the training sets with ensembles with the individual classifiers, they found that the ensemble classifications had better or equal performance. The best result was for the set trained with data balanced with SMOTE.

The work presented by Srivastava et al. [174] proposes the use of Ensemble with Support Vector Machine (SVM), Naive Bayes (NB), and Decision Tree (DT) algorithms for identifying intrusion attacks. The model is trained on the NSL-KDD dataset using the Stacking method. In their approach, the authors ensure security by employing a middleware that combines the Diffie Hellman function and SHA-256 to generate a hash message, thereby guaranteeing the integrity of the model by storing the IDS node information hash values. The achieved accuracy is 86.97%, with a classification time optimization of 35.60%.

The proposal from Oliveira et al. [175] addresses the challenge of finding the correct method for Ensemble classification, particularly in the context of Machine Learning in IoT. To tackle this issue, they present a Deep Learning-based classification approach. They construct an Artificial Neural Network using the Adam optimizer to apply penalties and achieve good results. The dataset used in their study, TON_IoT, is split into 40% for Train1, 30% for Train2, and 30% for testing. The results show gains of up to 1.5% in metrics such as Precision, Recall, and F1-Score compared to the classification using a single reference classifier. They also cite that by analyzing the raw records of the TON_IoT data set, they found that including new features or using the whole group can optimize the classification of the ensemble model and increase the statistical reliability of their proposed results. The accuracy of the ensemble model was approximately 1.1% higher than the base models.

The proposal from Mbasuva et al. [176] aims to leverage the popularity of Software-Defined Networks by applying Ensemble Stacking with Deep Learning techniques using combined Neural Networks (CNN, RNN, and DNN) to detect DDoS attacks in the CICIDS-2017 dataset. The model's evaluation used the Confusion Matrix, loss function, precision, and ROC Curve. The authors report achieving favorable results in comparison to the use of isolated techniques.

In Lucas et al. [177], the proposal aims to construct an ensemble stacked classifier using a combination of the k-NN, SVM, MLP, and DT algorithms for pruning in search of reduced computational cost and lower classification error rates for intrusion detection attacks. The authors utilize 6 subsets of data taken from the CICIDS-2017 dataset for experiment validation, where the proportionality of resources is preserved, data is normalized, and two classes (benign and attacks) are defined for binary classification and application of classifier diversity pruning to reduce training and testing time and achieve better performance. The results are validated using accuracy, F1-score, precision, and recall metrics, showing satisfactory results compared to similar studies and presenting optimized classification time with reduced computational cost. According to the authors, the best detection results were 99.94%, 99.90%, 99.95% and 99.95% for metrics of Accuracy, F1-Score, Precision and Recall, respectively, surpassing

the results of related works, besides the improvement in the computational cost.

The proposal from Houda et al. [178] aims to detect intrusion attacks in an SDN network built using the Mininet tool. In this context, they perform Boosting Resource Selection, which classifies resources based on an importance score, selecting the best help from the NSL-KDD and UNSW-NB15 datasets. Following the resource selection, they perform attack classification using Ensemble and Lightweight Boosting Algorithm (LBA), validating the model with accuracy, precision, confusion matrix metrics, and an analysis of the ROC Curve. As a result, they achieved 99% accuracy in 60 seconds of training with the model on the UNSWNB15 dataset and 96% precision using the NSL-KDD data trained within 14 seconds.

In Iacovazzi et al. [179], the work presents a Machine Learning model for intrusion detection in cloud containers and kernel-level isolation using Ensemble with Random Forest (RF) to compare the results of Support Vector Machine (SVM) and Local Outlier Factor (LOF) algorithms. The LOF calculates a score for each data point, indicating the density deviation from its neighboring data points. The RF's output is combined through a bagging meta-learning algorithm for attack classification. Data collection involved initializing a VM with all the necessary Docker packages and images and using the "perf" tool to record system calls at the VM level in a controlled environment, resulting in a dataset of 10,000 samples for each class. The results demonstrated the model's ability to detect attacks even without labels in the training set, showing superiority over SVM and LOF-based models, which was confirmed using ROC Curve and True Positive and False Positive rates metrics. The set of RF with isolation (EoF) with graph-based features achieved better results with an average of 0.024 FPR, while SVM and LOF presented 0.108 FPR, which are also confirmed in graphs of the RoC curve given by the authors. However, there are two attacks (Malicious Script and SQL Injection) in which the model based on SVM surpasses RF in the detection rate. However, the accuracy and F1 are lower due to the high FPR obtained by the model based on SVM.

The work presented by Abdulkareem et al. [180] introduces five different models using Ensemble Learning Classifiers on the Bot-IoT dataset, focusing on Bagging and Boosting techniques with the algorithms LightGBM, Random Forest, CatBoost, and XGBoost. The performance of these proposed models was evaluated using metrics such as Accuracy, Precision, Recall, F1-Score, and each model's training and testing time. The results indicated that all five classifier models achieved over 95% overall classification accuracy, demonstrating their effectiveness in correctly classifying attack instances. Regarding classification time, the CatBoost algorithm exhibited the best training time of 3.8 seconds, followed by RF with 4.7 minutes. The XGBoost classifier took more than an hour for training, making it the slowest, while the prediction tests were the fastest, taking only 25 seconds. The best results obtained were with the CatBoost classifier reaching Accuracy 99.99; Accuracy 99.89; Recall 99.73; F1 99.81; Workout time in seconds 229.42s; Test time in seconds 1.84s. In comparison with the performance of five tested machine learning classifiers, Ensemble was the one that presented the best results.

In Zhang et al. [181], the proposal focuses on intrusion detection, where Machine Learning and Deep Learning techniques are employed to classify the NSL-KDD and UNSW-NB15 datasets. The modeling process includes the utilization of Random Forest, XGBoost, and Deep Neural Network (DNN) for both binary and multiclass classification. Before classification, data pre-processing-processing is conducted, involving data cleansing, conversion of object-type values to numeric types, data normalization, and feature selection using the SelectfromModel method with a correlation threshold. Thus, after feature reduction, the classification yields promising results for both binary and multiclass models, achieving accuracies of 96.30% and 83.88% for NSL-KDD and UNSW-NB15, respectively.

The proposal from Liu et al. [182] combines neural networks (CNN, DNN, and RNN) and a dynamic voting mechanism using Ensemble Voting. For each probability value from CNN, DNN, and RNN, a blockchain protection mechanism is employed, where communication between the blocks collaboratively records and stores a hash to form a blockchain. This ensures privacy and prevents data modification, resulting in a system with high security and intrusion detection performance. The authors also mention that most traditional machine learning methods only learn superficial information from the data due to the absence of hidden layers, as present in the proposed neural network model. In the experiments, 500 training rounds were performed for each model, and the precision and loss rate obtained from each training round were recorded in memory. The best rates achieved using the CICIDS-2017 dataset, chosen for containing information from recent network devices for a well-performing model, were used as the results. Accuracy, F1-Score, Precision, and Recall evaluation metrics were utilized, achieving rates of 99.98%, 99.97%, 99.96%, and 99.98%, respectively, in identifying port-scan attacks.

In [183], the proposal addresses the challenge of constructing an intrusion detection system with high true and low false positive rates for recognizing cyber-attacks. They present an ensemble model that utilizes weighted classification probabilities derived from base classifiers. The hyperparameters are optimized and extracted from two datasets: KDD99 and CBTC (hardware-in-the-loop simulation platform). The results demonstrated strong performance, achieving a recognition rate of 98.6% for attacks with a false positive rate of 1.3%.

The proposal from Rajadurai et al. [184] incorporates blockchain to ensure privacy and prevent data modification, resulting in a system with high security and intrusion detection performance for different probabilities of CNN, DNN, and RNN. The work also mentions that most traditional machine learning methods only learn superficial information

from the data due to the lack of hidden layers like those in the proposed neural network model. In the experiments, 500 training rounds were conducted for each model, recording the accuracy and loss rate obtained from each training iteration. The best rates achieved using the CICIDS-2017 dataset, selected for its recent network device information, were stored in memory, yielding a well-performing model. The evaluation metrics used in the results were Accuracy, F1-Score, Precision, and Recall, achieving rates of 99.98%, 99.97%, 99.96%, and 99.98%, respectively, in identifying port-scan attacks.

In Abdoli et al. [185], the proposal revolves around applying Ensemble Stacking along with various CNN models trained at different temperatures where the neural networks' outputs are combined, serving as input to the ensemble with the stacking method. The authors also employ the technique of soft labeling to indicate the data class associations during the training phase with the NSL-KDD dataset. The results yield an accuracy of 90.18%.

The proposal from Rashid et al. [186] presents a learning algorithm to address the Non-Stationary Time Series Prediction (NS-TSP) problem, using six-time series datasets to evaluate the model's performance. The authors apply the Extreme Learning Machine with Kernels (ELMK) technique to solve NS-TSP problems. They utilize Cross-Validation to select the best values for parameters such as the time window size, subset of data size, and choice of kernel function in ELMK. Model validation is performed using performance measures like Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Absolute Error, and Standard Deviation (Std). In conclusion, they report that the model achieves good results, but further in-depth studies are needed to address multi-step advance NS-TSP and multivariate NS-TSP problems. The attack detection rate, Recall and Accuracy with a stacked ensemble, compared to the other classifiers tested in the work, proved to be superior, reaching 91.06% of Accuracy. The proposed stacked ensemble model is suitable for the intrusion detection system to detect the attacks.

Experiments documented in Zahoora et al. [187] introduces a two-stage ransomware detection system employing concepts of Zero-Shot Learning (ZSL), Deep Convolutional Autoencoder (DCAE), and Ensemble techniques. The DCAE-ZSL method aims to train a neural network with 10 hidden layers to generate descriptions independent of known and unknown attack classes (zero-day). The results show a significant trade-off between False Positives and False Negatives compared to the simple prediction models of individual algorithms. The authors report considerably improving the system's performance with the proposed approach. Compared to existing techniques, the proposed model outperformed other classifications such as LR and ResNet50 model Deep CNN, reaching Recall of 95%, TFP of 13%, and MCS calculation of the sum of error measures FP and FN of 19%, proving to be well designed for zero-day ransomware detection.

The work presented by Huang et al. [188] aims to recognize backdoor attacks by constructing a model capable of performing Ensemble Voting classification along with tree-based techniques, which incorporate two algorithms for black-box and white-box configuration purposes. They also perform knowledge extraction to detect outliers, measure computational time, and assess the effectiveness and limitations of some defenses. The three datasets used in the study include two from the UCI Machine Learning Repository and one from Kaggle called Microsoft Malware Prediction. Upon concluding the evaluation, their experiments reveal computational vulnerabilities between incorporation and knowledge extraction, raising security concerns regarding using tree-based classifiers, as they are more susceptible to attacks than defenses. The authors recommend adopting better and more effective backdoor detection methods. The proposed model can identify enhanced knowledge calls as outliers producing satisfactory results. In this way, the authors reach 100% accuracy values in the datasets that use the black box method and optimized time ranging from 0.117 seconds to 15261 seconds. The tests carried out with a white box model also obtained 100% accuracy and times ranging from 0.056 to 1831 seconds, depending on the amount of data in each tested set.

In Yu et al. [189], the proposal involves constructing an incremental learning algorithm with a dynamic weighting ensemble (DWE-IL) to address multi-step and multivariate non-stationary time series tasks (NS-TSP). This is achieved by combining extreme learning machine (ELM) models and extreme learning machine with the kernel (ELMK) models. The authors implement dynamic weighting rules and updated weight incrementation to forecast non-stationary time series and identify abnormal patterns. The model's validation compares five algorithms, using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) as evaluation metrics. The tests are performed on three stock market index datasets, namely Dow Jones Industrial Average Index (DJI), Nikkei 225 Index (N225), and Shanghai Stock Exchange Composite Index (SSE), all obtained from Yahoo Finance. According to the authors, the results obtained using their proposed algorithm demonstrated satisfactory performance, outperforming the other compared algorithms in prediction accuracy. The DWE-IL algorithm does not significantly improve generalization performance at the 5% significance level compared to other algorithms.

The work from Qiao et al. [190] addresses the issue of data imbalance in traditional methods, which may lead to classifier learning failures. They aim to develop a metric learning algorithm for imbalanced data, called Large Margin Nearest Neighbors Balance (LMNNB), which minimizes the distance between samples of the same class. To test the algorithm, they utilize ten reference datasets from the UCI Machine Learning Repository, applying PCA for feature selection and Ensemble Voting with SMOTE and Random Under Sampling to generate and remove an equal number of minority or majority samples, thus balancing the dataset. Classification is performed using five algorithms. For validation, the authors use the ROC curve metric to compare their method with

two balancing algorithms, EasyEnsemble and RUSBoost. According to the authors, the proposed method outperforms state-of-the-art classification algorithms. Compared with the MMC algorithm proposed in related work, the improvement is 130.2%, 43.1%, 90.5%, 96.4% and 88.7% in Precision, Recall, F1, GM and AUROC, respectively. The proposed algorithm obtained better results in all sets tested in the work.

The proposal from Fu et al. [153] addresses issues related to semi-supervised anomaly detection, particularly focusing on imbalanced class distributions. The authors introduce two outlier-sensitive measures to assess the accuracy of classifiers on evaluated datasets and to evaluate the overfitting of existing dynamic ensemble models on samples classified as normal. In this context, when obtaining a classification output, the proposed measures provide positive scores to compensate for the base classifier's correct classification of validation samples. The two measures used are Outlier-Sensitive (OOS) based on the output and Cost-Sensitive Outlier-Sensitive (CSOS). Through experiments, the authors conclude that the unified model of these algorithms can achieve improved performance based on F1-Score and G-mean metrics. Still, it may exhibit shortcomings on certain datasets due to the performance cost of handling outliers to identify features of the normal class, resulting in an increase in false positive rates. With the EM classifier and DESHF method that dynamically selects only base classifiers belonging to the most frequent interval, regardless of their scores, the average G-Mean was 0.5441 and F1 0.2951. With OOS and the DESST method using Friedman's non-parametric test and Nemenyi's non-parametric test to select base classifiers, the mean G-Mean obtained 0.6160 and F1 0.3386. In conclusion, when using EM for competence estimates of base classifiers, DESHF outperforms other methods for selecting base classifiers in terms of G-Mean and F1, while DESST outperforms other methods, except DESHF.

Experiments documented in Yang et al. [191] is based on a hybrid feature selection combined with an ensemble classification approach for identifying intrusion attacks in complex and imbalanced traffic data. The method utilizes the NRS algorithm to reduce the dimensionality of discrete data and the SSA algorithm to introduce a degree of dependence in each neighboring feature, resulting in a reduced set of characteristics. The classification is performed using 5 integrated base classifiers (Decision Tree, KNN, Random Forest, and eXtreme Gradient Boosting), forming a model named M-Tree and 10-fold cross-validation with configured and tuned parameters. The experiments are conducted on NSL-KDD, UNSW-NB15, and TON_IoT datasets, and the results are evaluated and compared using metrics such as Accuracy, Precision, Recall, and F1-score. According to the authors, the NRS-SSA feature selection algorithm demonstrated excellent performance, outperforming other existing methods such as CFS (Correlation-based Feature Selection), IG (Information Gain), GA (Genetic Algorithm), and PSO (Particle Swarm Optimization). Moreover, the proposed M-Tree classification algorithm showed effectiveness, surpassing similar homogeneous ensemble algorithms to M-Tree. In the NSL-KDD dataset, the proposed adaptive ensemble model obtains the highest accuracy and F1 score of 87.78% and 86.55%, respectively. The UNSW-NB15 dataset has an accuracy of 81.54% and an F1-score of 79.67%. And in the TON_IoT dataset showed good accuracy and F1 scores of 98.97% and 98.99%, respectively. Compared with other classification models and intrusion detection methods, the proposed work has a significant advantage in accuracy.

The proposal from Devprasad et al. [192] highlights that intrusion detection is a complex activity due to the automation generated by new attacks, which necessitates the application of machine learning. Accordingly, they propose a model applied to two datasets, NSL-KDD and UNSW-15, performing feature selection at two levels. The first level employs the Chi-Square algorithm, while the second level uses BA, a heuristic capable of producing favorable results. Subsequently, the output of four base classifiers (DT, SVM, LR, and NB) undergoes processing to choose the classifiers based on TOPSIS, an algorithm with complex decision-making processes capable of handling many attributes and generating an alternative ranking. In the experiment results, the authors conclude that for the UNSW-15 dataset, the classifiers (DT+SVM) exhibited the best results with 89.43% accuracy and 3.215% False Positive rate. For the NSL-KDD dataset, using DT as the sole base classifier resulted in the best performance, achieving an accuracy of 98.77% and a low False Positive rate of 0.03%.

In Krishnaveni et al. [193], the authors present a proposal in which an Ensemble Voting Classifier is employed, incorporating four algorithms, SVM, LR, NB, and DT, for intrusion detection in Cloud Computing. The classification is applied individually to three datasets (Honeypots, Kyoto, and NSL: KDD). During the data preparation phase, a rigorous feature selection is conducted using three techniques (gain-ratio, chi-square, and information gain). To validate the results, the authors utilize metrics such as Accuracy, Precision, and Detection Rate derived from the confusion matrix, F-measure, and ROC for a comparative evaluation through descriptive and inferential statistical analysis of the chosen datasets. The authors report that the proposed model pre-processes-processes data optimally, requiring low training time and achieving accurate results. Ultimately, they assert that deep learning techniques and resource optimization lead to more efficient intrusion detection in Cloud Computing. In the results, the authors compared accuracy and performance in the original and reduced datasets. The results of the proposed method clearly showed greater accuracy with reduced data than the original datasets. The study with the three datasets pointed out that the proposed method improved with the honeypot dataset obtaining the best accuracy of 98.29% and a reduction of false positives by 0.012%, respectively.

The proposal from Le et al. [194] aims to reduce the complexity of IoT traffic to improve intrusion detection. The solution proposes applying an FPA algorithm (enhanced flower pollination) and the Ensemble technique. According to the authors, this algorithm enables an ideal feature selection

through better data convergence. After data cleaning and normalization, feature selection is performed on two datasets, UNSW-NB15 and NSL-KDD. Subsequently, the training is carried out using Ensemble Voting with SVM, Decision Tree, and Random Forest classifiers for multiclass classification. The validation of results includes various metrics tested on both datasets, with the most satisfactory being the intrusion detection accuracy of 99.32% in UNSW-NB15 and 99.67% in NSL-KDD.

Experiments documented in Gangula et al. [195] aims to enhance intrusion detection performance in large IoT datasets by utilizing an Ensemble technique with Decision Tree and Random Forest classifiers, which do not require high hardware costs for training. The authors employ three extensive datasets, namely IoTID20, NF-BoT-IoT-v2, and NF-ToN-IoT-v2, and apply the SHAP game theory approach to explain individual predictions and demonstrate the contribution of each predictor/feature toward positive or negative output values. The validation of results involves Accuracy, F1-score, and ROC curve metrics. The proposed ensemble methods achieve 100% precision and F1-score performance on the datasets; however, they exhibit lower AUC measurements on the NF-ToN-IoT-v2 dataset. The utilization of the SHAP method provides valuable insights through a heat map, enabling cybersecurity experts to make more optimized decisions when dealing with large datasets.

The proposal from Bu et al. [196] addresses the limitations of simple learning methods in fully extracting features within a dimensional space to detect intrusions in Databases. The authors suggest enhancing the performance by employing various techniques using neural networks combined with heterogeneous topologies (RBAC) and a System (CN-LCS) that enables robust learning of input features focused on optimizing data accuracy. Through intrusion detection based on RBAC and comparison with single-model methods, the authors demonstrate the statistical significance of the constructed model using 10-fold cross-validation and chi-square validation. Ensemble-LCS guided by diversity can be applied for improved IDS if combined with RBAC-based classification. The accuracy results of the trained Ensemble-LCS Models reached 0.6640 in the function of 10 (queries with Role 10 (Trade-Result) are split based on queryMode feature) and an accuracy of 0.8874 with other accuracy functions surpassing single-model LCS.

In Jung et al. [197], the authors highlight the challenge of imbalanced data, leading to overfitting and impeding accurate classification. To overcome this issue, they present a mixed resampling method that combines a hybrid synthetic minority oversampling technique with a neural network to augment the minority class and eliminate noisy data, resulting in a balanced dataset. The proposed approach involves the use of SMOTE+ENN alongside Ensemble Bagging, employing RF, MLP, and XGBoost models to further enhance classification performance on two intrusion detection datasets: PKDD2007 (balanced) and CSIC2012 (imbalanced). Evaluation is conducted using metrics such as Accuracy, G-Mean, F1-Score, ROC Curve, and Confusion Matrix, demonstrating the influence of sampling methods on binary and multiclass classification. The results confirm the effectiveness of the proposed technique in accurately identifying attacks and achieving excellent true positive classification. The proposed method yielded a 4% higher F 1-score than other traditional resampling methods using SMOTE or similar techniques. The G-Mean and accuracy results were much more stable with Smote+ENN+Ensemble bagging, resampling of datasets to handle imbalanced data performed best on model results and found that of all the balancing techniques, the ones that produced the most stable results were the hybrid techniques SMOTE+ENN and SMOTE+TOMEK for both a binary classification and for multi classes. Furthermore, the proposed method improved the identification and classification of true positives of serious threats that rarely occur.

The proposal from Alghamdi et al. [198] addresses the challenges of DoS, DDoS, and service scanning attacks in IoT networks, where intrusion detection systems require precise identification. As a solution, they present a deep machine-learning approach to construct an Ensemble classifier with Integrated Evaluation Metrics. The authors emphasize the importance of detection time in discovering attacks and the need to integrate metrics that determine model efficiency. They utilize the F-score, Cohen's Kappa, Matthews Correlation Coefficient (MCC), and log loss as evaluation metrics. The experiments are conducted on three datasets (UNSW-NB15, Aposemat IoT-23, ToN_IoT). Pre-processing involves data cleaning, removing low-value features for learning, and selecting features using CatBoost and LightGBM. The results indicate significantly reduced training and processing time without compromising attack identification efficiency, achieving an accuracy of 99.45% for binary classification and 97.81% for multiclass classification.

Experiments documented in Zhang et al. [181] address the growing network traffic and its association with increased cyber-attacks. The proposal from the authors revolves around an automatic feature selection method using an ensemble, employing various basic approaches for feature selection. They sequentially select features based on their importance using a dynamic feature selection method called NSOM (Nested Selection of Features Optimization Method) and validate it on the UNSW-NB15, CIC-IDS2017, and CSE-CIC-IDS2018 datasets. The developed way achieved higher precision and lower false positive rates than other ensemble strategies in the literature. They reached the best results on the CICIDS-2017dataset, they completed the best results, with 99.92% accuracy, 99.92% F1 score, and a false alarm rate of 0.08%

In Mhawi et al. [199], the authors developed an IDS model based on an ensemble with a hybrid feature selection using Forest Panelized Attributes (CFS-FPA). Due to the high dimensionality, redundancies, and elevated False Positive and False Negative rates in intrusion detection algorithms, a robust classification approach was employed, involving AdaBoosting and bagging to modify four classifiers: Support Vector Machine, Random Forest, Naïve Bayes, and K-Nearest

Neighbor. Initially, AdaBoost was applied, followed by bagging, and a voting aggregation technique was utilized on the CICIDS 2017 dataset. The data was pre-processed-processed by eliminating duplicate values, normalizing the features, and applying the Encoding technique. The experimental results yielded significant values, with 99.7% Accuracy and False Negative and False Positive rates of 0.053 and 0.004, respectively.

The work presented by Chen et al. [152] aims to simplify the complex process of Machine Learning directed toward intrusion detection. The authors perform a consistent feature extraction on the NSL-KDD dataset, proposing using fused convolutional neural networks for feature selection and data classification using Ensemble Stacked. The performance verification of the proposed model was undertaken with the Radar Chart Method, employed for performance analysis, enabling better visualization and comparison of multiple variables. As a result, the authors claim that their combination could improve feature extraction and intrusion detection on the NSL-KDD dataset compared to other models. When comparing the performance of the tested models, the accuracy metrics surpassed the DA, KNN, DT, NB, and LR classifiers, in the five classes: Normal, Probe, DoS, U2R and R2L of the NSL-KDD dataset with 0.9700, 0, 9850, 0.9905, 0.9960 and 0.9975 respectively. With F1-Score also presented the best results using the proposed model, presenting the results 0.9587, 0.9774, 0.9542, 0.9960 and 0.9975, respectively and with Comprehensive Performance Evaluation Value, it surpassed all, reaching 0.9296, 09297, 0.9296,, 0.9297 and 0.9299 respectively. Finally, the Ensemble Stacked proposal and the Radar Chart Method technique using fusion convolutional neural networks (FCNN-SE) proved effective in Machine Learning aimed at intrusion detection.

Experiments documented in Shen et al. [200] propose an Ensemble Voting classification approach, employing Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Decision Tree (DT) algorithms. As an improvement, they apply the Chaos Bat Algorithm (CBA) to configure weights in the voting ensemble, followed by feature selection using the SelectKBest function. Three datasets, namely NSL-KDD, UNSW-NB15, and CICIDS-2017, are utilized for the training and testing experiments. The model's performance is evaluated using metrics such as False Positive Rate, False Negative Rate, Accuracy, Precision, and F1-Score. In conclusion, the proposed model achieves better predictive performance than individual methods, and the configured weights in the CBA are highlighted as essential contributors to the results. Comparison of the performance of Class-Level Soft-Voting Ensemble (CLSVE) with feature selection based on different optimization algorithms in the NSL-KDD, CICIDS-2017 and UNSW-NB15 sets showed a detection rate of 97.90%, 99.74%, 99.23% respectively and an F1-Score of 97.98%, 99.50%, 96.82% respectively and an accuracy of 97.21%, 99.18%, 94.81% respectively. The set voting methods obtained the best predictive performance and were more consistent than the base classifiers.

In Liao et al. [201], the proposal is to enhance the challenge of introducing learning for transfer protection and extracting relevant features to prevent False Data Injection (FDI) attacks. The authors present a study on transferability and adversarial domain training (TADA). Initially, various divergence and data distribution metrics are utilized to predict the accuracy loss of the model trained with Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks in feature extraction. Subsequently, a regression model is applied to predict the decrease in accuracy in unlabeled target domains, considering the temporal space targeted by attacks in power systems. The Ensemble approach is employed to combine diverse data distribution metrics. The results demonstrate the effectiveness of the TADA model, exhibiting high performance with an average F1-Score of 92.02%, reducing distribution divergence and improving detection.

The proposal from Wang et al. [202] is based on constructing an intrusion detection system for identifying unsupervised attacks, reducing classification time using ensemble, Principal Component Analysis (PCA), and AutoEncoder (AE). According to the authors, they employ an incremental Packet-to-Vector (iP2V) technique based on a three-layer neural network, which enables faster classification time. To test their model, they use a self-developed open-source dataset called RWDIDS, generated by a network with nine IoT and Bluetooth devices infected with various types of attacks: DoS Slowhttptest, DoS Hulk, SYN DoS, SSDP Flood, Brute Force, Infiltration, Port Scan, Mirai, Fuzzing, and ARP MitM. For validation, they compare it with two other datasets, CICIDS-2017 and Kitsune, and apply Precision, Recall, and F1-Score metrics to verify the results and performance of RWDIDS along with the proposed ENIDrift model. In conclusion, the constructed model is fast and adaptable for network intrusion detection, achieving 100% F1-Score.

IV. BACKGROUND

This chapter details the classifiers, datasets, and ensemble algorithms most used in related works, and presents an overview of intrusion detection systems. An overview about Intrusion Detection Systems is also available.

A. IDS – INTRUSION DETECTION SYSTEMS

According to [203], related actions focused on maintaining the integrity, accessibility, or reliability of an electronic data source or a communication network define the objectives of an IDS. The authors provide a historical reading of the emergence and evolution of IDS, attributing to James Anderson - when publishing the article "*Computer Security Threat Monitoring and Surveillance*", by [204], the first report of an IDS. Two landmark events in the evolution of IDS occurred in 1986 and later in 1993 when publications scientifically documented intrusion detection models based on statistical network traffic analyses (both created by Dorothy Denning and Peter Neumann in [205]).

[206] defines an IDS as software developed to detect attacks that have the potential to cause damage to the com-

munication network or systems, whether they come from an insecure medium such as the Internet or the local network. Such software performs security countermeasures when detecting any anomaly in network traffic or host behavior that could characterize an attack.

The authors highlight that recently, many approaches have been applied to increase the detection rate, in general, involving *Machine Learning* techniques (situations where the detector learns through training to detect anomalies), *Rule Based* (in which case the detector has signatures characteristic of attacks and only compares real traffic with the signatures of already known attacks) and/or classical statistical methods (employment of calculations of mean, standard deviation, median, among others).

IDS have two classification dimensions: by Network Topology and by Packet Approach. The first dimension, according to [207], defines whether the detector works by analyzing the network flow (topologically, in this case, the IDS is normally placed in the communication network in the form of *bridge*[1] or receiving a copy of all packets transmitted through mirroring[2] a port on the *switch*[3]) or analyzing the behavior of the operating system, a situation where the detector works by observing processing measurements, disk space, memory usage or by auditing records made in the system logs. The second classification dimension concerns how the detector analyzes the data. [208] state that in this categorization, there are two classes: signature detection and anomaly detection. The authors explain that in signature detection, the IDS has a database of known attacks, and basically, its job is to compare the packets that reach the network with its database to verify similarities. In anomaly detection, however, the system defines a model that would be the ''normal'' behavior of the network or system, causing packets that are not classified as ''normal'' to be labeled as malicious.

B. DATASETS

Many efforts have been put into creating intrusion detection models based on Data Mining. Some datasets have been used for this. This section briefly describes the most common datasets from the research documented in our work. Table 3 compiles a relationship for most used datasets segregated by moust used datasets per year.

1) KDD-Cup'99 and NSL-KDD

The KDD-Cup'99 dataset is a subset of the DARPA dataset (which contains data from simulated operations from the US Air Force's local area network) widely used for training intrusion detectors in computer networks and made available at the Massachusetts Institute of Technology (MIT) Lincoln Laboratory[4]. It was initially introduced as part of a challenge at the KDD-Cup competition held in 1999.

A problem with the KDD-Cup'99 dataset is that it has a lot of redundant and duplicated data, respectively 78% and 75%. To provide a more robust set, Tavallaee et al. [209] presented the NSL-KDD dataset in 2009, as a derivation of KDD-Cup'99t. The most obvious benefits are eliminating redundant records in the training sample and deleting duplicate records in the testing sample.

Table 4 describes the package relationship for each class available in the aforementioned datasets.

TABLE 3. Publications segregated by moust used datasets per year.

Year	KDD'99	NSL-KDD	ISCX	UNSW	CICIDS
2015	3	4	1	0	0
2016	5	2	2	0	0
2017	5	10	1	5	0
2018	7	9	0	2	1
2019	6	11	1	6	1
2020	6	16	2	8	4
2021	4	12	0	8	7
2022	2	14	0	12	19

TABLE 4. Packages by class in the KDD-Cup'99 and NSL-KDD datasets

Dataset	Total	Benign	DoS	Probe	U2R	R2L
KDD'99	76896	70217	5728	520	50	381
NSL-KDD	69483	53561	12517	2976	50	379

The KDD-Cup'99 dataset contains 41 features that organize 4,898,431 simulated connections. The NSL-KDD dataset contains 21 different types of attacks in the training sample while having 37 attacks in the testing sample.

2) CICIDS 2017

The set of CICIDS-2017 datasets [210], made available by the *Canadian Institute for Cybersecurity*[5] through the *University of New Brunswick*[6] is divided into eight files (Table 5), according to Panigrahi et al. [211] and Stiawan et al. [212]. Each dataset contains data for a specific attack type accompanied by benign traffic data (only one subset contains only benign traffic). All have 78 *features* with their labels in column 79.

TABLE 5. Subsets of CICIDS-2017 Dataset.

Dataset	Attack type	Benign packets	Attack packets
1	Bruteforce	432074	13835
2	Infiltration	288566	36
3	DDoS	97718	128027
4	Portscan	127537	158930
5	Botnet	189067	1966
6	Web	168186	2180
7	DoS	440031	252672
8	Benign packets	529918	0

[1] situation topological where the network device is the only means through which traffic occurs.

[2] technique used to send copies of *frames* from other ports to a specific port where the IDS will be connected.

[3] network interconnection device acting in the Link layer of the OSI reference model.

[4] https://www.ll.mit.edu/

[5] https://www.unb.ca/cic/

[6] https://www.unb.ca/

3) ISCX 2012

The ISCX 2012 dataset [213], provided by the Information Security Center of Excellence at the University of New Brunswick[7] contains packages referring to DDoS attacks, HTTP DoS, Infiltration, and SSH brute force. The data is organized into seven subsets whose title is related to the day of the week when the traffic was captured (Table 6). There are 20 features where about 2% of the traffic corresponds to malicious packets.

TABLE 6. Subsets of ISCX 2012 Dataset.

Day of traffic	Attack type	Amount of data
Friday	Benign	16.1 GB
Saturday	Benign	4.22 GB
Sunday	Infiltration + benign	3.95
Monday	HTTP DoS + benign	6.85 GB
Tuesday	DDoS	23.4 GB
Wednesday	Benign	17.6 GB
Thursday	SSH brute force + benign	12.3 GB

4) UNSW-NB15

The UNSW-NB15 Moustafa et al. [214] dataset was developed at the School of Engineering and Information Technology at the University of New South Wales at the Australian Defense Force Academy[8]. 100 GB of benign and malicious traffic was captured to deliver ten classes of data: Benign, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms.

The dataset was developed based on a critique of the scientific popularity of the KDD-Cup 1999 and NSL-KDD datasets. The authors Moustafa et al. [214] highlight that the datasets mentioned above contemplate old attacks in a way that the IDS generated from this data would not be robust enough to detect modern attacks.

Table 7 compiles a relationship for most used datasets segregated by year.

C. ENSEMBLE METHODS

According to Zhou et al. [215], Ensemble Machine Learning methods consist of the joint application of different or the same classification or regression algorithms to solve the same problem. While traditional techniques build learning based on a specific technique, ensemble techniques seek to aggregate, either in a parallel or sequential way, multiple learning techniques to achieve improvements in terms of variance, bias, or predictions Smolyakov et al. [216]. The construction of ensemble methods can be performed with classifiers of different types, called "heterogenous ensembles" or multiple classifiers that use the same learning, which is called "homogeneous ensembles" Zhou et al. [215].

Different machine learning algorithms are applied to generate individual models with a hypothetical training sample. Generically, the learning or the way of classifying the methods is grouped in the item "Combination" depending on the ensemble technique used, which illustrates the possibility of sharing the power of classification between the methods of the previous time. As a consequence, we have a "Classifier" at the end (to the right) of the flow that is nothing more than a classification process that will add the intelligence of the four methods present in the ensemble.

Still, according to the author, there are three ways to organize an ensemble method concerning how the individual algorithms relate: "combining classifiers", "ensembles of weak" and "mixture of experts". The first form is the most used in the literature, especially for solving pattern recognition problems. It is based on the combination of classifiers with excellent accuracy and seeks to combine them to improve the final (joint) accuracy in the classification. The second form is the most used by the machine learning community and consists of the joint application of light classification algorithms (with reasonable accuracy), making the final classifier have excellent accuracy, extracting the best from each light classifier. Finally, the last method is more explored in neural network-based approaches and assumes that a better solution can be obtained by combining rules and mixing parameters.

Table 8 compiles a relationship for most used Ensemble methods segregated by year.

1) Bagging

The authors of Khonde et al. [217] emphasize that this is one of the first proposed ensemble algorithms. It uses the bootstrap[9] to obtain subsamples of the training dataset, which means sampling a large amount of data collected randomly and without repetition. Then each classifier is trained with a specific sample obtained using bootstrap. The algorithm's ranking process consists of a majority of votes. Given a test sample, each algorithm defines the class according to its previous learning. The ensemble will define the common classification method by choosing the most chosen class among the ensemble's algorithms. The above method can be understood as follows, considering in dataset D, the training sample D_a, and the test sample D_b:

Train

1) Choose the number of samples n and the base classifier C;
2) Create n training samples with bootstrap D_{a1}, D_{a2}, ..., D_{an};
3) Fit base classifier C for each sample D_{ai} to create n classifiers C_1, C_2, ..., C_n.

Test

1) For each sample x in D_b, test x per all classifiers C_1, C_2, ..., C_n;
2) Choose the final class based in voting - consider the label most chosen by the classifiers C_1, C_2, ..., C_n.

[7]https://www.unb.ca/cic/about/hub.html
[8]https://www.unsw.adfa.edu.au/seit

[9]Data selection method that obtains a sample for analysis of the given dataset and, when necessary, search for new samples obtained from the set whole, considering the data that were selected in the previous sampling(s) - also known as sampling with replacement.

TABLE 7. Relation of datasets used per article per year

Dataset	2015	2016	2017	2018	2019	2020	2021	2022
CIC-IDS-2017				[60]	[78]	[95] [102] [117] [120]	[136] [140] [150]	[152] [155] [156] [157] [159] [162] [163] [166] [157] [167] [169] [171] [176] [177] [181] [182] [184] [181] [199] [152] [200] [202]
ISCX-IDS-2012	[16]	[27] [34]	[48]			[87] [110]		
KDD-Cup'99	[14] [17] [18] [19] [24]	[29] [30] [32] [33] [34] [35]	[37] [38] [41] [42] [46] [47] [48] [52]	[54] [55] [57] [59] [60] [62] [68] [69] [70]	[72] [73] [76] [77] [78] [82] [83] [84]	[88] [94] [95] [97] [101] [103] [104] [107] [108] [112] [120]	[124] [125] [127] [130] [137] [140] [145] [150]	[154] [155] [157] [162] [157] [178] [183] [194]
NSL-KDD	[15] [19] [21] [24]	[33] [35]	[36] [37] [39] [40] [41] [42] [46] [50] [51]	[53] [55] [56] [59] [60] [62] [63] [64] [67] [68] [69] [70]	[72] [73] [74] [76] [77] [78] [80] [81] [85]	[87] [88] [94] [95] [97] [99] [100] [103] [104] [105] [111] [112] [113] [116] [118] [120]	[127] [130] [132] [133] [135] [137] [139] [140] [142] [143] [144] [150]	[152] [153] [154] [163] [164] [174] [178] [181] [185] [153] [191] [192] [193] [194] [181] [152] [200]
UNSW-NB15			[37] [41] [42] [45]	[60] [65]	[71] [78] [82] [85]	[89] [92] [101] [103] [115] [119] [120] [121]	[122] [123] [126] [127] [132] [140] [141] [142] [147] [150]	[157] [163] [165] [157] [168] [178] [181] [191] [192] [194] [198] [181] [200]
Other	[22] [23]	[25] [26] [28] [31]	[49]	[58] [66]	[75]	[90] [91] [96] [106] [109] [114]	[129] [131] [134] [146] [149] [151]	[153] [158] [160] [161] [172] [173] [175] [179] [180] [186] [187] [188] [189] [190] [153] [195] [196] [201]

TABLE 8. Publications segregated by most used ensemble methods per year.

Year	Bagging	Boosting	Stacking	Voting
2015	5	5	1	1
2016	2	1	0	5
2017	4	6	1	7
2018	6	5	1	7
2019	2	4	2	7
2020	7	5	9	3
2021	4	8	7	6
2022	9	8	7	6

3) Repeat the process for each sample x in D_b.

The use of decision trees generates a method called a random forest, a classification algorithm widely used in the literature and a classic example of the application of the bagging method. According to Alsouda et al. [218], the random forest algorithm can be considered an example of bagging with the characteristic that the decision trees are similar to each other, which leads to decisions with a high correlation index.

2) Boosting

In 1990, Schapire showed that clustering several weak classifiers could generate a general strong classification method, except when the dataset is very small Schapire et al. [219]. Years later, Freund et al. [220] presented AdaBoost, a technique capable of implementing the initial concept of Schapire et al. [219] effectively. The algorithm consists of performing an initial classification with some learning method and, in the following classifications, modifying the weights to make the labelling errors more evident, thus forcing a new classification that is more biased towards success. The AdaBoost algorithm can be observed as follows, considering a dataset

D with N instances:

Train

1) Choose the start classifier C;
2) Choose starting weights w_{1i} whereas $w \in [0,1]$ is the sum of all weights $\sum_{i=1}^{N} w_{1i} = 1$. Usually $w_{1i} = \frac{1}{N}$;
3) For $k = 1 \rightarrow N$, create a training sample D_k with samples of D;
4) Fit machine learning algorithm C in D_k to crate classifier C_k;
5) For each misclassification of C_k in D, calculate the general error e_k being the sum of the weights of all errors $\sum_{j=1}^{N} w_{ki}$;
6) If $e_k \in (0, 0.5)$ calculate new weight $\beta_k = \frac{e_k}{1-e_k}$ and update $w_{k+1,i} = w_{ki} \cdot \beta_k$ causing the weight of correctly classified elements to decrease;
7) Normalize $w_{k+1,i}$;
8) For elements classified incorrectly e_k assign $w_{ki} = \frac{1}{N}$;
9) Repeat the process for all $C_1, C_2, ..., C_n$ classifiers in order to update the weights based on $\beta_1, \beta_2, ..., \beta_n$.

Test

1) For each object x on test dataset, classify x based on $C_1, C_2, ..., C_n$;
2) For each y class attributed to x by C_k calculate $\mu_y(x) = \sum C_k(x) = y\ ln\left(\frac{1}{\beta_k}\right)$, so that the smaller the error β_k greater will be the value of $\mu_y(x)$;
3) The class with the maximum $\mu_y(x)$ is chosen as the label of x;
4) Repeat the process for each x element in the test dataset.

It is noticed that the AdaBoost penalizes the errors to keep their weight high while the hits have their weights reduced each round. While the bagging method selects the final label by the majority of votes, the boosting specified here by the AdaBoost algorithm uses the weighted majority of votes.

There are other boosting algorithms extensively explored in the literature, according to [221]: Gradient Boosting and XGBoosting. While AdaBoost works by adjusting the weights for each new classification round, Gradient Boosting and XGBoosting try to adjust the classification of new rounds based on the residual errors of the previous classification using gradient descent for weight adjustment. The difference between Gradient Boosting and XGBoosting is that the former, by the method, is slow. The second is an evolution that, despite using the same methodology, has the capacity for parallelization (using multiple CPUs during training), support for distribution between processor nodes (clusters in a distributed systems environment), and optimization of the cache of way to improve the use of the hardware that will run it.

3) Stacking

The ensemble model called stacking was initially proposed by Wolpert et al. [222]. The technique consists of implementing two classification layers, the first composed of n classifiers and the last composed of a single final classifier. Unlike the bagging and boosting techniques, the classifiers do not necessarily need to be the same; the first layer can be composed of heterogeneous classifiers.

Agarwal et al. [223] define the stacking process as a combination of predictions made by multiple learning methods $(L_1, L_2, ..., L_n)$ generated by multiple classifiers $(l_1, l_2, ..., l_n)$ in an initial layer. Such classifiers are trained with the same training sample D_{Train} which contains samples in the format $s_i =< x_1, y_i >$ where x_i is the feature vector containing values for all *features* of D_{Train} and y_i is the label of the class to which the vector belongs.

In the first phase the classifiers $l_1, l_2, ..., l_n$ perform predictions for a vector x_q. In the second phase, a M meta-classifier makes the final prediction of the class, taking into account the prediction made in the previous layer. Agarwal et al. [223] highlights the importance of choosing a meta-classifier as fundamental for increasing classification performance and D et al. Dvzeroski et al. [224] compliments stating that the use of a meta-classifier is justified only when the ensemble's performance is superior to the performance of the singular classifier so that it is necessary to observe in the implementation process which individual classifier has the best performance for comparison with the performance of the ensemble.

Works Aggarwal et al. [225] and Rocca et al. [226] define the stacking algorithm as the following process:

1) Split the training dataset D_{Train} in two folds $D_{Train}A$ and $D_{Train}B$;
2) Choose the individual classifiers $l_1, l_2, ..., l_n$ on first layer and fit us using $D_{Train}A$;
3) For each classifier $l_1, l_2, ..., l_n$ make predictions on fold $D_{Train}B$; and
4) Fir the meta-classifier on fold $D_{Train}B$ using as input parameters a new dataset which is $D_{Train}B$ concatenated with the predictions of the previous step made by $l_1, l_2, ..., l_n$.

Figure 6 illustrates stacking algorithmic process.

Consider letters A-I (Figure 6) as a way to analyse the process:

A – Illustrates the training sample, which is part of a hypothetical dataset;

B – After the partitioning stage, a sample of A is found in B;

C – Likewise, a sample of A is found in C;

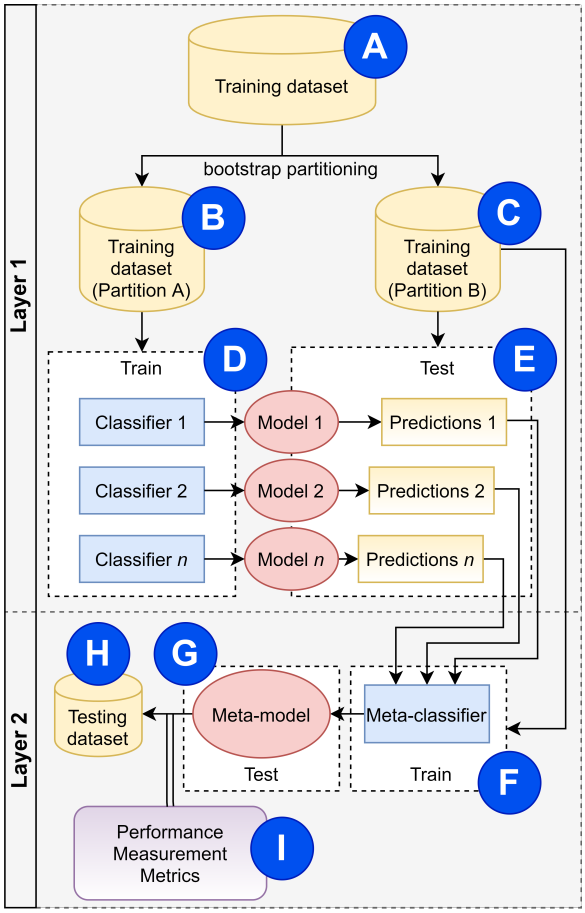
D – Represents the training process of individual classifiers $\{1, 2, ..., n\}$ aiming to create predictor models $\{1, 2, ..., n\}$;

E – Test stage, in which the predictor models $\{1, 2, ..., n\}$ make predictions using as test sample the second partition obtained in C (C1). The predictions are linked to partition B as new features (C2);

F – Represents the training process of a meta-classifier that is made by using the partition obtained in C (C3), linked to the predictions from C2;

G – Test process of meta-model, in which the predictions are made in the sample tests from a hypothetical dataset;

FIGURE 6. Workflow of the stacking algorithm.



H – Represents the test sample from a hypothetical dataset; and

I – Process of analysing the stacking performance, from which factors such as accuracy measurements, precision, recall can be obtained.

Finally, quoting Wolpert et al. [222], the stacking method can be defined as a way of forwarding information from a group of classifiers to another before reaching the final classification. This process can be useful as it may reduce the error rate in a prediction model.

4) Voting

A simple way to implement classifiers together is through voting. Suppose a binary classification scenario with five classifiers where two choose class 0 while three choose class 1. By applying the majority vote, class 1 will be assigned as the label predicted by the ensemble because most classifiers chose that class.

A common type of voting is the weighted vote, where specific classifiers (usually those that produce a better performance) have their vote with greater weight, making their predictions have a multiplier factor allowing a more significant influence on the classification decision.

Table 9 compiles a relationship for ensemble methods used per article per year.

D. BASE CLASSIFIERS

Machine Learning is a line of artificial intelligence belonging to computer science that seeks, through mathematical and statistical methods and in conjunction with related areas, to analyze a dataset and learn patterns through classification, clustering, and regression. Many algorithms can be employed to solve such tasks. Classification, regression, and clustering are problems that, according to Lee et al. [227], the methods of machine learning, in general, solve very well. The authors of Bakshi et al. [228] define that "classification" consists of training an algorithm of learning in a sample of tests (a portion of the dataset with data in a variable) and testing it on the remaining sample of the data to verify the performance of the algorithm.

In classification problems, machine learning algorithms learn patterns in the data present in the training sample and must be able to identify the classes to which the test sample data belongs. Regression is about finding patterns in data distribution in a dataset and predict continuous values. Clustering is a way of grouping similar data according to specific characteristics Lee et al. [227]. Table 10 organizes the list of classifiers that have been most used in recent years by the criterion used in the systematic literature review. Next, the highlighted classifiers are briefly described.

1) Support vectors machine

An SVM-based classifier seeks, given in the parameter space two linearly separable classes (a and b), to draw the best hyperplane so that the element of class a closest to the first element of class b is considered the limits. The limits are called "support vectors" because they represent the closest elements a and b. The hyperplane, therefore, will be constructed at exactly half of these limits.

The SVM strategy is to find the most significant margin of separation between classes, extracting a classification function for a sample of observations Mueller et al. [229]. The idea of keeping as much margin as possible for tracing the hyperplane is that the test samples will be distributed differently in the parameter space, so the more significant the margin, the lower the error rate.

Figure 7 illustrates a situation of SVM application in intrusion detection in computer networks.

For the analysis of Figure 7, it must be taken into account that the author created a hypothetical situation of two-dimensional distribution to illustrate the purpose of the application of SVM and that, not necessarily, only the parameters "Source IP first octet" (y axis) and "Destination port" (x axis) help classify network traffic effectively.

Furthermore, the leftmost cluster represents "class A" and all its objects where the element closest to "class B" (located more to the right) is chosen as the support vector. The same applies, in reverse logic, to another class. Therefore, when finding the support vectors of the analyzed classes, the SVM

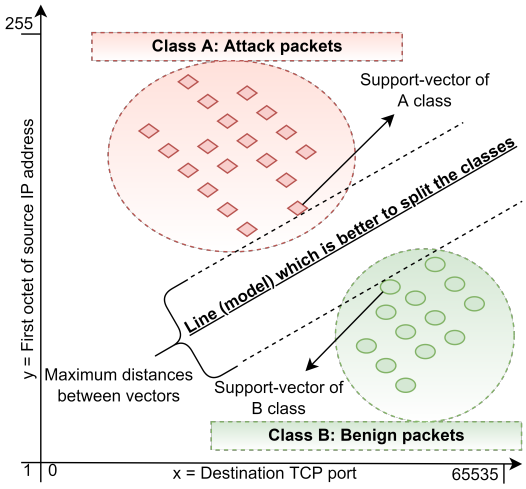
TABLE 9. Relation of ensemble methods used per article per year.

Dataset	2015	2016	2017	2018	2019	2020	2021	2022
Bagging	[17] [19] [22] [23]	[25] [26] [29] [33]	[36] [45] [46] [49]	[53] [55] [57] [58] [59] [63] [65]	[80] [81] [85]	[94] [96] [100] [101] [104] [114] [116] [121]	[127] [129] [131] [134] [135] [136] [141] [143] [149]	[155] [157] [158] [157] [168] [171] [179] [180] [197] [199]
Boosting	[14] [17] [18] [22] [23]	[27] [33]	[36] [37] [40] [49] [50]	[55] [57] [58] [63] [64] [65] [67] [70]	[71] [76] [80] [81] [84]	[92] [93] [96] [99] [102] [104] [111] [114] [116] [118] [120] [121]	[122] [123] [124] [126] [127] [129] [131] [132] [134] [135] [136] [139] [145] [147] [149]	[152] [155] [156] [157] [165] [166] [157] [167] [169] [171] [173] [178] [180] [181] [190] [191] [197] [198] [181] [199] [152]
Stacking	[16]		[45]	[55]	[80] [85]	[88] [89] [90] [105] [108] [109] [111] [119] [120]	[126] [129] [131] [134] [136] [139] [144]	[152] [169] [171] [172] [174] [176] [177] [185] [152]
Voting	[15]	[26] [30] [32] [33]	[39] [40] [41] [42] [46]	[56] [59] [60] [62] [67] [70]	[74] [77] [78] [81] [85]	[87] [91] [106] [107] [116]	[132] [133] [137] [141] [147] [149]	[163] [165] [182] [188] [190] [193] [194] [199] [200]
Other	[21] [24]	[28] [31] [34] [35]	[38] [47] [48] [51] [52]	[54] [66] [68] [69]	[72] [73] [75] [82] [83]	[95] [97] [98] [103] [110] [112] [113] [115] [117]	[125] [128] [130] [138] [140] [142] [146] [150] [151]	[153] [157] [159] [160] [161] [162] [164] [157] [175] [181] [183] [184] [186] [187] [189] [153] [192] [195] [196] [181] [201] [202]

TABLE 10. Publications segregated by Base Classifier and Year.

Year	SVM	k-NN	MLP	Decision tree
2015	2	1	2	8
2016	1	1	1	6
2017	4	2	2	8
2018	4	0	6	7
2019	6	5	6	14
2020	7	6	6	25
2021	8	5	4	18
2022	14	9	2	7

FIGURE 7. SVM hypothetical example



establishes the margins representing the most significant possible distance between the classes. Finally, the hyperplane that best divides the classes is drawn, a midline between the margins.

Moreover, the ability of SVM to deal with situations of class separation in n-dimensional spaces is important. In the example of Figure 7 (above to understand the basic functioning of the method), a simple straight line deals with dividing the classes. Baeza-Yates et al. [230] emphasize that in the existence of more dimensions, the method can solve binary classification problems with equal efficiency. If in a two-dimensional situation, the division of classes is carried out by drawing a straight line, in a three-dimensional situation, for example, a plane would be drawn to complete the classification. In short, SVM can handle several dimensions, highlighting the computational cost required as more dimensions

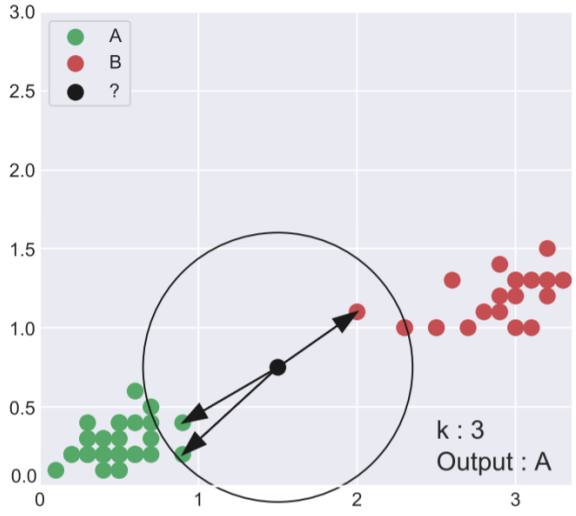
(and features) are analyzed.

2) K-Nearest Neighbors

K-Nearest Neighbors (kNN) is a simple machine learning algorithm used for classification and regression. Given a set of labeled data points, the algorithm can predict the label or value of a new data point by finding the k data points in the training set that are closest to it (i.e., its "neighbors") and returning the majority label or average value of those neighbors. The value of k is a hyperparameter that can be tuned for a particular problem. The algorithm is easy to implement and does not require any training phase, but it can be computationally expensive for large datasets and may not generalize well to unseen data. Because kNN algorithm needs to compute the distance to all training samples for all of the testing samples, its computational cost can be explained quite simply Deng et al. [231]. By establishing a value for k, the algorithm, in the parameter space, calculates the distance from the test sample to all training samples and orders them by the distance. Finally, for the closest k neighbors, the most frequent class is observed, which will finally be assigned to the analyzed object.

According to Tchaye-Kondi et al. [232], although it is simple, k-NN can obtain excellent results in data classification. The best value of the k hyperparameter is usually obtained empirically. Figure 8 illustrates a hypothetical parameter space for a better understanding of how k-NN works.

FIGURE 8. Hypothetical parameter space for k-NN Tchaye-Kondi et al. [232]



Analyzing Figure 8, it is possible to observe that, when the value of $k == 3$, the two nearest neighbors (most of 3) are in the cluster on the left. In this way, the analyzed central object should be assigned with the class on the left as a label.

kNN with the Euclidean distance metric is one of the classifiers used to create the *Stackings* presented in this work. For example, observing the elements A, B, the values referring to the axes x and y are obtained (in a two-dimensional analysis), and A has the values x_A and y_A and B has the values x_B and y_B. The distance D will therefore be calculated as $D = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$. When considering more dimensions n in the parameter space, the equation must be adjusted by adding the sum of the values $(x_n + y_n)^2$ to the square root. The tested k values were stipulated in a loop ranging from 3 to 9, where the best k in the range $\{3, 5, 7, 9\}$ was chosen for each test.

3) Multi Layer Perceptron

An artificial neural network is a structure imitating in a simple way the human brain in the sense of communication between neurons. The human brain responds to stimuli at the inputs of its neurons, and the cellular organization causes others to be activated (or not) in response to the processing of that stimulus. The structure of the other layers and the propagation of stimuli depend on the type of neural network created. The last layer of neurons corresponds to the classes of the analyzed data so that it is possible, given the input parameters and the propagation of the stimuli, to trigger the neuron corresponding to the estimated class Tinós et al. [233]. Likewise, an artificial neural network typically has the number of inputs corresponding to the number of features of a dataset to be processed.

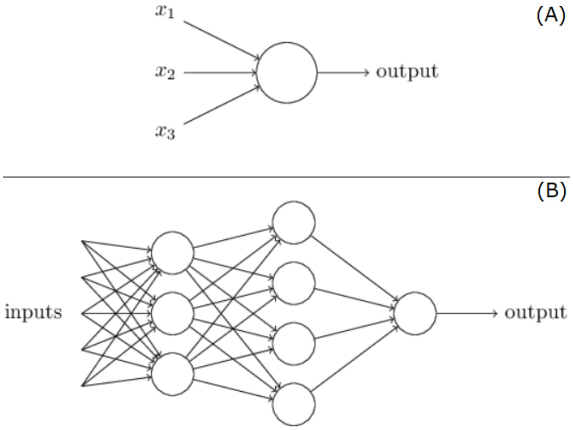
Multilayer perceptron (MLP) is a type of artificial neural network used for supervised learning tasks such as classification and regression. It consists of an input layer, one or more hidden layers, and an output layer. Each layer is composed of nodes (also known as neurons) that apply a nonlinear activation function to the weighted sum of their inputs, and the outputs of one layer become the inputs to the next. The weights of the connections between the neurons are learned from the training data using a optimization algorithm such as stochastic gradient descent. During the training phase, the model iteratively adjusts the weights to minimize a loss function that measures the difference between the predicted and actual outputs. The resulting model can be used to make predictions for new, unseen data. MLP has been widely used in many applications and has been shown to be effective in solving complex problems.

The original Perceptron model was developed by Rosenblatt et al. [234], where the author presented an architecture that can be analyzed, according to Goodfellow et al. [235], based on Figure 9 (A).

For each entry x_1, x_2, x_3, a weight w_1, w_2, w_3 is introduced, valued according to the importance of the feature represented by each entry in the hypothetical dataset analyzed. From this point, a threshold t is defined that will be the basis for the neuron's output, being 0 if the weighted sum of the weights for each input is less than the value stipulated in the *threshold* (0 *if* $\sum_j w_j x_j \leq t$) or 1 if the result is less (1 *if* $\sum_j w_j x_j > t$).

In Figure 9 (B) it is possible to observe a model of *neural network* (or Perceptron with an intermediate layer - multilayer perceptron). Such a model can classify problems of more complex order since this layer will deal with the outputs of the neurons of the first processing layer, making the model

FIGURE 9. Perceptron models Goodfellow et al. [235]. Simple architecture (A), and multilayer architecture (B).



able to separate data in a non-linear way. Each decision made by neurons in the second layer will weigh the decisions made in the previous layer.

Furthermore, it is a ''feed-forward'' network; that is, data propagation always takes place in the direction of the inputs to the output. The training process for these types of networks consists of learning the best weights (variables that specify the importance of each feature) and bias (a variable that will balance the threshold). Tinós et al. [233] highlights that, although it is relatively simple, the multilayer perceptron has been successfully applied as a type of *neural network* in the solution of several complex problems.

4) Decision tree

Decision trees are a popular machine learning algorithm used for both classification and regression tasks. The algorithm builds a tree-like model of decisions and their possible consequences. Each node in the tree represents a test on an input feature, each branch represents the outcome of the test, and each leaf node represents a prediction. The tree is constructed by recursively splitting the data into subsets based on the feature that provides the most information gain until a stopping criterion is reached. The final prediction is made by traversing the tree from the root to a leaf node, following the branches corresponding to the feature values of a new sample. Decision trees are easy to interpret, fast to train, and can handle both categorical and numerical data. However, they are prone to overfitting and can result in complex trees that are difficult to interpret. To overcome these limitations, various decision tree ensembles such as random forests and Gradient Boosted Trees have been developed. According to Rezende et al. [236], the methods that use decision trees belong to the family ''Top Down Induction of decision trees'' (TDIDT) which is a data structure that defines ''nodes'' and ''decision nodes''. The first node is responsible for representing a class, while the second node is responsible for representing a test on some attribute. For each example path to be a leaf, the attributes are conditioned by the ''decision nodes'' to make one up to a ''node'' class. This sequence of logical ''*ifs*'' will determine the path to be followed in the tree for each example.

Work Sharma et al. [237] defines DT as a flowchart-like structure where each internal node represents a test for an attribute, each branch represents the result of a test, and each leaf represents a class. The authors emphasize that less time can be constructed, and fewer processing methods stand out as less DT.

Table 11 compiles a relationship for most used classifiers segregated by year.

V. DISCUSSION AND OPEN ISSUES

In this section, we will describe the summary of the compilation of the analysis of the results obtained after the review.

Machine learning algorithms present a robust alternative for building Intrusion Detection Systems due to their ability to recognize attacks in computer network traffic, identifying patterns in large amounts of data. Typically, classifiers are trained for this task. Together, ensemble learning algorithms have increased the performance of these detectors, reducing misclassifications and allowing computer networks to be more protected. This research presents a comprehensive Systematic Literature Review in which works related to intrusion detection with joint learning was obtained from the most relevant scientific bases. We analyzed 256 works, but after applying filters, we reached 188 works related to our theme. We compiled several datasets, classifiers, and ensemble algorithms and documented experiments that excelled in performance. A characteristic of this research is its originality. We found no surveys specifically focused on the relationship between ensemble techniques and intrusion detection. For the last 8 years covered by this research, we present a timeline-based view of the works studied to highlight evolutions and trends. Our analysis shows a growing area, with excellent results in attack detection but with needs for improvement in the selection of classifiers, making this work unprecedented for this context.

It is possible to perceive a significant growth in the scientific community's interest in applying ensemble techniques within the context of cybersecurity. This is evident from the year-on-year increase in the number of publications.

The CICIDS-2017 dataset has increased use, likely because it contains more recent attacks. The NSL-KDD and KDD-Cup'99 datasets have also been used frequently, although their data is old and does not reflect current attacks. One explanation could be that because they were widely used in the past, they provide an interesting comparison metric with recent algorithms. Furthermore, the UNSW-NB15 dataset has experienced growth in usage.

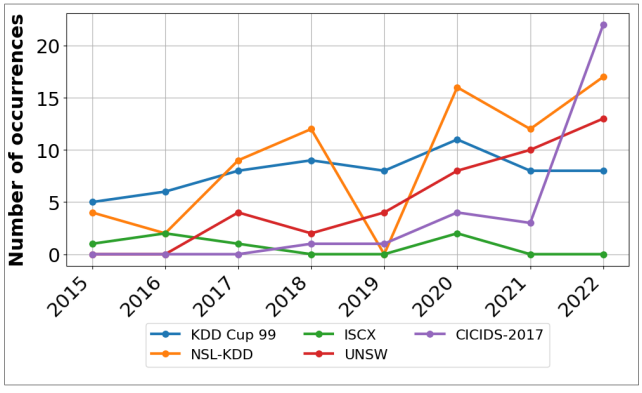
It is worth highlighting an analysis by Engelen et al. [238] where several significant issues related to traffic generation, flow construction, feature extraction, and labeling were revealed, negatively affecting usefulness of CICIDS-2017. The authors investigate the causes of these deficiencies and propose an improved data processing methodology resulting in

TABLE 11. Relation of base classifiers algorithm used per article per year.

Dataset	2015	2016	2017	2018	2019	2020	2021	2022
Decision tree	[15] [17] [18] [21] [22] [23]	[26] [28] [29] [33]	[36] [37] [41] [45] [46] [47] [49]	[56] [58] [60] [63] [65] [67] [68]	[71] [74] [75] [76] [77] [78] [80] [81] [84] [85]	[88] [89] [90] [91] [93] [94] [97] [102] [107] [109] [111] [112] [116] [117] [119] [120] [121]	[123] [126] [127] [129] [131] [132] [134] [135] [137] [139] [141] [143] [144]	[152] [154] [155] [157] [158] [159] [163] [165] [166] [157] [168] [173] [174] [179] [180] [181] [191] [194] [195] [181] [199] [152] [200]
k-NN	[18]	[27]	[38] [42]	[55]	[74] [75] [77] [78] [80] [81]	[89] [90] [95] [116] [117] [119]	[122] [127] [131] [136] [137] [150]	[153] [157] [160] [166] [157] [169] [171] [177] [179] [190] [153] [191] [199] [200]
Neural	[18] [21]	[35]	[45] [51]	[54] [56] [60] [64] [65]	[71] [72] [74] [77] [78]	[87] [91] [95] [106] [115] [116]	[128] [130] [136] [146] [151]	[152] [157] [161] [163] [157] [172] [175] [176] [177] [181] [182] [184] [185] [187] [196] [197] [181] [152] [201] [202]
SVM	[18] [21]	[30]	[38] [39] [40] [48]	[53] [55] [60] [62] [65] [70]	[73] [74] [75] [77] [78] [80] [85]	[89] [91] [105] [107] [109] [114]	[125] [127] [134] [136] [137] [140] [147] [150]	[153] [157] [160] [162] [163] [164] [166] [157] [168] [171] [174] [177] [179] [153] [192] [193] [194] [199] [200]
Other	[14] [19] [24]	[25] [31] [34]	[50]	[59] [66] [69]	[82] [83]	[92] [96] [98] [99] [100] [101] [104] [108] [110] [113] [118]	[124] [133] [138] [142] [145] [149]	[153] [156] [157] [157] [167] [178] [181] [183] [186] [188] [189] [153] [198] [181]

the reconstruction or reclassification of more than 20% of the original traffic samples.

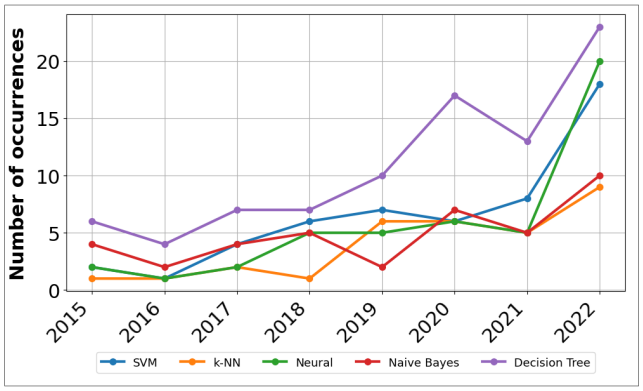
FIGURE 10. Publications by most used datasets per year of publication.



Neural networks, decision trees, SVM, k-NN, and Naive Bayes, have been used more frequently in recent publications, emphasizing neural networks, decision trees, and SVM.

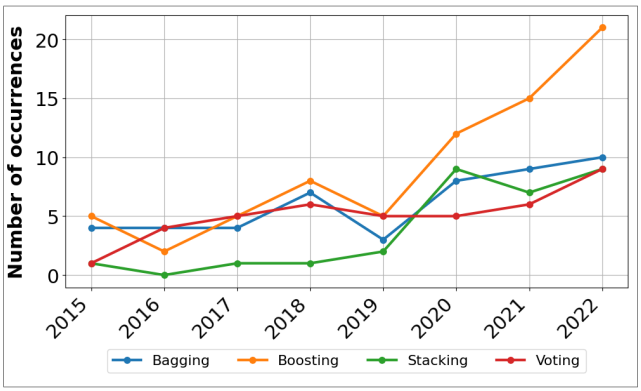
The Boosting algorithm has stood out as a trend among the most used. In addition, the Bagging, Stacking, and Voting techniques are the most used among the ensemble algorithms found.

FIGURE 11. Most used classifiers by year of publication.



There is a gap in terms of semantic interpretability when adopting a set approach within the cybersecurity context, demonstrated by Mink et al. [239], enhanced by the use of ensemble learning algorithms, which can often demonstrate high metrics in set sample identification tasks of data but may fail to understand attack patterns truly. While these algorithms effectively detect threats, their lack of interpretability hinders security professionals' ability to understand how and why these threats are being detected. Therefore, future research

FIGURE 12. Four most used ensemble methods by year of publication.



should focus on developing explanatory machine learning techniques that improve accuracy in threat detection and enable clearer and more detailed interpretation of the decision processes of ensemble learning algorithms, thus addressing this semantic gap. Crucial in the context of security operations.

Despite the growing interest in ensemble techniques for intrusion detection, we observed another gap in the scientific literature that deserves attention. Few studies have examined the impacts of oversampling, undersampling, or resampling on ensemble algorithms. These resampling techniques are widely used in class imbalance problems and, therefore, are relevant to the context of intrusion detection.

Another gap we identified is the lack of research that evaluates the impacts of applying different feature selection algorithms on ensemble algorithms. Proper feature selection can significantly improve model performance, but few studies have explored this relationship in the context of ensemble learning applied to cybersecurity.

Finally, we note the lack of studies investigating the impacts of applying pruning techniques on ensemble algorithms. Pruning is a strategy that aims to reduce the complexity of the model, eliminating less relevant branches or nodes in the decision tree. This technique can be valuable for improving the interpretability and efficiency of ensemble algorithms, but its investigation has been limited.

VI. CONCLUSION

This research systematically reviews the literature to explore the relationship between Ensemble Learning and Cybersecurity, specifically in Intrusion Detection. By analyzing 188 related works, we compiled diverse datasets, classifiers, and ensemble algorithms and documented the experiments that excelled in performance.

One of the main contributions of this work is its originality, as we did not find research in the literature that specifically focused on this relationship. The expressive growth of the scientific community's interest in applying ensemble techniques in cybersecurity clearly indicates the relevance and promise of these methods for detecting attacks on computer networks.

The identified trends, such as the growth in the use of the CICIDS-2017 dataset and the highlight of the Boosting algorithm among the most used ones, provide important insights for researchers and professionals in the field. In addition, information about the frequent use of old datasets, such as NSL-KDD and KDD-Cup'99, brings to light the importance of new efforts to build more up-to-date and realistic datasets for evaluating algorithms.

Regarding public datasets, a critical problem was documented by Catillo and Villano [240]. Despite the rapid growth in the number of articles in this area, many follow a common pattern: propose intrusion detection systems, conduct tests with public datasets, and obtain exceptional detection performance. However, this approach may be misleading. The collection, dissemination, and use of these datasets should be cautiously approached and offer concrete guidelines for building future datasets and more rigorous machine learning experiments.

Finally, we identified some gaps in the literature that deserve to be addressed in future research. The lack of studies that observe the impacts of oversampling, undersampling, or resampling on ensemble algorithms, as well as the absence of investigations on the effect of applying different feature selection algorithms and pruning techniques, highlight the need to advance in understanding and optimization of these approaches.

We hope that this research serves as a solid basis for the scientific community in cybersecurity and that the results and trends identified can guide future research and the development of more effective and reliable intrusion detection systems. The comprehensive analysis of related works and the historical view presented can contribute to the continuous advancement of this area and, consequently, to the security of computer networks and information systems.

REFERENCES

[1] G. Folino and P. Sabatino, "Ensemble based collaborative and distributed intrusion detection systems: A survey," *Journal of Network and Computer Applications*, vol. 66, pp. 1–16, 2016.
[2] A. A. Aburomman and M. B. I. Reaz, "A survey of intrusion detection systems based on ensemble and hybrid classifiers," *Computers & security*, vol. 65, pp. 135–152, 2017.
[3] D. Y. Fraimovich, O. A. Donichev, S. A. Grachev, and M. A. Gundorova, "The role of information and digital resources in regional development," in *Growth Poles of the Global Economy: Emergence, Changes and Future Perspectives*, pp. 1305–1316, Springer, 2020.
[4] K. Schwab, *A quarta revolução industrial*. Edipro, 2019.
[5] Forbes, "Cybersecurity in 2022 – a fresh look at some very alarming stats." shorturl.at/bfpqS, Jan 2022. (Accessed on 07/05/2022).
[6] Allianz, "Allianz risk barometer | agcs." shorturl.at/gptw7, Jan 2022. (Accessed on 07/05/2022).
[7] S. Morgan, "Cybercrime to cost the world $10.5 trillion annually by 2025." shorturl.at/coUZ6, Nov 2020. (Accessed on 07/05/2022).
[8] Accenture, "Cost of cybercrime study | 9th annual | accenture." shorturl.at/bemsu, Mar 2019. (Accessed on 07/05/2022).
[9] K. R. F. Scannavino, E. Y. Nakagawa, S. C. P. F. Fabbri, and F. C. Ferrari, "Revisão sistemática da literatura em engenharia de software: teoria e prática," 2017.
[10] B. Kitchenham, P. Brereton, Z. Li, D. Budgen, and A. Burn, "Repeatability of systematic literature reviews," in *15th Annual Conference on*

Evaluation & Assessment in Software Engineering (EASE 2011), pp. 46–55, IET, 2011.

[11] S. Keshav, "How to read a paper," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 3, pp. 83–84, 2007.

[12] C. S. Ishikiriyama, D. Miro, and C. F. S. Gomes, "Text mining business intelligence: a small sample of what words can say," *Procedia Computer Science*, vol. 55, pp. 261–267, 2015.

[13] T. J. Lucas, "About intrusion detection in computer networks and computational systems: a pruning proposal to reduce computational cost and gain performance using ensemble learning. ph.d. thesis, dept. of comput. sci., são paulo state university, bauru, br," 2023.

[14] L. Mehra, M. K. Gupta, and H. S. Gill, "An effectual & secure approach for the detection and efficient searching of network intrusion detection system (nids)," in *2015 International Conference on Computer, Communication and Control (IC4)*, pp. 1–5, IEEE, 2015.

[15] N. F. Haq, A. R. Onik, and F. M. Shah, "An ensemble framework of anomaly detection using hybridized feature selection approach (hfsa)," in *2015 SAI Intelligent Systems Conference (IntelliSys)*, pp. 989–995, IEEE, 2015.

[16] M. Milliken, Y. Bi, L. Galway, and G. Hawe, "Ensemble learning utilising feature pairings for intrusion detection," in *2015 World Congress on Internet Security (WorldCIS)*, pp. 24–31, IEEE, 2015.

[17] P. Amudha, S. Karthik, and S. Sivakumari, "Intrusion detection based on core vector machine and ensemble classification methods," in *2015 International Conference on Soft-Computing and Networks Security (ICSNS)*, pp. 1–5, IEEE, 2015.

[18] P. Sornsuwit and S. Jaiyen, "Intrusion detection model based on ensemble learning for u2r and r2l attacks," in *2015 7th international conference on information technology and electrical engineering (ICITEE)*, pp. 354–359, IEEE, 2015.

[19] D. Gaikwad and R. C. Thool, "Intrusion detection system using bagging ensemble method of machine learning," in *2015 International Conference on Computing Communication Control and Automation*, pp. 291–295, IEEE, 2015.

[20] M. Sreenath and J. Udhayan, "Intrusion detection system using bagging ensemble selection," in *2015 IEEE International Conference on Engineering and Technology (ICETECH)*, pp. 1–4, IEEE, 2015.

[21] Z. Ye and Y. Yu, "Network intrusion classification based on extreme learning machine," in *2015 IEEE International Conference on Information and Automation*, pp. 1642–1647, IEEE, 2015.

[22] R. R. Robinson and C. Thomas, "Ranking of machine learning algorithms based on the performance in classifying ddos attacks," in *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, pp. 185–190, IEEE, 2015.

[23] S. Gonzalez, A. Herrero, J. Sedano, U. Zurutuza, and E. Corchado, "Different approaches for the detection of ssh anomalous connections," 2015.

[24] B. A. Tama and K. H. Rhee, "A Combination of PSO-Based Feature Selection and Tree-Based Classifiers Ensemble for Intrusion Detection Systems," in *Advances in Computer Science and Ubiquitous Computing* (D.-S. Park, H.-C. Chao, Y.-S. Jeong, and J. J. Park, eds.), vol. 373, pp. 489–495, Singapore: Springer Singapore, 2015.

[25] O. Y. Al-Jarrah, O. Alhussein, P. D. Yoo, S. Muhaidat, K. Taha, and K. Kim, "Data randomization and cluster-based partitioning for botnet intrusion detection," *IEEE transactions on cybernetics*, vol. 46, no. 8, pp. 1796–1806, 2016.

[26] B. Alotaibi and K. Elleithy, "A majority voting technique for wireless intrusion detection systems," in *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pp. 1–6, IEEE, 2016.

[27] G. Folino, F. S. Pisani, and P. Sabatino, "An incremental ensemble evolved by using genetic programming to efficiently detect drifts in cyber security datasets," in *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, pp. 1103–1110, 2016.

[28] B. A. Tama and K.-H. Rhee, "Classifier ensemble design with rotation forest to enhance attack detection of ids in wireless network," in *2016 11th Asia Joint Conference on Information Security (AsiaJCIS)*, pp. 87–91, IEEE, 2016.

[29] P. Mehetrey, B. Shahriari, and M. Moh, "Collaborative ensemble-learning based intrusion detection systems for clouds," in *2016 International Conference on Collaboration Technologies and Systems (CTS)*, pp. 404–411, IEEE, 2016.

[30] A. A. Aburomman and M. B. I. Reaz, "Ensemble of binary svm classifiers based on pca and lda feature extraction for intrusion detection," in *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pp. 636–640, IEEE, 2016.

[31] B. Kiranmai and A. Damodaram, "Extenuate ddos attacks in cloud," in *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pp. 235–238, IEEE, 2016.

[32] Y. Wang, Y. Shen, and G. Zhang, "Research on intrusion detection model using ensemble learning methods," in *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 422–425, IEEE, 2016.

[33] D. Gaikwad and R. Thool, "DAREnsemble: Decision Tree and Rule Learner Based Ensemble for Network Intrusion Detection System," in *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 1* (S. C. Satapathy and S. Das, eds.), vol. 50, pp. 185–193, Cham: Springer International Publishing, 2016.

[34] G. Folino, F. S. Pisani, and P. Sabatino, "A Distributed Intrusion Detection Framework Based on Evolved Specialized Ensembles of Classifiers," in *Applications of Evolutionary Computation* (G. Squillero and P. Burelli, eds.), vol. 9597, pp. 315–331, Cham: Springer International Publishing, 2016.

[35] T. Ma, F. Wang, J. Cheng, Y. Yu, and X. Chen, "A Hybrid Spectral Clustering and Deep Neural Network Ensemble Algorithm for Intrusion Detection in Sensor Networks," *Sensors*, vol. 16, p. 1701, Oct. 2016.

[36] S. T. Miller and C. Busby-Earle, "Multi-perspective machine learning a classifier ensemble method for intrusion detection," in *Proceedings of the 2017 International Conference on Machine Learning and Soft Computing*, pp. 7–12, 2017.

[37] M. H. Kamarudin, C. Maple, T. Watson, and N. S. Safa, "A logitboost-based algorithm for detecting known and unknown web attacks," *IEEE Access*, vol. 5, pp. 26190–26200, 2017.

[38] M. Rajasekaran and A. Ayyasamy, "A novel ensemble approach for effective intrusion detection system," in *2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM)*, pp. 244–250, IEEE, 2017.

[39] W. Chen, F. Kong, F. Mei, G. Yuan, and B. Li, "A novel unsupervised anomaly detection approach for intrusion detection system," in *2017 ieee 3rd international conference on big data security on cloud (bigdatasecurity), IEEE international conference on high performance and smart computing (hpsc), and IEEE international conference on intelligent data and security (ids)*, pp. 69–73, IEEE, 2017.

[40] B. A. Tama, A. S. Patil, and K.-H. Rhee, "An improved model of anomaly detection using two-level classifier ensemble," in *2017 12th Asia Joint Conference on Information Security (AsiaJCIS)*, pp. 1–4, IEEE, 2017.

[41] R. Primartha and B. A. Tama, "Anomaly detection using random forest: A performance revisited," in *2017 International conference on data and software engineering (ICoDSE)*, pp. 1–6, IEEE, 2017.

[42] M. Jabbar, R. Aluvalu, and S. S. S. Reddy, "Cluster based ensemble classification for intrusion detection system," in *Proceedings of the 9th International Conference on Machine Learning and Computing*, pp. 253–257, 2017.

[43] M. M. Aravind and V. Kalaiselvi, "Design of an intrusion detection system based on distance feature using ensemble classifier," in *2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN)*, pp. 1–6, IEEE, 2017.

[44] S. Ruoti, S. Heidbrink, M. O'Neill, E. Gustafson, and Y. R. Choe, "Intrusion detection with unsupervised heterogeneous ensembles using cluster-based normalization," in *2017 IEEE International Conference on Web Services (ICWS)*, pp. 862–865, IEEE, 2017.

[45] M. Belouch and S. E. hadaj, "Comparison of ensemble learning methods applied to network intrusion detection," in *Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing*, pp. 1–4, 2017.

[46] A. Niranjan, A. Prakash, N. Veena, M. Geetha, P. D. Shenoy, and K. Venugopal, "Ebjrv: An ensemble of bagging, j48 and random committee by voting for efficient classification of intrusions," in *2017 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, pp. 51–54, IEEE, 2017.

[47] S. Garg, A. Singh, S. Batra, N. Kumar, and M. S. Obaidat, "Enclass: Ensemble-based classification model for network anomaly detection in massive datasets," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pp. 1–7, IEEE, 2017.

[48] R. R. Reddy, Y. Ramadevi, and K. Sunitha, "Enhanced anomaly detection using ensemble support vector machine," in *2017 International Confer-*

ence on Big Data Analytics and Computational Intelligence (ICBDAC), pp. 107–111, IEEE, 2017.

[49] V. Timčenko and S. Gajin, "Ensemble classifiers for supervised anomaly based network intrusion detection," in *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 13–19, IEEE, 2017.

[50] N. N. Mkuzangwe and F. Nelwamondo, "Ensemble of classifiers based network intrusion detection system performance bound," in *2017 4th International Conference on Systems and Informatics (ICSAI)*, pp. 970–974, IEEE, 2017.

[51] S. A. Ludwig, "Intrusion detection of multiple attack classes using a deep neural net ensemble," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–7, IEEE, 2017.

[52] U. R. Salunkhe and S. N. Mali, "Security Enrichment in Intrusion Detection System Using Classifier Ensemble," *Journal of Electrical and Computer Engineering*, vol. 2017, pp. 1–6, 2017.

[53] S. Tengl, Z. Zhang, L. Teng, W. Zhang, H. Zhu, X. Fang, and L. Fei, "A collaborative intrusion detection model using a novel optimal weight strategy based on genetic algorithm for ensemble classifier," in *2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD))*, pp. 761–766, IEEE, 2018.

[54] X. Yuan, R. Wang, Y. Zhuang, K. Zhu, and J. Hao, "A concept drift based ensemble incremental learning approach for intrusion detection," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 350–357, IEEE, 2018.

[55] C. Sun, K. Lv, C. Hu, and H. Xie, "A double-layer detection and classification approach for network attacks," in *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–8, IEEE, 2018.

[56] Y. Gao, Y. Liu, Y. Jin, J. Chen, and H. Wu, "A novel semi-supervised learning approach for network intrusion detection on cloud-based robotic system," *IEEE Access*, vol. 6, pp. 50927–50938, 2018.

[57] R. K. S. Gautam and E. A. Doegar, "An ensemble approach for intrusion detection system using machine learning algorithms," in *2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 14–15, IEEE, 2018.

[58] F. D. Vaca and Q. Niyaz, "An ensemble learning based wi-fi network intrusion detection system (wnids)," in *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, pp. 1–5, IEEE, 2018.

[59] Y. Shen, K. Zheng, C. Wu, M. Zhang, X. Niu, and Y. Yang, "An ensemble method based on selection using bat algorithm for intrusion detection," *The Computer Journal*, vol. 61, no. 4, pp. 526–538, 2018.

[60] A. H. Mirza, "Computer network intrusion detection using various classifiers and ensemble learning," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, 2018.

[61] N. Marir, H. Wang, G. Feng, B. Li, and M. Jia, "Distributed abnormal behavior detection approach based on deep belief network and ensemble svm using spark," *IEEE Access*, vol. 6, pp. 59657–59671, 2018.

[62] A. Abdullah, R. Ponnan, and D. Asirvatham, "Improving multiclass classification in intrusion detection using clustered linear separator analytics," in *2018 8th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, pp. 32–37, IEEE, 2018.

[63] N. T. Pham, E. Foo, S. Suriadi, H. Jeffrey, and H. F. M. Lahza, "Improving performance of intrusion detection system using ensemble methods and feature selection," in *Proceedings of the Australasian Computer Science Week Multiconference*, pp. 1–6, 2018.

[64] B. Zhang, Y. Yu, and J. Li, "Network intrusion detection based on stacked sparse autoencoder and binary tree ensemble method," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, IEEE, 2018.

[65] S. Zwane, P. Tarwireyi, and M. Adigun, "Performance analysis of machine learning classifiers for intrusion detection," in *2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC)*, pp. 1–5, IEEE, 2018.

[66] A. Muallem, S. Shetty, L. Hong, and J. W. Pan, "Tddeht: Threat detection using distributed ensembles of hoeffding trees on streaming cyber datasets," in *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, pp. 1–6, IEEE, 2018.

[67] M. A. Jabbar, K. Srinivas, and S. Sai Satyanarayana Reddy, "A Novel Intelligent Ensemble Classifier for Network Intrusion Detection System," in *Proceedings of the Eighth International Conference on Soft Computing and Pattern Recognition (SoCPaR 2016)* (A. Abraham, A. K. Cherukuri, A. M. Madureira, and A. K. Muda, eds.), vol. 614, pp. 490–497, Cham: Springer International Publishing, 2018.

[68] A. Parvat, S. Dev, S. Kadam, and J. Chavan, "Network Intrusion Detection System Using Ensemble of Binary Deep Learning Classifiers," in *Smart Trends in Information Technology and Computer Communications* (A. Deshpande, A. Unal, K. Passi, D. Singh, M. Nayak, B. Patel, and S. Pathan, eds.), vol. 876, pp. 3–10, Singapore: Springer Singapore, 2018.

[69] S. Kaur and I. Garg, "Ensemble Technique Based on Supervised and Unsupervised Learning Approach for Intrusion Detection," in *Advances in Computing and Data Sciences* (M. Singh, P. K. Gupta, V. Tyagi, J. Flusser, and T. Ören, eds.), vol. 905, pp. 228–238, Singapore: Springer Singapore, 2018.

[70] I. S. Thaseen, C. A. Kumar, and A. Ahmad, "Integrated Intrusion Detection Model Using Chi-Square Feature Selection and Ensemble of Classifiers," *Arabian Journal for Science and Engineering*, vol. 44, pp. 3357–3368, Apr. 2018.

[71] N. Moustafa, B. Turnbull, and K.-K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4815–4830, 2018.

[72] M. Labonne, A. Olivereau, B. Polvé, and D. Zeghlache, "A cascade-structured meta-specialists approach for neural network-based intrusion detection," in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–6, IEEE, 2019.

[73] D. Liang, Q. Liu, B. Zhao, Z. Zhu, and D. Liu, "A clustering-svm ensemble method for intrusion detection system," in *2019 8th International Symposium on Next Generation Electronics (ISNE)*, pp. 1–3, IEEE, 2019.

[74] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," *IEEE Access*, vol. 7, pp. 82512–82521, 2019.

[75] F. Salo, M. Injadat, A. Moubayed, A. B. Nassif, and A. Essex, "Clustering enabled classification using ensemble feature selection for intrusion detection," in *2019 International Conference on Computing, Networking and Communications (ICNC)*, pp. 276–281, IEEE, 2019.

[76] F. J. P. Montalbo and E. D. Festijo, "Comparative analysis of ensemble learning methods in classifying network intrusions," in *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*, pp. 431–436, IEEE, 2019.

[77] S. Das, A. M. Mahfouz, D. Venugopal, and S. Shiva, "Ddos intrusion detection through machine learning ensemble," in *2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pp. 471–477, IEEE, 2019.

[78] W. He, H. Li, and J. Li, "Ensemble feature selection for improving intrusion detection classification accuracy," in *Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science*, pp. 28–33, 2019.

[79] A. Binbusayyis and T. Vaiyapuri, "Identifying and benchmarking key features for cyber intrusion detection: An ensemble approach," *IEEE Access*, vol. 7, pp. 106495–106513, 2019.

[80] L. Lu, S. Teng, W. Zhang, Z. Zhang, D. Liu, and X. Fang, "Error-correcting ability based collaborative multi-layer selective classifier ensemble model for intrusion detection," in *2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 4–9, IEEE, 2019.

[81] P. Illy, G. Kaddoum, C. M. Moreira, K. Kaur, and S. Garg, "Securing fog-to-things environment using intrusion detection system based on ensemble learning," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–7, IEEE, 2019.

[82] J. Sharma, C. Giri, O.-C. Granmo, and M. Goodwin, "Multi-layer intrusion detection system with ExtraTrees feature selection, extreme learning machine ensemble, and softmax aggregation," *EURASIP Journal on Information Security*, vol. 2019, p. 15, Dec. 2019.

[83] S. M. Mousavi, V. Majidnezhad, and A. Naghipour, "A new intelligent intrusion detector based on ensemble of decision trees," *Journal of Ambient Intelligence and Humanized Computing*, Nov. 2019.

[84] B. Hu, J. Wang, Y. Zhu, and T. Yang, "Dynamic Deep Forest: An Ensemble Classification Method for Network Intrusion Detection," *Electronics*, vol. 8, p. 968, Aug. 2019.

[85] Y.-F. Hsu, Z. He, Y. Tarutani, and M. Matsuoka, "Toward an online network intrusion detection system based on ensemble learning," in *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*, pp. 174–178, IEEE, 2019.

[86] B. A. Tama, M. Comuzzi, and K.-H. Rhee, "Tse-ids: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system," *IEEE Access*, vol. 7, pp. 94497–94507, 2019.

[87] G. Kumar, "An improved ensemble approach for effective intrusion detection," *The Journal of Supercomputing*, vol. 76, pp. 275–291, Jan. 2020.

[88] W. Lian, G. Nie, B. Jia, D. Shi, Q. Fan, and Y. Liang, "An Intrusion Detection Method Based on Decision Tree-Recursive Feature Elimination in Ensemble Learning," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–15, Nov. 2020.

[89] S. Rajagopal, P. P. Kundapur, and K. S. Hareesha, "A Stacking Ensemble for Network Intrusion Detection Using Heterogeneous Datasets," *Security and Communication Networks*, vol. 2020, pp. 1–9, Jan. 2020.

[90] F. L. Aryeh and B. K. Alese, "A Multi-layer Stack Ensemble Approach to Improve Intrusion Detection System's Prediction Accuracy," in *2020 15th International Conference for Internet Technology and Secured Transactions (ICITST)*, (London, United Kingdom), pp. 1–6, IEEE, Dec. 2020.

[91] S. Das, M. Ashrafuzzaman, F. T. Sheldon, and S. Shiva, "Network Intrusion Detection using Natural Language Processing and Ensemble Machine Learning," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, (Canberra, ACT, Australia), pp. 829–835, IEEE, Dec. 2020.

[92] S. Divakar, R. Priyadarshini, and B. Kishore Mishra, "A Robust Intrusion Detection System using Ensemble Machine Learning," in *2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*, (Bhubaneswar, India), pp. 344–347, IEEE, Dec. 2020.

[93] Q. R. S. Fitni and K. Ramli, "Implementation of Ensemble Learning and Feature Selection for Performance Improvements in Anomaly-Based Intrusion Detection Systems," in *2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, (Bali, Indonesia), pp. 118–124, IEEE, July 2020.

[94] S. R. Khonde and V. Ulagamuthalvi, "Ensemble and Feature Selection-based Intrusion Detection System for Multi-attack Environment," in *2020 5th International Conference on Computing, Communication and Security (ICCCS)*, (Patna, India), pp. 1–8, IEEE, Oct. 2020.

[95] A. S. Kyatham, M. A. Nichal, and B. S. Deore, "A novel approach for network intrusion detection using probability parameter to ensemble machine learning models," in *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 608–613, IEEE, 2020.

[96] Z. Lin and D. Hongle, "Research on SDN intrusion detection based on online ensemble learning algorithm," in *2020 International Conference on Networking and Network Applications (NaNA)*, (Haikou City, China), pp. 114–118, IEEE, Dec. 2020.

[97] S. Nandi, S. Maity, and M. Das, "NIDF: An Ensemble-inspired Feature Learning Framework for Network Intrusion Detection," in *2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*, (Bhubaneswar, India), pp. 9–12, IEEE, Dec. 2020.

[98] S. Otoum, B. Kantarci, and H. T. Mouftah, "A Novel Ensemble Method for Advanced Intrusion Detection in Wireless Sensor Networks," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, (Dublin, Ireland), pp. 1–6, IEEE, June 2020.

[99] A. Rai, "Optimizing a New Intrusion Detection System Using Ensemble Methods and Deep Neural Network," in *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, (Tirunelveli, India), pp. 527–532, IEEE, June 2020.

[100] M. M. Rashid, J. Kamruzzaman, M. Ahmed, N. Islam, S. Wibowo, and S. Gordon, "Performance Enhancement of Intrusion detection System Using Bagging Ensemble Technique with Feature Selection," in *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, (Gold Coast, Australia), pp. 1–5, IEEE, Dec. 2020.

[101] X. Shi, Y. Cai, and Y. Yang, "Extreme trees network intrusion detection framework based on ensemble learning," in *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA)*, (Dalian, China), pp. 91–95, IEEE, Aug. 2020.

[102] J. Yang, Y. Sheng, and J. Wang, "A GBDT-Paralleled Quadratic Ensemble Learning for Intrusion Detection System," *IEEE Access*, vol. 8, pp. 175467–175482, 2020.

[103] J. Zhang, F. Li, and F. Ye, "An Ensemble-based Network Intrusion Detection Scheme with Bayesian Deep Learning," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, (Dublin, Ireland), pp. 1–6, IEEE, June 2020.

[104] C. Iwendi, S. Khan, J. H. Anajemba, M. Mittal, M. Alenezi, and M. Alazab, "The Use of Ensemble Models for Multiple Class and Binary Class Classification for Improving Intrusion Detection Systems," *Sensors*, vol. 20, p. 2559, Apr. 2020.

[105] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, "Hybrid Intrusion Detection System Based on the Stacking Ensemble of C5 Decision Tree Classifier and One Class Support Vector Machine," *Electronics*, vol. 9, p. 173, Jan. 2020.

[106] A. Mahfouz, A. Abuhussein, D. Venugopal, and S. Shiva, "Ensemble Classifiers for Network Intrusion Detection Using a Novel Network Attack Dataset," *Future Internet*, vol. 12, p. 180, Oct. 2020.

[107] N. Martindale, M. Ismail, and D. A. Talbert, "Ensemble-Based Online Machine Learning Algorithms for Network Intrusion Detection Systems Using Streaming Data," *Information*, vol. 11, p. 315, June 2020.

[108] T. Feng, M. Dou, P. Xie, and J. Fang, "Network intrusion detection based on data feature dynamic ensemble model," in *International Conference on Artificial Intelligence and Security*, pp. 661–673, Springer, 2020.

[109] M. Abirami, U. Yash, and S. Singh, "Building an ensemble learning based algorithm for improving intrusion detection system," in *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, pp. 635–649, Springer, 2020.

[110] G. Folino, F. S. Pisani, and L. Pontieri, "A GP-based ensemble classification framework for time-changing streams of intrusion detection data," *Soft Computing*, vol. 24, pp. 17541–17560, Dec. 2020.

[111] H. Rajadurai and U. D. Gandhi, "A stacked ensemble learning model for intrusion detection in wireless network," *Neural Computing and Applications*, May 2020.

[112] B. S. Bhati and C. Rai, "Ensemble based approach for intrusion detection using extra tree classifier," in *Intelligent computing in engineering*, pp. 213–220, Springer, 2020.

[113] A. Sadiwala, K. Rathore, Y. Shah, H. Shah, and K. Srivastava, "Intrusion detection system against malign packets—a comparative study between autoencoder and ensemble model," in *Advanced Computing Technologies and Applications*, pp. 165–175, Springer, 2020.

[114] J. Wei, C. Long, J. Li, and J. Zhao, "An intrusion detection algorithm based on bag representation with ensemble support vector machine in cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 32, Dec. 2020.

[115] O. Alkadi, N. Moustafa, B. Turnbull, and K.-K. R. Choo, "A deep blockchain framework-enabled collaborative intrusion detection for protecting iot and cloud networks," *IEEE Internet of Things Journal*, 2020.

[116] N. Lower and F. Zhan, "A study of ensemble methods for cyber security," in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 1001–1009, IEEE, 2020.

[117] F. Folino, G. Folino, and L. Pontieri, "A p2p environment to validate ensemble-based approaches in the cybersecurity domain," in *2020 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pp. 344–351, IEEE, 2020.

[118] H. Jiang, Z. He, G. Ye, and H. Zhang, "Network intrusion detection based on pso-xgboost model," *IEEE Access*, vol. 8, pp. 58392–58401, 2020.

[119] O. O. Olasehinde, O. V. Johnson, and O. C. Olayemi, "Evaluation of selected meta learning algorithms for the prediction improvement of network intrusion detection system," in *2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS)*, pp. 1–7, IEEE, 2020.

[120] B. A. Tama, L. Nkenyereye, S. R. Islam, and K.-S. Kwak, "An enhanced anomaly detection in web traffic using a stack of classifier ensemble," *IEEE Access*, vol. 8, pp. 24120–24134, 2020.

[121] O. J. Mebawondu, O. D. Alowolodu, A. O. Adetunmbi, and J. O. Mebawondu, "Optimizing the classification of network intrusion detection using ensembles of decision trees algorithm," in *International Conference on Information and Communication Technology and Applications*, pp. 286–300, Springer, 2020.

[122] P. Kumar, G. P. Gupta, and R. Tripathi, "A distributed ensemble design based intrusion detection system using fog computing to protect the internet of things networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 9555–9572, Oct. 2021.

[123] G. Srivastava, T. R. G, N. Deepa, B. Prabadevi, and P. K. Reddy M, "An ensemble model for intrusion detection in the Internet of Softwarized Things," in *Adjunct Proceedings of the 2021 International Conference on Distributed Computing and Networking*, (Nara Japan), pp. 25–30, ACM, Jan. 2021.

[124] B. S. Bhati, G. Chugh, F. Al-Turjman, and N. S. Bhati, "An improved ensemble based intrusion detection technique using XGBoost," *Transactions on Emerging Telecommunications Technologies*, vol. 32, June 2021.

[125] Z. Zhao, L. Ge, and G. Zhang, "A novel DBN-LSSVM ensemble method for intrusion detection system," in *2021 9th International Conference on Communications and Broadband Networking*, (Shanghai China), pp. 101–107, ACM, Feb. 2021.

[126] Y. Zhao and G. Gan, "Research on Intrusion Detection Technology Based on Ensemble Learning," in *International Conference on Frontiers of Electronics, Information and Computation Technologies*, (Changsha China), pp. 1–6, ACM, May 2021.

[127] T. Acharya, I. Khatri, A. Annamalai, and M. F. Chouikha, "Efficacy of Heterogeneous Ensemble Assisted Machine Learning Model for Binary and Multi-Class Network Intrusion Detection," in *2021 IEEE International Conference on Automatic Control & Intelligent Systems (I2CACIS)*, (Shah Alam, Malaysia), pp. 408–413, IEEE, June 2021.

[128] N. Ahmed and R. Durga, "A Trust Aware Behavioral Based Intrusion Detection in Cloud Environment Using Ensemble Service Centric Featured Neural Network," in *2021 4th International Conference on Computing and Communications Technologies (ICCCT)*, (Chennai, India), pp. 342–349, IEEE, Dec. 2021.

[129] Ainurrochman, A. Nugroho, R. Wahyuwidayat, S. T. Sianturi, M. Fauzi, M. F. Ramadhan, B. A. Pratomo, and A. M. Shiddiqi, "Ensemble Methods Classifier Comparison for Anomaly Based Intrusion Detection System on CIDDS-002 Dataset," in *2021 13th International Conference on Information & Communication Technology and System (ICTS)*, (Surabaya, Indonesia), pp. 62–67, IEEE, Oct. 2021.

[130] S. Ennaji, N. E. Akkad, and K. Haddouch, "A Powerful Ensemble Learning Approach for Improving Network Intrusion Detection System (NIDS)," in *2021 Fifth International Conference On Intelligent Computing in Data Sciences (ICDS)*, (Fez, Morocco), pp. 1–6, IEEE, Oct. 2021.

[131] T. T. Khoei, G. Aissou, W. C. Hu, and N. Kaabouch, "Ensemble Learning Methods for Anomaly Intrusion Detection System in Smart Grid," in *2021 IEEE International Conference on Electro Information Technology (EIT)*, (Mt. Pleasant, MI, USA), pp. 129–135, IEEE, May 2021.

[132] A. Z. Kiflay, A. Tsokanos, and R. Kirner, "A Network Intrusion Detection System Using Ensemble Machine Learning," in *2021 International Carnahan Conference on Security Technology (ICCST)*, (Hatfield, United Kingdom), pp. 1–6, IEEE, Oct. 2021.

[133] X. Li, M. Zhu, L. T. Yang, M. Xu, Z. Ma, C. Zhong, H. Li, and Y. Xiang, "Sustainable Ensemble Learning Driving Intrusion Detection Model," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2021.

[134] H. Li and D. Chasaki, "Ensemble Machine Learning for Intrusion Detection in Cyber-Physical Systems," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, (Vancouver, BC, Canada), pp. 1–2, IEEE, May 2021.

[135] Z. Lin, "Network intrusion detection based of semi-supervised ensemble learning algorithm for imbalanced data," in *2021 International Conference on Networking and Network Applications (NaNA)*, pp. 338–344, IEEE, 2021.

[136] S. M. Nzuva, L. Nderu, and T. Mwalili, "Ensemble Model for Enhancing Classification Accuracy in Intrusion Detection Systems," in *2021 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, (Cape Town, South Africa), pp. 1–7, IEEE, Dec. 2021.

[137] P. Parkar, "A Network Intrusion Detection System Based on Ensemble Machine Learning Techniques," in *2021 IEEE 2nd International Conference on Applied Electromagnetics, Signal Processing, & Communication (AESPC)*, (Bhubaneswar, India), pp. 1–6, IEEE, Nov. 2021.

[138] S. Seth, K. K. Chahal, and G. Singh, "A Novel Ensemble Framework for an Intelligent Intrusion Detection System," *IEEE Access*, vol. 9, pp. 138451–138467, 2021.

[139] V. Sidharth and C. R. Kavitha, "Network Intrusion Detection System Using Stacking and Boosting Ensemble Methods," in *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, (Coimbatore, India), pp. 357–363, IEEE, Sept. 2021.

[140] E. Jaw and X. Wang, "Feature Selection and Ensemble-Based Intrusion Detection System: An Efficient and Comprehensive Approach," *Symmetry*, vol. 13, p. 1764, Sept. 2021.

[141] H.-C. Lin, P. Wang, K.-M. Chao, W.-H. Lin, and Z.-Y. Yang, "Ensemble Learning for Threat Classification in Network Intrusion Detection on a Security Monitoring System for Renewable Energy," *Applied Sciences*, vol. 11, p. 11283, Nov. 2021.

[142] A. Wang, W. Wang, H. Zhou, and J. Zhang, "Network Intrusion Detection Algorithm Combined with Group Convolution Network and Snapshot Ensemble," *Symmetry*, vol. 13, p. 1814, Sept. 2021.

[143] A. Subasi, S. Algebsani, W. Alghamdi, E. Kremic, J. Almaasrani, and N. Abdulaziz, "Intrusion detection in smart healthcare using bagging ensemble classifier," in *International Conference on Medical and Biological Engineering*, pp. 164–171, Springer, 2021.

[144] M. A. Bertoni, G. H. de Rosa, and J. R. F. Brega, "Optimum-path forest stacking-based ensemble for intrusion detection," *Evolutionary Intelligence*, May 2021.

[145] D. Mulimani, S. G. Totad, P. Patil, and S. V. Seeri, "Adaptive ensemble learning with concept drift detection for intrusion detection," in *Data Engineering and Intelligent Computing*, pp. 331–339, Springer, 2021.

[146] A. P. Psathas, L. Iliadis, A. Papaleonidas, and D. Bountas, "A hybrid deep learning ensemble for cyber intrusion detection," in *International Conference on Engineering Applications of Neural Networks*, pp. 27–41, Springer, 2021.

[147] S. R. Khonde, G. Kulanthaivel, and V. Ulagamuthalvi, "An ensemble approach for intrusion detection in collaborative attack environment," in *Smart Computing Techniques and Applications*, pp. 137–146, Springer, 2021.

[148] A. J. Siddiqui and A. Boukerche, "Adaptive ensembles of autoencoders for unsupervised IoT network intrusion detection," *Computing*, vol. 103, pp. 1209–1232, June 2021.

[149] P. Singh and V. Ranga, "Attack and intrusion detection in cloud computing using an ensemble learning approach," *International Journal of Information Technology*, vol. 13, pp. 565–571, Apr. 2021.

[150] M. Yousefnezhad, J. Hamidzadeh, and M. Aliannejadi, "Ensemble classification for intrusion detection via feature extraction based on deep Learning," *Soft Computing*, vol. 25, pp. 12667–12683, Oct. 2021.

[151] T. N. Thinh, T. H. Q. Bao, D.-M. Ngo, and C. Pham-Quoc, "High-performance anomaly intrusion detection system with ensemble neural networks on reconfigurable hardware," *Concurrency and Computation: Practice and Experience*, May 2021.

[152] L. Chen, S. E. Weng, C. J. Peng, Y. C. Li, H. H. Shuai, and W. H. Cheng, "The hierarchical ensemble model for network intrusion detection in the real-world dataset," vol. 2022-May, 2022.

[153] G. Fu, B. Li, Y. Yang, and Q. Wei, "A multi-distance ensemble and feature clustering based feature selection approach for network intrusion detection," 2022.

[154] M. S. Hossen, M. J. Hossain, M. A. Masud, M. Samsuzzaman, and C. Bepery, "Anomaly based network intrusion detection using ensemble classifiers," in *2022 4th International Conference on Sustainable Technologies for Industry 4.0 (STI)*, pp. 1–6, IEEE, 2022.

[155] K. Gosai, H. Mehta, and V. Katkar, "An intrusion detection using ensemble classifiers," 2022.

[156] J. Zheng, X. Ni, L. Li, K. Yu, and J. Zhang, "An ensemble learning-based two-level network intrusion detection method," 2022.

[157] F. Li, W. Ma, H. Li, and J. Li, "Improving intrusion detection system using ensemble methods and over-sampling technique," 2022.

[158] T. Sun, K. Yan, T. Li, X. Lu, and Q. Dong, "A network anomaly intrusion detection method based on ensemble learning for library e-learning platform," 2022.

[159] M. V. Tayde, R. B. Adhao, and V. Pachghare, "Ensemble based feature selection technique for flow based intrusion detection system," 2022.

[160] H. Siddharthan, T. Deepa, and P. Chandhar, "Senmqtt-set: An intelligent intrusion detection in iot-mqtt networks using ensemble multi cascade features," *IEEE Access*, vol. 10, 2022.

[161] L. Yang, Y. Song, S. Gao, A. Hu, and B. Xiao, "Griffin: Real-time network intrusion detection system via ensemble of autoencoder in sdn," *IEEE Transactions on Network and Service Management*, vol. 19, 2022.

[162] Z. Wu, P. Gao, L. Cui, and J. Chen, "An incremental learning method based on dynamic ensemble rvm for intrusion detection," *IEEE Transactions on Network and Service Management*, vol. 19, 2022.

[163] S. Das, S. Saha, A. T. Priyoti, E. K. Roy, F. T. Sheldon, A. Haque, and S. Shiva, "Network intrusion detection and comparative analysis using ensemble machine learning and feature selection," *IEEE Transactions on Network and Service Management*, vol. 19, 2022.

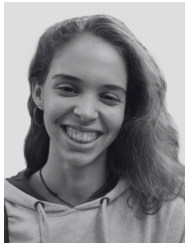
[164] C. Long, J. Xiao, J. Wei, J. Zhao, W. Wan, and G. Du, "Autoencoder ensembles for network intrusion detection," vol. 2022-February, 2022.

[165] W. Yao, L. Hu, Y. Hou, and X. Li, "A two-layer soft-voting ensemble learning model for network intrusion detection," 2022.

[166] J. Thaker, N. K. Jadav, S. Tanwar, P. Bhattacharya, and H. Shahinzadeh, "Ensemble learning-based intrusion detection system for autonomous vehicle," 2022.
[167] L. Yang, A. Shami, G. Stevens, and S. De Rusett, "Lccde: A decision-based ensemble framework for intrusion detection in the internet of vehicles," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*, pp. 3545–3550, IEEE, 2022.
[168] Amarudin, R. Ferdiana, and Widyawan, "Performance of intrusion detection system using bagging ensemble with sdn-base classifier," 2022.
[169] M. Data and M. Aritsugi, "Ab-ht: An ensemble incremental learning algorithm for network intrusion detection systems," 2022.
[170] M. Behravan, N. Zhang, A. Jaekel, and M. Kneppers, "Intrusion detection systems based on stacking ensemble learning in vanet," 2022.
[171] P. K. Danso, E. C. P. Neto, S. Dadkhah, A. Zohourian, H. Molyneaux, and A. A. Ghorbani, "Ensemble-based intrusion detection for internet of things devices," 2022.
[172] L. P. Khan, T. T. Anika, S. I. Hanif, and R. M. Rahman, "Network intrusion detection using stack-ensemble ann," 2022.
[173] J. W. O'Meara, M. S. Elsayed, T. Saber, and A. D. Jurcut, "Sdn intrusion detection: An ensemble approach to reducing false negative rate for novel attacks," 2022.
[174] M. Srivastava, S. S. Yadav, and J. Dheeba, "A novel secured wireless sensor network with ensemble based intrusion detection system and middleware architecture," 2022.
[175] G. A. D. S. Oliveira, P. S. S. Lima, F. Kon, R. Terada, D. M. Batista, R. Hirata, and M. Hamdan, "A stacked ensemble classifier for an intrusion detection system in the edge of iot and iiot networks," 2022.
[176] U. Mbasuva and G. A. L. Zodi, "Designing ensemble deep learning intrusion detection system for ddos attacks in software defined networks," 2022.
[177] T. J. Lucas, K. A. D. Costa, R. Scherer, and J. P. Papa, "An ensemble pruning approach to optimize intrusion detection systems performance," vol. 2022-October, 2022.
[178] Z. A. E. Houda, B. Brik, and L. Khoukhi, "Ensemble learning for intrusion detection in sdn-based zero touch smart grid systems," 2022.
[179] A. Iacovazzi and S. Raza, "Ensemble of random and isolation forests for graph-based intrusion detection in containers," 2022.
[180] S. A. Abdulkareem, C. H. Foh, H. Lee, F. Carrez, and K. Moessner, "Iot network intrusion detection with ensemble learners," vol. 2022-October, 2022.
[181] Y. Zhang, Y. Gandhi, Z. Li, and Z. Xiao, "Improving the classification effectiveness of network intrusion detection using ensemble machine learning techniques and deep neural networks," 2022.
[182] L. Liu and J. Li, "A blockchain-assisted collaborative ensemble learning for network intrusion detection," in *2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 1042–1047, IEEE, 2022.
[183] B. Yin, B. Bu, B. Gao, and Q. Li, "A hybrid intrusion detection method using improved stacking ensemble algorithm and false positive elimination strategy for cbtc," vol. 2022-October, 2022.
[184] H. Rajadurai and U. D. Gandhi, "A stacked ensemble learning model for intrusion detection in wireless network," *Neural Computing and Applications*, vol. 34, 2022.
[185] H. N. Abdoli, A. J. Bidgoly, and S. Fallah, "Intrusion detection system using soft labeling and stacking ensemble," *International Journal of Information Technology (Singapore)*, vol. 14, 2022.
[186] M. Rashid, J. Kamruzzaman, T. Imam, S. Wibowo, and S. Gordon, "A tree-based stacking ensemble technique with feature selection for network intrusion detection," *Applied Intelligence*, vol. 52, 2022.
[187] U. Zahoora, M. Rajarajan, Z. Pan, and A. Khan, "Zero-day ransomware attack detection using deep contractive autoencoder and voting based ensemble classifier," *Applied Intelligence*, vol. 52, 2022.
[188] W. Huang, X. Zhao, and X. Huang, "Embedding and extraction of knowledge in tree ensemble classifiers," *Machine Learning*, vol. 111, 2022.
[189] H. Yu and Q. Dai, "Dwe-il: a new incremental learning algorithm for non-stationary time series prediction via dynamically weighting ensemble learning," *Applied Intelligence*, vol. 52, 2022.
[190] S. Qiao, N. Han, F. Huang, K. Yue, T. Wu, Y. Yi, R. Mao, and C. an Yuan, "Lmnnb: Two-in-one imbalanced classification approach by combining metric learning and ensemble learning," *Applied Intelligence*, vol. 52, 2022.
[191] Z. Yang, Z. Liu, X. Zong, and G. Wang, "An optimized adaptive ensemble model with feature selection for network intrusion detection," *Concurrency and Computation: Practice and Experience*, vol. 35, 2023.
[192] K. D. Devprasad, S. Ramanujam, and S. B. Rajendran, "Context adaptive ensemble classification mechanism with multi-criteria decision making for network intrusion detection," *Concurrency and Computation: Practice and Experience*, vol. 34, 2022.
[193] S. Krishnaveni, S. Sivamohan, S. Sridhar, and S. Prabhakaran, "Network intrusion detection based on ensemble classification and feature selection method for cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 34, 2022.
[194] T. T. H. Le, H. Kim, H. Kang, and H. Kim, "Classification and explanation for intrusion detection system based on ensemble trees and shap method," *Sensors*, vol. 22, 2022.
[195] R. Gangula, V. M. Mohan, and M. R. Kumar, "Network intrusion detection system for internet of things based on enhanced flower pollination algorithm and ensemble classifier," *Concurrency and Computation: Practice and Experience*, vol. 34, 2022.
[196] S. J. Bu, H. B. Kang, and S. B. Cho, "Ensemble of deep convolutional learning classifier system based on genetic algorithm for database intrusion detection," *Electronics (Switzerland)*, vol. 11, 2022.
[197] I. Jung, J. Ji, and C. Cho, "Emsm: Ensemble mixed sampling method for classifying imbalanced intrusion detection data," *Electronics (Switzerland)*, vol. 11, 2022.
[198] R. Alghamdi and M. Bellaiche, "Evaluation and selection models for ensemble intrusion detection systems in iot," *Internet of Things*, vol. 3, 2022.
[199] D. N. Mhawi, A. Aldallal, and S. Hassan, "Advanced feature-selection-based hybrid ensemble learning algorithms for network intrusion detection systems," *Symmetry*, vol. 14, 2022.
[200] Y. Shen, K. Zheng, Y. Yang, S. Liu, and M. Huang, "Cba-clsve: A class-level soft-voting ensemble based on the chaos bat algorithm for intrusion detection," *Applied Sciences (Switzerland)*, vol. 12, 2022.
[201] P. Liao, J. Yan, J. M. Sellier, and Y. Zhang, "Tada: A transferable domain-adversarial training for smart grid intrusion detection based on ensemble divergence metrics and spatiotemporal features," *Energies*, vol. 15, 2022.
[202] X. Wang, "Enidrift: A fast and adaptive ensemble system for network intrusion detection under real-world drift," 2022.
[203] S. Osken, E. N. Yildirim, G. Karatas, and L. Cuhaci, "Intrusion detection systems with deep learning: A systematic mapping study," in *2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT)*, pp. 1–4, IEEE, 2019.
[204] J. P. Anderson, "Computer security threat monitoring and surveillance," *Technical Report, James P. Anderson Company*, 1980.
[205] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on software engineering*, no. 2, pp. 222–232, 1987.
[206] G. Karatas and O. K. Sahingoz, "Neural network based intrusion detection systems with different training functions," in *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, pp. 1–6, IEEE, 2018.
[207] M. Kaouk, J.-M. Flaus, M.-L. Potet, and R. Groz, "A review of intrusion detection systems for industrial control systems," in *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 1699–1704, IEEE, 2019.
[208] J. Ran, Y. Ji, and B. Tang, "A semi-supervised learning approach to ieee 802.11 network anomaly detection," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pp. 1–5, IEEE, 2019.
[209] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1–6, IEEE, 2009.
[210] I. Sharaf., A. Lashkari, Habibi, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization.," in *ICISSP*, pp. 108–116, 2018.
[211] R. Panigrahi and S. Borah, "A detailed analysis of cicids2017 dataset for designing intrusion detection systems," *International Journal of Engineering & Technology*, vol. 7, no. 3.24, pp. 479–482, 2018.
[212] D. Stiawan, M. Y. B. Idris, A. M. Bamhdi, R. Budiarto, *et al.*, "Cicids-2017 dataset feature analysis with information gain for anomaly detection," *IEEE Access*, vol. 8, pp. 132911–132921, 2020.
[213] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *computers & security*, vol. 31, no. 3, pp. 357–374, 2012.
[214] N. Moustafa, J. Slay, others, *et al.*, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 military communications and information systems conference (MilCIS)*, pp. 1–6, IEEE, 2015.

[215] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2012.
[216] V. Smolyakov, "Ensemble learning to improve machine learning results." https://blog.statsbot.co/ensemble-learning-d1dcd548e936, 2017. (Acesso em 30/09/2019).
[217] S. Khonde and V. Ulagamuthalvi, "A machine learning approach for intrusion detection using ensemble technique-a survey," 2018.
[218] Y. Alsouda, S. Pllana, and A. Kurti, "Iot-based urban noise identification using machine learning: Performance of svm, knn, bagging, and random forest," in *Proceedings of the International Conference on Omni-Layer Intelligent Systems*, pp. 62–67, ACM, 2019.
[219] R. E. Schapire, "The strength of weak learnability," *Machine learning*, vol. 5, no. 2, pp. 197–227, 1990.
[220] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.
[221] J. D'Souza, "A quick guide to boosting in ml - greyatom - medium." shorturl.at/GJM38, 2018. (Acesso em 01/10/2019).
[222] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.
[223] S. Agarwal and C. R. Chowdary, "A-stacking and a-bagging: Adaptive versions of ensemble learning algorithms for spoof fingerprint detection," *Expert Systems with Applications*, vol. 146, p. 113160, 2020.
[224] S. Džeroski and B. Ženko, "Is combining classifiers with stacking better than selecting the best one?," *Machine learning*, vol. 54, no. 3, pp. 255–273, 2004.
[225] C. Aggarwal, "Data classification: Algorithms and applications, ser," *Frontiers in physics. Chapman and Hall/CRC*, 2014.
[226] J. Rocca, "Ensemble methods: bagging, boosting and stacking - towards data science." shorturl.at/itWX0, 2019. (Accesso en 23/04/2020).
[227] W.-M. Lee, *Python Machine Learning*. John Wiley & Sons, 2019.
[228] K. Bakshi and K. Bakshi, "Considerations for artificial intelligence and machine learning: Approaches and use cases," in *2018 IEEE Aerospace Conference*, pp. 1–9, IEEE, 2018.
[229] J. Mueller and L. Massaron, *Aprendizado de Máquina Para Leigos*. Para Leigos, Alta Books, 2019.
[230] R. Baeza-Yates and B. Ribeiro-Neto, *Recuperação de Informação-: Conceitos e Tecnologia das Máquinas de Busca*. Bookman Editora, 2013.
[231] Z. Deng, X. Zhu, D. Cheng, M. Zong, and S. Zhang, "Efficient knn classification algorithm for big data," *Neurocomputing*, vol. 195, pp. 143–148, 2016.
[232] J. Tchaye-Kondi, Y. Zhai, and L. Zhu, "A new hashing based nearest neighbors selection technique for big datasets," *arXiv preprint arXiv:2004.02290*, 2020.
[233] R. Tinós, "Perceptron multicamadas." shorturl.at/bE138, 2020. (Acesso em 04/15/2020).
[234] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.
[235] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.
[236] S. O. Rezende, *Sistemas inteligentes: fundamentos e aplicações*. Editora Manole Ltda, 2003.
[237] H. Sharma and S. Kumar, "A survey on decision tree algorithms of classification in data mining," *International Journal of Science and Research (IJSR)*, vol. 5, no. 4, pp. 2094–2097, 2016.
[238] G. Engelen, V. Rimmer, and W. Joosen, "Troubleshooting an intrusion detection dataset: the cicids2017 case study," in *2021 IEEE Security and Privacy Workshops (SPW)*, pp. 7–12, IEEE, 2021.
[239] J. Mink, H. Benkraouda, L. Yang, A. Ciptadi, A. Ahmadzadeh, D. Votipka, and G. Wang, "Everybody's got ml, tell me what else you have: Practitioners' perception of ml-based security tools and explanations," in *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 2068–2085, IEEE Computer Society, 2023.
[240] M. Catillo, A. Pecchia, and U. Villano, "Machine learning on public intrusion datasets: Academic hype or concrete advances in nids?," in *2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S)*, pp. 132–136, IEEE, 2023.

THIAGO J. LUCAS received the Ph.D. degree in Computer Science from the Advanced Network Security Laboratory of the São Paulo State University "Júlio de Mesquita Filho" (LARS/Unesp) in Bauru/SP. Master in Computer Science from the Paulista State University "Júlio de Mesquita Filho" (Institute of Biosciences, Letters and Exact Sciences - Unesp) in São José do Rio Preto/SP. Specialization in Design and Implementation of Computer Networks by the Federal Technological University of Paraná. Degree in Information Security from the Higher Education College of Technology of Ourinhos. He holds the permanent public job of Professor of Higher Education at the Higher Education College of Technology of Ourinhos. Also is a Cybersecurity instructor at "Hacker Security" (SP). He has high school technical graduate in Electronics at the State Technical School of Ourinhos.



INAÊ S. DE FIGUEIREDO received the B.Sc. in computer science from São Paulo State University, Bauru, Brazil, in 2023 and is currently pursuing a Master's degree in computer science at the same university. From 2021 to 2023, she received a research grant from the São Paulo Research Foundation (FAPESP) to research intrusion detection with deep learning as a Scientific Initiation Student. She is a member of the advanced search group in computer network security (LARS). Her research interests include cybersecurity and its intersection with quantum computing, as well as machine learning for intrusion and anomaly detection.

CARLOS A. C. TOJEIRO received the B.Sc. degree in Systems Analysis and Information Technology from the Higher Education College of Technology of Ourinhos 2008. He holds an MBA in Business Management from the Faculty of Higher Education of Santa Bárbara in 2011, is a Specialist in Teaching in Higher Education from Faculdade Estácio de Sá in 2015 and a Specialist in Computer Network Security from the Higher Education College of Technology of Ourinhos in 2015. He holds a Pedagogical Degree for Teachers for Middle-Level Professional Education equivalent to a Full Degree from the State Center for Technological Education of the State of São Paulo.

ALEX M. G. DE ALMEIDA received the Ph.D. degree in 2021 from the São Paulo State University "Julio de Mesquita Filho". Holds a degree in Data Processing from the State Center for Technological Education Paula Souza (1999), a M.Sc. degree in Computer Science from the State University of Londrina (2016). He is currently a course coordinator at the State Center for Technological Education "Paula Souza" and associate professor at the State Center for Technological Education "Paula Souza". He has experience in Computer Science, with an emphasis on Information Systems, working mainly on the following topics: machine learning, information security, sentiment analysis, voice recognition and text mining.

RAFAL SCHERER received the Ph.D. degree in computer science (Methods of Classification Using Neuro-Fuzzy Systems) at the Department of Mechanical Engineering and Computer Science of Czestochowa University of Technology. Is an associate professor at the Institute of Computational Intelligence at the Częstochowa University of Technology. His research focuses on developing new methods in computational intelligence and data mining, ensembling methods in machine learning, content-based image indexing. He authored more than 80 research papers. He was a principal investigator of the Polish Ministry of Science and Higher Education project Computational Intelligence Methods in Data Mining and a researcher in the Polish-Singapore Research Project "Development of intelligent techniques for modeling, controlling and optimizing complex manufacturing systems". He authored a book on multiple classification techniques published in Springer. He was a reviewer for major computational intelligence journals. Scherer earned M.Sc. degree in electrical engineering at Department of Electrical Engineering.

JOÃO PAULO PAPA received the Ph.D. degree in Computer Science from the State University of Campinas in 2009. Bachelor's degree in Information Systems from the São Paulo State University in 2002, a M.Sc. degree in Computer Science from the Federal University of São Carlos (2005) and postgraduate-PhD in Computer Science from the State University of Campinas (2009) and Harvard University (2015). He is Associate Professor III at the Department of Computing at the São Paulo State University, and a collaborating researcher at the Department of Computing at the Federal University of São Carlos. He is currently the senior editor of the IEEE Signal Processing Letters journal and a member of the editorial board of Computer Methods in Biomechanics and Biomedical Engineering, Journal of Next Generation Information Technology, and International Journal of Bio-Inspired Computation. He is co-editor of a book in the area of optimization by metaheuristics, as well as collaborations with several international institutions. He has a senior IEEE member.

JOSÉ REMO F. BREGA received the Ph.D. degree in Transport Engineering from the University of São Paulo in 1997 and holds a degree in Civil Engineering from the University of São Paulo (1986), a degree in Technology in Data Processing from the Federal University of São Carlos (1987), a Master's degree in Geotechnics from the University of São Paulo (1991). He is currently a professor at the Paulista State University Júlio de Mesquita Filho (Unesp) at the Department of Computing at the Faculty of Sciences in Bauru. He has experience in Computer Science, with an emphasis on Information Visualization and Interfaces, working mainly on the following topics: Interaction with Visualizations, Interfaces for Visualization, Virtual Reality and Distributed Virtual Environments. He is an accredited professor at the Graduate Program in Computer Science at Unesp. He was responsible for coordinating Unesp's institutional systems between 2012 and 2014.

KELTON A. P. DA COSTA received the Ph.D. degree from the São Paulo University (USP) and holds a B.Sc. degree in Systems Analysis from the "Sagrado Coração" University - USC, M.Sc. Degree in Computer Science from the "Euripides Foundation" of Marilia University - UNIVEM. Post-doctoral in Computer Networks by Campinas State University-UNICAMP and post-doctoral in Anomaly Detection in Computer Networks by São Paulo State University-UNESP. Professor of Computer Science and Information Systems courses at São Paulo State University (UNESP-Bauru). Professor of the Master's and Doctoral Program in Computer Science at UNESP (Bauru), and has to experience in Computer Science, with emphasis on Systems Architecture Computing and Distributed Systems, working mainly on the following topics: Management in Computer Networks, Computer Security, Anomaly Detection Systems and Signatures in Computer Networks, and Data Flow Analysis in Computer Networks.