mathematics

MDPI

Review

Performance Assessment of Supervised Classifiers for Designing Intrusion Detection Systems: A Comprehensive Review and Recommendations for Future Research

Ranjit Panigrahi [1,†], Samarjeet Borah [1], Akash Kumar Bhoi [2], Muhammad Fazal Ijaz [3,†], Moumita Pramanik [1], Rutvij H. Jhaveri [4] and Chiranji Lal Chowdhary [5,*]

[1] Department of Computer Applications, Sikkim Manipal Institute of Technology, Sikkim Manipal University, Majitar 737136, Sikkim, India; ranjit.panigrahi@gmail.com (R.P.); samarjeetborah@gmail.com (S.B.); moumita.pramanik@gmail.com (M.P.)
[2] Department of Electrical and Electronics Engineering, Sikkim Manipal Institute of Technology, Sikkim Manipal University, Majitar 737136, Sikkim, India; akashkrbhoi@gmail.com
[3] Department of Intelligent Mechatronics Engineering, Sejong University, Seoul 05006, Korea; fazal@sejong.ac.kr
[4] Department of Computer Science and Engineering, School of Technology, Pandit Deendayal Energy University, Gandhinagar 382007, India; rutvij.jhaveri@sot.pdpu.ac.in
[5] School of Information Technology & Engineering, Vellore Institute of Technology, Vellore 632014, India
* Correspondence: chiranji.lal@vit.ac.in
† These authors contributed equally to this work and are first co-authors.

Citation: Panigrahi, R.; Borah, S.; Bhoi, A.K.; Ijaz, M.F.; Pramanik, M.; Jhaveri, R.H.; Chowdhary, C.L. Performance Assessment of Supervised Classifiers for Designing Intrusion Detection Systems: A Comprehensive Review and Recommendations for Future Research. *Mathematics* **2021**, *9*, 690. https://doi.org/10.3390/math9060690

Academic Editor: Daniel Gómez Gonzalez

Received: 17 February 2021
Accepted: 20 March 2021
Published: 23 March 2021

Abstract: Supervised learning and pattern recognition is a crucial area of research in information retrieval, knowledge engineering, image processing, medical imaging, and intrusion detection. Numerous algorithms have been designed to address such complex application domains. Despite an enormous array of supervised classifiers, researchers are yet to recognize a robust classification mechanism that accurately and quickly classifies the target dataset, especially in the field of intrusion detection systems (IDSs). Most of the existing literature considers the accuracy and false-positive rate for assessing the performance of classification algorithms. The absence of other performance measures, such as model build time, misclassification rate, and precision, should be considered the main limitation for classifier performance evaluation. This paper's main contribution is to analyze the current literature status in the field of network intrusion detection, highlighting the number of classifiers used, dataset size, performance outputs, inferences, and research gaps. Therefore, fifty-four state-of-the-art classifiers of various different groups, i.e., Bayes, functions, lazy, rule-based, and decision tree, have been analyzed and explored in detail, considering the sixteen most popular performance measures. This research work aims to recognize a robust classifier, which is suitable for consideration as the base learner, while designing a host-based or network-based intrusion detection system. The NSLKDD, ISCXIDS2012, and CICIDS2017 datasets have been used for training and testing purposes. Furthermore, a widespread decision-making algorithm, referred to as Techniques for Order Preference by Similarity to the Ideal Solution (TOPSIS), allocated ranks to the classifiers based on observed performance reading on the concern datasets. The J48Consolidated provided the highest accuracy of 99.868%, a misclassification rate of 0.1319%, and a Kappa value of 0.998. Therefore, this classifier has been proposed as the ideal classifier for designing IDSs.

Keywords: classifiers ranking; class-imbalance learning; IDS; IDS base learner; intrusion detection systems; network-based IDS

1. Introduction

The footprint of artificial intelligence-enabled Internet of Things (IoT) devices [1] in our day-to-day life attracts hackers and potential intrusions. In 2017, WannaCry ransomware, a self-propagating malware, devastatingly impacted computing resources by infecting more

than 50,000 NHS systems [2]. The network threats such as WannaCry become a nightmare for the security manager and remain an open research area. Many intrusion detection schemes have been proposed to counter malicious activities in a computer network [3–6]. All the network anomaly counter mechanisms are either unsupervised, supervised, or a combination of both. The supervised algorithms are rigorously used to design state-of-the-art intrusion detectors. This is because the ability to learn from examples makes the supervised classifiers robust and powerful. In data science, an array of supervised classifiers exists, and each one of them claims to be the best among others. However, in the real world of classification processes, the scenario is somewhat different. The supervised classifiers are susceptible to misclassification if overfit or underfit during the training process [7]. Another aspect is a class-imbalance issue [8] in the underlying dataset of a classification model. A supervised classifier always favors the majority class if the training is incorporated on a high class-imbalance dataset [9,10]. Apart from the class-imbalance issue, the data purity also decides the performance of the supervised classifiers. The data are stored and are available in numerous formats and include several outliers such as missing class information, NULL, and NaN values. The raw data with outliers drastically limit the performance of the classifiers. The classifiers reveal unrealistic results with the data of outliers [11,12]. This leads to the development of robust and versatile classifiers for impure data. In this regard, numerous researchers are concerned about pattern recognition, and data extraction [13,14], which is the main objective of data mining, and perhaps one of the motivational aspects for exploring [15–17] supervised machine learning algorithms. Numerous classification mechanisms are available in the literature to handle impure data, especially in designing full-bodied network intrusion detection systems (IDS). However, the central question of the researchers is associated with the selection of the optimum classifiers to develop a base learner for IDS.

Furthermore, there is a lack of a standard guideline to select the most suitable classifier for their datasets. Multiple studies have been conducted on the before-mentioned problem. However, most of the proposed studies available in the literature evaluate the classifiers using standard performance measures such as classification accuracy and false-positive rate [18–22]. It is worth mentioning that the quality of a classifier does not depend only on the classification accuracy. Other performance measures such as misclassification rate, precision, recall, and F-Score empirically define the classifier's performance quality. Therefore, it is necessary to study a comprehensive review that can be used as a guideline to analyze classifiers using various performance measures in various datasets. Therefore, the main objective of this paper is to examine several research papers in the field of host-based and network-based intrusion detection considering multiple aspects. This study analyzes the type of classification used, the datasets used to consider the sample size, performance measures discussed in evaluating classifier performance, inferences, and research gaps encountered.

Moreover, the proposed study provides a guideline for designing a host-based or network-based intrusion detection system. This study's main contribution is to present an in-depth analysis of fifty-four widely used classifiers considering six different classifier groups across thirteen performance measures. These classifiers are comprehensively analyzed through three well-recognized binary and multiclass NSLKDD, ISCXIDS2012, and CICIDS2017 datasets. The decision-making algorithm referred to as Techniques for Order Preference by Similarity to the Ideal Solution (TOPSIS) [23,24] is incorporated as a reliable feature to allocate weight to these classifiers. These weights are subsequently used for ranking the performance of the classifiers. Consequently, the best classifier for a dataset and the best of each group of classifiers is proposed. Moreover, the best classifier across all the datasets is suggested as the most generic classifier for designing an IDS.

The research of this analysis is structured as follows. In Section 2, the most recent study of supervised classifiers is delineated; the materials and methods has been mentioned in Section 3. Furthermore, in Section 4, the results of the analysis has been discussed. Section 5 is dedicated for J48Consolidated classifier, followed by the conclusion in Section 6.

2. Related Works

Supervised classifiers are extensively used in the field of network security. The most potential applications of machine learning techniques are in risk assessment after the deployment of various security apparatus [25], identifying risks associated with various network attacks and in predicting the extent of damage a network threat can do. Apart from these, supervised classification techniques have been explored and analyzed by numerous researchers in a variety of application areas. Most of those studies' analyses focused on a detailed exploration to validate a theory or performance evaluation to come across a versatile classifier [26–28]. The performance of supervised classifiers has been explored in intrusion detection [29], robotics [18], semantic web [19], human posture recognition [30], face recognition [20], biomedical data classification [31], handwritten character recognition [22] and land cover classification [21]. Furthermore, an innovative semi-supervised heterogeneous ensemble classifier called Multi-train [32] was also proposed, where a justifiable comparison was made with other supervised classifiers, such as k-Nearest Neighbour (kNN), J48, Naïve Bayes, and random tree. Multi-train was also successfully achieved, and its prediction accuracy of unlabeled data was improved, which, therefore, can reduce the risk of incorrectly labeling the unlabeled data. A study on this topic, which exclusively deals with classifiers' accuracy measures using multiple standard datasets, is proposed by Labatut et al. [33]. An empirical analysis of supervised classifiers was carried out by Caruana et al. [34] using eleven datasets with eight performance measures, where the calibrated boosted trees appeared as the best learning algorithm. Besides, a systematic analysis of supervised classifiers was carried out by Amancio et al. [35] under varying classifiers' settings.

The focus of this paper is to analyze the performance of various supervised classifiers using IDS datasets. Therefore, the authors have decided to review related articles in the literature that examined different classifiers using IDS datasets. The classifier analysis is expected to provide a platform for the researchers to devise state-of-the-art IDSs and quantitative risk assessment schemes for various cyber defense systems. Numerous studies and their detailed analytical findings related to supervised classifiers have been outlined in Table 1.

Table 1 summarizes the taxonomy of analyzed articles. In the last column, an attempt has been made to outline the inferences/limitation or research gaps encountered. The summarization of these analyses provides scope for meta-analysis about the supervised classifiers, which ultimately shows direction or justification for further investigation in the field of supervised classification using intrusion detection datasets. From Table 1, it has been observed that the decision tree and function-based approaches are mostly explored. The usage statistics of supervised classifiers are presented in Figure 1.

According to Figure 1, J48 (C4.5) and Random Forest of decision trees and function-based SVM and Multilayer Perceptron (Neural Network) have been analyzed considerably by numerous researchers. In this work, the authors have tried to understand the reason behind decision trees' popularity and function-based approaches. Therefore, the authors have summarized the performance metrics results used to explore those classifiers in the analyzed papers. Most of the researchers focused on accuracy scores; therefore, the authors used the accuracy score as a base measure to understand the reason behind the use of decision trees and function-based classifiers.

Therefore, in this study, the authors have calculated the minimum, maximum, and average accuracy of Bayes, Decision trees, Functions, Lazy, and Rules group of classifiers concerning the literature outlined in Table 1. The calculated detection accuracy of the research papers surveyed is presented in Figure 2. In Figure 2, almost all groups of classifiers show a maximum accuracy rate of more than 99%.

Table 1. Detailed findings and analysis of supervised classifiers.

Inferences/Observations/ Limitations/Research Gaps	With 20 features, BayesNet shows the highest amount of accuracy of 99.3% for classifying DDoS attacks, and PART shows 98.9% for classifying Probe attacks. No class imbalance issue was found. Tested on an older dataset, which is now obsolete. Completely ignored U2R and R2L attacks. Hence, classifiers performance may vary with the inclusion of U2R and R2L instances	Gaussian classifier seems to be effective for R2L and Probe attacks with the highest detection rate of 0.136 and 0.874, respectively. Naïve Bayes proved suitable for U2R attacks with the highest detection rate of 0.843, Decision Tree and Random Forest classified DoS attacks with the highest detection rate of 0.972. Considering the highest detection rate of three training sets is not convincing. Instead, the average detection rate could have highlighted better classifiers for the given scenario.	A decent number of performance measures were used to analyze the classifiers, Other state-of-the-art classifiers are missing from the comparison. Dataset sample size, number of features considered are not precise. Although the Naïve Bayes proved to be a better classifier in FP Rate, the ID3 performs far ahead than the Naïve Bayes. Class imbalance issues are not considered during evaluation.	The accuracy of the induction tree is promising, with an overall rate of 99.839/%. Although it is appreciable that the induction tree performs well in the class imbalance KDD'99 dataset, the size of the training set and the class-wise breakup of training stances are not precise. The reason for considering different training instances for three different classifiers is not clear. Considering the ROC area, it is evident that the Induction tree correctly classified Neptune, Smurf, pod, teardrop, port sweep, and back attack instances.	C4.5 scores the highest average accuracy of 64.94% as compared to 62.7% of SVM. Considering attacks accuracy, C4.5 seems to be suitable for detecting Probe, DoS, and U2R attacks, whereas SVM classifies R2L threats better. Class imbalance issue is not addressed.	J48 (C4.5) proved to be an accurate classifier for classifying test instances. Data extraction and the preprocessing procedure is not clearly defined. The training set is a high-class imbalance, so the evaluation of the classifiers in terms of accuracy and detection rate is not sufficient.
Performance measures used	Accuracy Kappa Mean Absolute Error Root Mean Squared Error	Detection Rate	Accuracy, Kappa, RMSE, Precision Recall, FP Rate Precision, Recall FN Rate, F-Measure	Accuracy, MA Error, RMS Error RA Error, RRS Error, TP Rate FP Rate, Precision, Recall, F-Measure, ROC Area	Accuracy Detection Rate FP Rate	Accuracy Detection Rate FP Rate
Dataset, Features and Sample Size	Dataset: KDD'99 FS procedure: Information Gain Number of Features Selected: 20 Training instances: 492,842 Testing Instances: N/A	Dataset: KDD'99 Features Selected: All features Training instances: 270,000 Testing Instances: 311,029	Dataset: KDD'99 Features Selected: All features	Dataset: KDD'99 Features Selected: All features Training instances: N/A Testing Instances: 19,870	Dataset: KDD'99 Features Selected: All features Training instances: N/A	Dataset: KDD'99 Features Selected: All features Training instances: 311,029 Testing Instances: 494,014
Classification Type	Multi Class Normal DoS Probe	Multi Class Normal DoS Probe U2R R2L	Multi-Class Normal DoS Probe U2R R2L	Multi-Class Normal Neptune Smurf guess_passwd Pod	Multi-Class Normal DoS Probe U2R R2L	Multi-Class Normal DoS Probe U2R R2L
Classifiers Evaluated	J48 (C4.5), BayesNet, Naïve Bayes, Part, Multilayer Perceptron, SVM	Gaussian, Naïve Bayes, Decision Tree (C4.5), Random Forest	Naïve Bayes J48(C4.5) ID3	Induction Tree Naïve Bayes ANN	C4.5 SVM	SVM J48 (C4.5) Multilayer Perceptron
Author/Year/Reference	Araar et al. (2005) [36]	Gharibian et al. (2007) [37]	Panda et al. (2008) [38]	Srinivasulu et al. (2009) [39]	Wu et al. (2009) [40]	Jalil et al. (2010) [41]
Inferences/Observations/ Limitations/Research Gaps	Random Forest appears to be effective for detecting DoS and Probe attacks. NB Tree is useful for detecting R2L and U2R attacks The classifiers' performances are measured in a binary environment. Performance many vary in a multiclass environment with a very high-class imbalance rate.	C5.0 decision tree shows the highest detection rate of 98.75% for the KDD dataset's testing samples. Both DoS and Probe attacks are detected with 99.56% and 97.25% of the detection rate. The sample size and the basis of selecting the sample size in not defined in the research.	J48 evolved as the best classifier with 99.13% accuracy. OneR is very fast in classifying instances. The basis of sampling, training, and testing size is not mentioned. How the classifiers will behave in a class imbalance situation is not defined.	Brilliantly evaluated. It can be extended to other groups of classifiers. NBTree achieves 97.76% highest accuracy.	Random Forest proves to provide a high accuracy rate for classifying threats. Considering 15 features, Random Forest shows an accuracy rate of 99.8% for Normal, 99.1% for DoS, 98.9% for Probe, 98.7% for U2R, and 97.9% for R2L. Average accuracy of Random forest achieves 98.88% for 15 features of NSL-KDD dataset	kNN proved to be the best classifier in terms of accuracy. No benchmark datasets were used for the evaluation of classifiers. Class imbalance issue has not been explored.
Performance measures used	Accuracy Detection Rate FP Rate Testing time	Detection Rate	Testing time, Accuracy, TP Rate FP Rate, MA Error, RMS Error, RA Error, RRS Error	Training time, Accuracy, MAE, RMSE, Kappa, Recall, Precision, F-Measure, Precision, FP Rate	Accuracy	Accuracy

Table 1. *Cont.*

Dataset, Features and Sample Size	Dataset: KDD'99 Feature Selection Technique: CFS Features: 7 Training instances: N/A Testing Instances: N/A	Dataset: KDD'99 Feature Selection Technique: N/A Training instances: N/A Testing Instances: N/A	Dataset: NSLKDD Training instances: N/A Testing instances: 2747	Dataset: NSL-KDD Feature election Techniques: CONS: 12 features, CFS: 3 features Training instances: 25,192 Testing instances: 11,850	Dataset: NSL-KDD Feature Selection Techniques: CFS Features: 15 Training instances: 125,937 Testing instances: 22,544	Dataset: Artificial Dataset Feature Selection Scheme: CFS Features: 2 to 10 Training instances: N/A Testing instances: N/A
Classification Type	Binary Class Normal Instances of any one other class.	Multi-Class Normal DoS Probe	N/A	Multi-Class Normal DoS Probe U2R R2L	Multi-Class Normal DoS Probe U2R R2L	N/A
Classifiers Evaluated	J48, Naïve Bayes, NB Tree, Random Forest	SVM, Ripper Rule, C5.0 decision tree	Naive Bayes, J48, OneR, PART, RBF Network	ADTree, C4.5, LADTree, NBTree, Random Tree, Random Forest, REP Tree	Random Forest J48 SVM CART Naïve Bayes	Naïve Bayes, Bayes Net, C4.5, Random Forest, CART, kNN, Logistic Regression, MLP, SVM
Author/Year/Reference	Amudha et al. (2011) [42]	Naidu et al. (2012) [43]	Kalyani et al. (2012) [44]	Thaseen et al. (2013) [45]	Revathi et al. (2013) [46]	Amancio et al. (2014) [35]
Inferences/Observations/ Limitations/Research Gaps	Random Forest shows the highest accuracy of 97.75% and 100% for the LLsDDoS and CAIDA Conficker dataset. J48 and Random Forest both show equal highest accuracy of 99.26% for the CAIDA DDoS 2007 dataset. Class imbalance issue has not been addressed. The type of classification, whether binary or multiclass, is not clear.	Random Forest shows the highest amount of accuracy of 91.52%. Considering False Positive Rate, BayesNet seems to be better. The test could have been conducted with varying sample sizes or with the maximum sample size possible to confirm the suitable classifier.	Proposed two IDS models for classifying the different type of attack instances. Random Forest and Fuzzy Logic seem to be ideal classifiers for classifying various attacks. The training time of a classifier does not provide a clear picture of designing an IDS. Hence, testing time per instance would provide a precise result.	PART shows the highest accuracy of 99.97% Many other prominent classifiers are missed from the evaluation. Tested on an obsolete dataset. Declaring the best classifier just based on accuracy may not reveal the real capabilities of the classifier. Other measures, such as ROC and PRC values, should be considered for judging the classifiers' performance in class imbalance learning.	Random Forest proved to be the best classifier, among others. The class imbalance issue found as NSL-KDD is a class imbalance dataset. A similar test on other state-of-the-art classifiers are required	Random Forest shows the highest accuracy of 93.77% Class imbalance issues found with Normal-U2R and Normal-R2L instances. Tested on an obsolete dataset
Performance measures used	Accuracy FN Rate FP Rate Precision Recall	Training Time, Sensitivity, Specificity, Accuracy, FP Rate, Kappa, F-Measure, Precision, ROC,	TP Rate FP Rate Training Time	Accuracy, Recall, Precision, F-Measure, TP Rate, TN Rate ROC Area Kappa	Accuracy F-Measure ROC Value Precision Recall	Accuracy, FP Rate, FN Rate, TP Rate, Precision, ROC value, RMS Error
Dataset, Features and Sample Size	Datasets: LLsDDoS, CAIDA, DdoS2007, Conficker Feature Selection Procedure: Manual, Features Selected: 7 Training and Testing Instances: N/A	Dataset: NSL-KDD Features Selected: All features Training instances: 1166 Testing instances: 7456	Dataset: KDD'99 Feature Selection Technique: Information Gain Features: 20 Training and Testing instances: N/A	Dataset: KDD'99 No. of Features: All features Training and Testing instances: N/A	Dataset: NSL-KDD No. of Features selected: All Training and Testing instances: N/A	Dataset: KDD'99 No. of features: All Training instances: 148,753 Testing instances: 60,000
Classification Type	N/A	Multi-Class, Normal, DoS Probe U2R R2L	Multi-Class Normal, DoS Probe U2R R2L	Multi-Class Normal DoS Probe U2R R2L	Multi-Class Normal DoS Probe U2R R2L	Multi-Class Normal DoS Probe U2R R2L
Classifiers Evaluated	Naïve Bayes, RBF Network, Multilayer Perceptron, BayesNet, IBK, J48 (C4.5), Random Forest	BayesNet, Logistic, IBk, JRip, PART, J48, Random Forest, Random Tree, REPTree	Bayes Net, Naïve Bayes, C4.5, ID3, NBTree, Fuzzy Logic SVM, Decision Table, JRip, OneR, MLP, IBk	Decision Table, JRip, ZeroR, OneR, PART, BayesNet, Naïve Bayes, MLP, SMO, Simple Logistic, IBk Kstar, LWL	Logistic Regression Gaussian Naïve Bayes, SVM, Random Forest	J48 (C4.5), Random Forest, Random Tree, Decision Table, Multilayer Perceptron, Naïve Bayes, BayesNet
Author/Year/Reference	Robinson et al. (2015) [47]	Choudhury et al. (2015) [48]	Jain et al. (2016) [49]	Bostani et al. (2017) [50]	Belavagi et al. (2016) [51]	Almseidin et al. (2017) [52]

Table 1. *Cont.*

Inferences/Observations/ Limitations/Research Gaps	The best classifier to classify attacks of the NSL-KDD dataset in an anomalous traffic condition: DOS attacks—Multilayer Perceptron, Probe attacks—BFTree, U2R attacks—J48, R2L attacks—Naïve Bayes. Overall, all the classifiers except Naïve Bayes worked well with the NSL-KDD dataset. No performance measures were used to validate the classifiers in this class imbalance situation; therefore, the classifier seems to be ideal, but it may not be consistent in this scenario.	Decision Tree shows the highest accuracy of 99%. Class imbalance issue, not present. Class wise samples contradict the total training data size.	Random Forest proved to be the best classifier, among others. The class imbalance issue found as NSL-KDD was a class imbalance dataset. Similarly, the U2R and R2L attacks were not perfectly detected due to inherent class-imbalance issue. A similar test on other state-of-the-art classifiers is required	With all the features of the NSL-KDD dataset, the J48 classifier outperforms all other classifiers. With a reduced feature set through information gain feature selection, the IBk seems to be a better classifier. The under-sampling of highly dominant classes and over a sampling of poor classes improves the detection accuracy of R2L and U2R attacks.	The two-class decision forest model evolved as the best detection scheme with a detection accuracy of 99.2%. The generic, exploits, shellcode, and worms attacks were also detected with 99%, 94.49%, 91.79% and 90.9% accuracy, respectively. The evaluation has been carried out with the cutting-edged Microsoft Azure Machine Learning Studio to handle huge instances of the UNSW NB-15 dataset.	The Random Forest emerged as the best classifier for multi attacks scenarios. On the other hand, in a binary attack scenario, the C4.5 was found to be the best classifier for detection.
Performance measures used	Accuracy FP Rate TP Rate FN Rate Precision Recall F-Score	Accuracy Recall Precision F-Measure	Accuracy F-Measure Precision Recall	Accuracy, True Positive Rate, False Positive Rate, Precision, Recall F-Measure, ROC Area	Accuracy, Precision, Recall, F1-Score, AUC, False Alarm Rate, Training Time, Testing Time	Detection Rate, True Negative Rate, False Alarm Rate, Accuracy, Training Time, Testing Time
Dataset, Features and Sample Size	Dataset: NSL-KDD Feature Selection Technique: Sequential Floating Forward Selection (SFFS), No of Features: 26 Training instances: 125,973 Testing instances: 22,544	Dataset: CICIDS 2017 Feature Selection Techniques: Fisher Score, No of Features: 30, Training instances: 203,171 Testing instances: 22,575	Dataset: KDD' 99, NSL-KDD, No. of features: All Testing instances: KDD' 99 Sample Size: 494,021 NSL-KDD Sample Size: 125,973	Dataset: NSL-KDD Separately evaluated on Information Gain Feature Selection and All Features, 10-fold cross validation on instances of the dataset	Dataset: UNSW NB-15 Feature Selection Scheme: Mutual information Training samples: 1,75,341 Testing samples: 82,332	Dataset: CICIDS2017, Feature Selection Techniques: Manual feature selection. Features having unique values for each instance of the dataset has been considered. Training instances: 40,000 Testing instances: 40,000
Classification Type	Multi-Class Normal DoS Probe U2R R2L	Binary Benign DoS	Multi-Class Normal DoS Probe U2R R2L	Multi-Class Normal DoS Probe U2R R2L	Multi-Class Normal, Analysis, Backdoor, Reconnaissance, Shellcode, Worms, DOS, Fuzzers, Generic, Exploits	Multi-Class Benign, DoS, PortScan, Bot, Brute Force, Web Attacks, Infiltration
Classifiers Evaluated	Naïve Bayes BF Tree J48 Multilayer Perceptron NB Tree RFT	SVM IBk(k-NN) Decision Tree	Random Forest J48 (C4.5) BayesNet Naïve Bayes SVM	Naïve Bayes, Logistic Regression, MLP, SVM, IBk, J48 (C4.5)	Average Perceptron, Bayes point machine, Boosted Decision Tree, Decision Forest, Decision Jungle, Locally deep SVM, Logistic Regression	J48 (C4.5), ForestPA, Random Forest, REP Tree, Jrip, FURIA, RIdor, MLP, RBF, LIBSVM, SVM, Naïve Bayes
Author/Year/Reference	Aziz et al. (2017) [53]	Aksu et al. (2018) [54]	Nehra et al. (2019) [55]	Mahfouz et al. (2020) [56]	Rajagopal et al. (2020) [57]	Ahmim et al. (2020) [58]

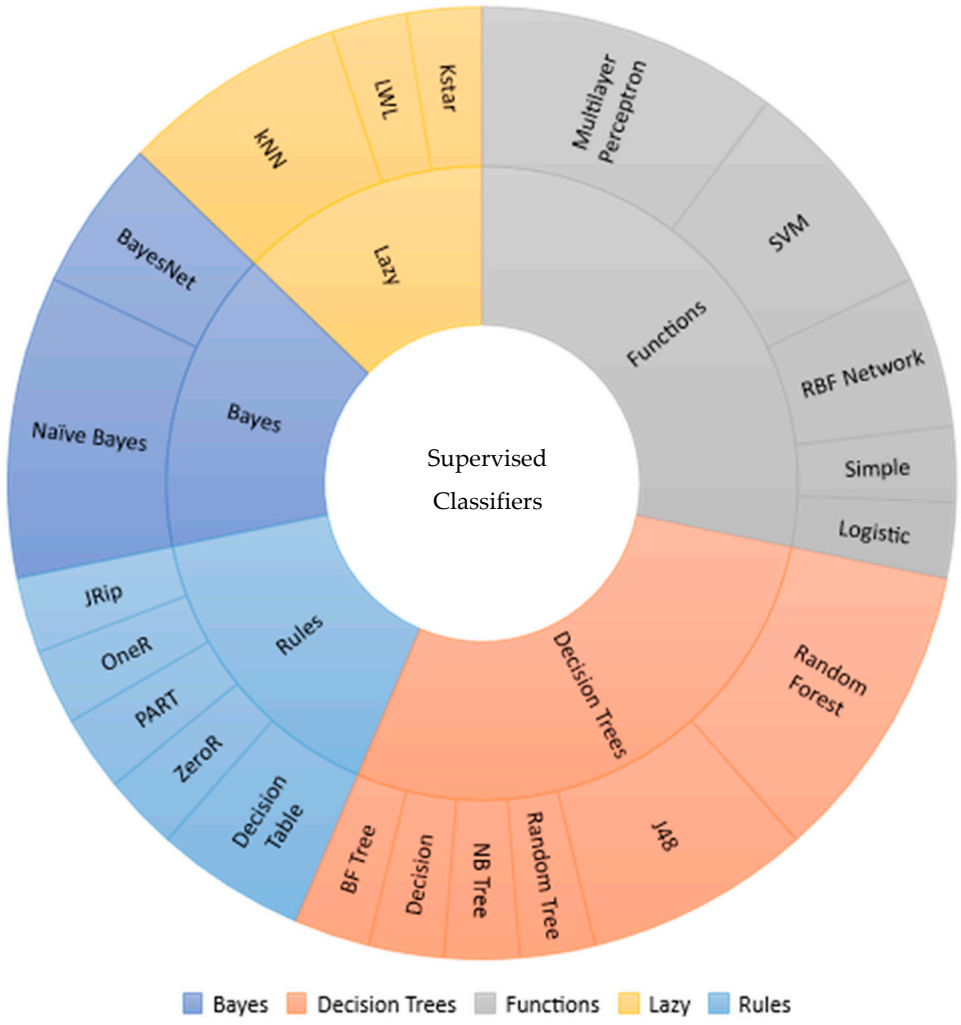

Figure 1. Usage statistics of supervised classifiers.

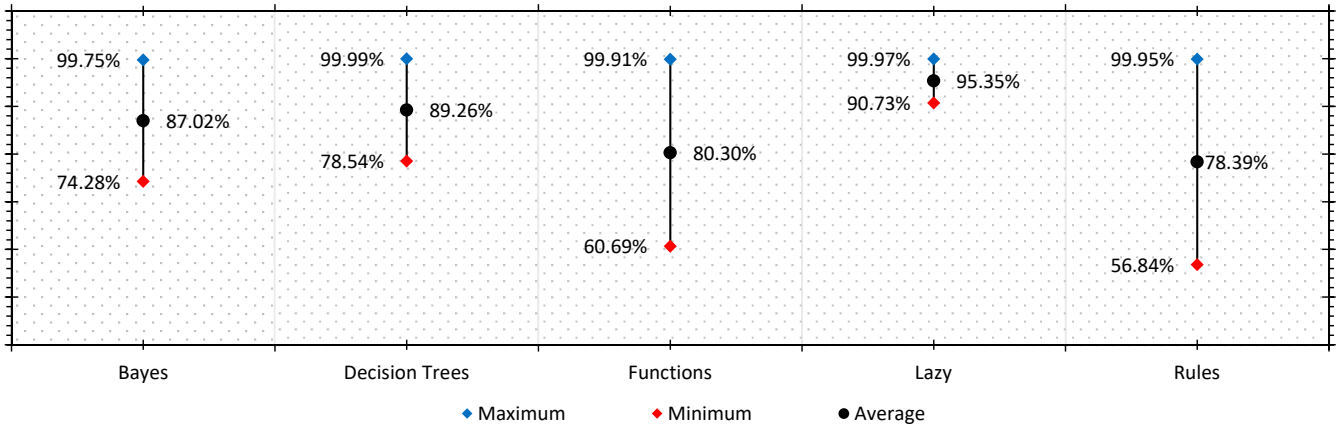

Figure 2. Comparison of classification accuracy in various classifier groups found in the literature.

Similarly, considering the average accuracy, the Lazy classifiers are far ahead of different groups of classifiers. Despite having an impressive accuracy rate, the Lazy group classifiers were deeply analyzed by a handful of researchers [48–50]. On the other hand, decision trees and function-based classifiers were the center point of many research papers. Consequently, in this paper, the authors have decided to explore multiple classifiers of all the classifier groups. In this work, fifty-four state-of-the-art classifiers of six different classi-

fier groups were analyzed. The classifier groups were created based on their functionality and the guidelines presented by Frank et al. [59]. The classifiers under evaluation and their groups are presented in Tables 2–7 under six different classifier groups.

Table 2. Bayes classifiers for evaluation.

Sl. No.	Name of Classifiers	Short Name
1	Discriminative Multinomial Naive Bayes [60]	DMNB
2	Hidden Markov Models [61,62]	HMM
3	Naive Bayes [63,64]	NB
4	Sparse Generative Model [65]	SGM

Table 3. Functions classifiers.

Sl. No.	Name of Classifiers	Short Name
1	Linear Discriminant Analysis [66]	LDA
2	LibLINEAR [67]	LLNR
3	LibSVM [68]	LSVM
4	Logistic Regression [69]	LR
5	Multilayer Perceptron—With one hidden layer [70]	MLPH
6	Multilayer Perceptron—Back Propagation Neural Network [71]	MLPB
7	Quadratic Discriminant Analysis [72]	QDA
8	Radial Basis Function [73]	RBF
9	Radial Basis Function Network [74]	RBFN
10	Simple Logistic Regression [75]	SLR
11	Sequential Minimal Optimization [76,77]	SMO

Table 4. Lazy group classifiers.

Sl. No.	Name of Classifiers	Short Name
1	IB1 (Nearest Neighbor approach) [78]	IB1
2	IBk (k-nearest neighbor approach) [78]	IBK
3	IBkLG (k-nearest neighbor with Log and Gaussian kernel) [78]	IBKLG
4	KStar [79]	KSTAR
5	Local Knn [80]	LKNN
6	Locally Weighted Learning [81,82]	LWL
7	Rseslib Knn [80]	RLKNN

Table 5. Rule-based classifiers.

Sl. No.	Name of Classifiers	Short Name
1	Conjunctive Rule [83]	CR
2	Decision Table [84]	DTBL
3	Decision Table Naïve Bayes hybrid classifier [85]	DTNB
4	Fuzzy Rule Induction [86]	FURIA
5	JRip [87]	JRIP
6	MODLEM [88]	MODLEM
7	Nearest Neighbor with Generalization [89,90]	NNGE
8	Ordinal Learning Method [91]	OLM
9	OneR [92]	ONER
10	PART [93]	PART
11	RIpple-DOwn Rule learner [94]	RIDOR
12	Rough Set [95]	ROUGHS
13	ZeroR [96]	ZEROR

Table 6. Decision tree classifiers.

Sl. No.	Name of Classifiers	Short Name
1	Best-First Decision Tree [97]	BFT
2	Criteria Based Decision Tree [98]	CDT
3	ForestPA [99]	FPA
4	Functional Tree [100]	FT
5	J48 [101]	J48
6	J48Consolidated [101–103]	J48C
7	J48Graft [104]	J48G
8	Logit Boost-based Alternating Decision Tree [105]	LADT
9	Logistic Model Trees [106,107]	LMT
10	Naïve Bayes based Decision Tree [108]	NBT
11	Reduces Error Pruning Tree [109]	REPT
12	Random Forest [110,111]	RF
13	Random Tree [111]	RT
14	Simple Cart [112]	SC
15	SysFor [113]	SF

Table 7. Miscellaneous classifiers.

Sl. No.	Name of Classifiers	Short Name
1	Composite Hypercubes on Iterated Random Projections [114]	CHIRP
2	Fuzzy Lattice Reasoning [115]	FLR
3	Hyper Pipes [116]	HP
4	Voting Feature Intervals [117]	VFI

3. Materials and Methods

The authors used Weka 3.8.1 [59] software in a CentOS platform on the Param Shavak supercomputing facility provided by the Centre for Development of Advanced Computing (CDAC), India. The supercomputing system consists of 64 GB RAM with two multicore CPUs, each with 12 cores having a performance of 2.3 Teraflops. To evaluate all the classifiers of Tables 2–7, the authors have considered samples of NSLKDD [118–120], ISCXIDS2012 [121], and CICIDS2017 [122] datasets. The training and testing sample size for each dataset is outlined in Table 8. The training and testing samples were generated with 66% and a 34% split of the total sample size.

Table 8. Miscellaneous classifiers.

Datasets	Sample Size	Training Instances	Testing Instances
NSLKDD	7781	5135	2646
ISCXIDS2012	5494	3626	1868
CICIDS2017	8917	5885	3032

All three NSLKDD, CICIDS2017 and ISCXIDS2012, have a high-class imbalance. Additionally, NSLKDD and CICIDS2017 are multi-class, and the ISCXIDS2012 dataset contains binary class information. The performance of a classifier cannot be explored only through its accuracy and detection rate. Therefore, the authors have considered a variety of performance measures such as training time, testing time, model accuracy, misclassification rate, kappa, mean absolute error, root mean squared error, relative absolute error, root relative squared error, true positive rate, false-positive rate, precision, and receiver operating curve (ROC). The ROC value reveals the real performance on class imbalance datasets such as the CICIDS2017 and the NSL-KDD. Similarly, the Matthews correlation coefficient (MCC) and precision-recall curve (PRC) are useful for evaluating binary classification on the ISCXIDS2012 dataset.

The experiment for evaluating classifiers covers five essential steps [123], such as dataset selection, classification, weight calculation using multi-criteria decision making, weight to rank transformation, and finally, global rank generation. Figure 3 shows the methodology used by the authors.

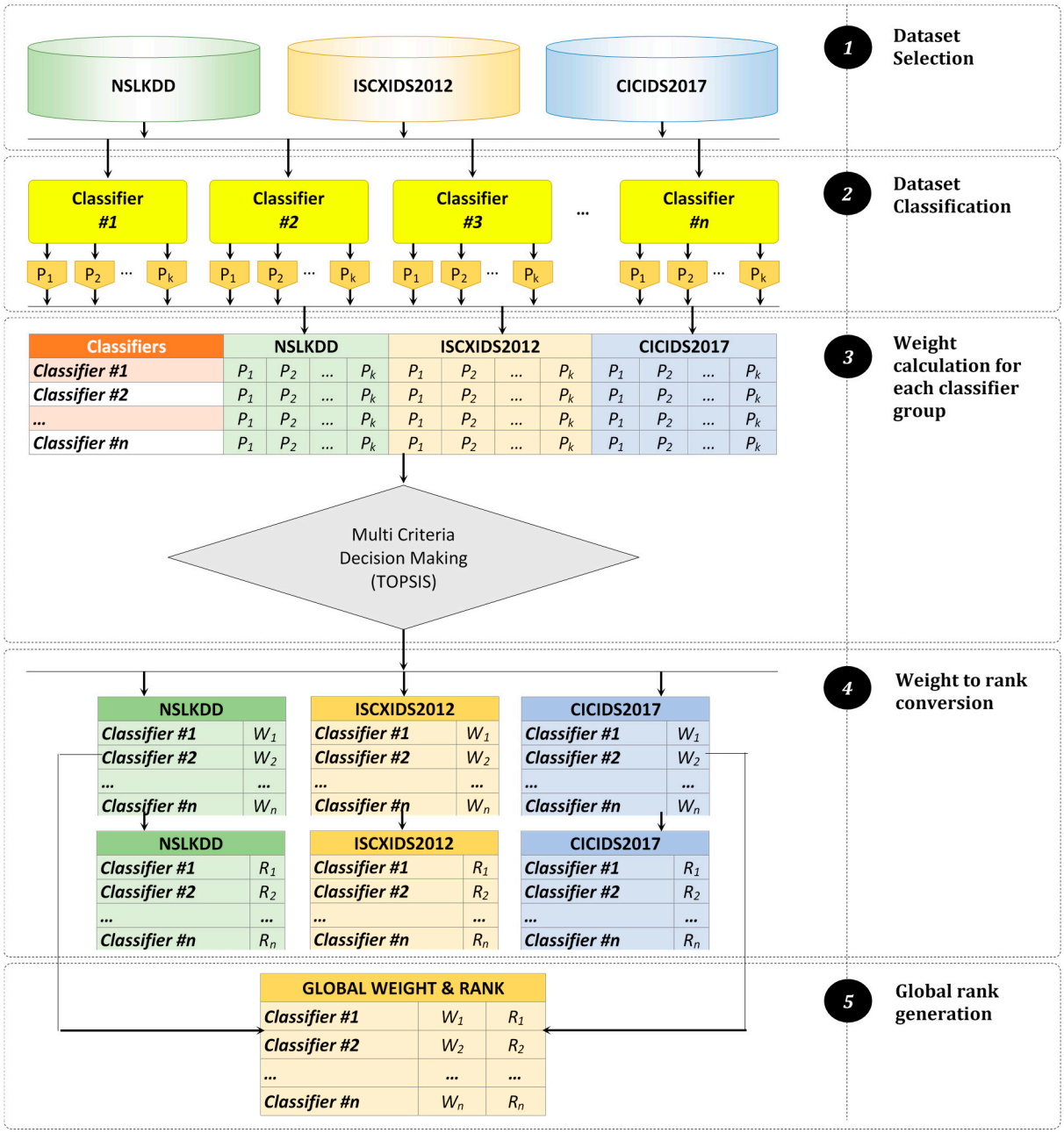

Figure 3. The methodology of classification to rank allocations of supervised classifiers.

The authors have conducted all five steps iteratively for all datasets and classifiers under evaluation. In the initial steps from the pool of datasets, a dataset has been selected. The dataset initially contains several tuples with variable class densities. From each dataset, the requisite number of random samples were generated. The output of this step has been presented in Table 8. This procedure was conducted deliberately to ensure that all the classifiers were not biased for a specific dataset. The second step began by classifying each dataset using each classifier that is presented in the classifier pool. The performance of each classifier was tabulated for future reference. The process has been recursively conducted for each dataset.

The third and fourth steps jointly work to achieve the research objectives. In this process, the average performance score of each group of classifiers has been analyzed. Additionally, each group's ranking has also been calculated to retrieve the best classifier group specific to the dataset. All the group's classifiers with better results were considered to evaluate their consistent performance across the three datasets. Furthermore, considering the performances of the best performing group's classifiers, the authors have calculated the weight and rank of each classifier of that group, specific to each dataset. The authors aimed to provide a reliable evaluation of the best classifier for each dataset.

The final step involved global weight and rank calculation. At this stage, the global weight of a classifier of the best performing group was calculated based on the ranking received for each dataset. The average performance results of those included in the group with the better score across the three datasets were based on the individual score of each classifier. The scores were further arranged in ascending order to provide a clear presentation about the best performance classifier.

All the five steps of methodologies included a two-stage procedure. First, the best classifier group was selected, and the second-best classifier was proposed. The best classifier and classifier group were based on an extensively used conventional multiple-criteria decision-making (MCDM) method named TOPSIS. Before applying TOPSIS, the performance outcome of each classifier and each classifier group were calculated. Therefore, the authors have calculated 13 performance metrics of the classifiers.

Furthermore, the authors considered only eight performance measures, i.e., testing time per instance, accuracy, kappa value, mean absolute error, false-positive rate, precision, and receiver operating curve value for weighting and ranking purpose. On the one hand, these eight measures are in line with the aim of this research. On the other hand, all the other performance metrics can be calculated through one of these measures that are considered in this study. Consequently, the significance of those 17 measures did not affect the weighting and ranking process. The algorithmic method of the weighting of each classifier and classifier group based on TOPSIS has been demonstrated in Table 9.

It should be noted that in algoWeighting, C_1, C_2, C_3, ... , C_n are the classifier or classifier group labels, and P_1, P_2, P_3, ... , P_k are the performance or average performance score, respectively.

The algorithm begins with constructing a decision matrix M_d, where the nth classifier or classifier group is the performance outcome for kth performance measure. The decision matrix is the basis for the evaluation of the best classifier. It helps the decision-making module (TOPSIS) to calculate the weight for each feature.

At the second stage, a weightage normalized decision matrix has been calculated, which is the weight of the jth performance measures.

The idea behind allocating appropriate weight to performance measures is in its ability to rank classifiers specific to domain area and learning environment. For instance, in high class-imbalance learning, the performance measure Matthews correlation coefficient (MCC), Kappa, and receiver operating curve (ROC) value should be given more weightage than other performance matrices. The datasets used here were class imbalance in nature; therefore, more emphasis has been given to performance matrices suitable for the class imbalance environment. In this regard, eight performance matrices have been shortlisted, and corresponding weights have been allocated for TOPSIS processing. The weight for eight performance measures is presented in Table 10. Another reason for not considering all the performance matrices is because other performance measures themselves can be derived from the matrices presented in Table 10. For instance, detection accuracy can be calculated from True Positives (TP) and True Negatives (TN). Therefore, the True Positive Rate (TPR) and True Negative Rate (TNR) have been dropped from calculating weight for classifiers. In this way, out of the 13 performance measures, only eight performance measures have been selected.

Table 9. The algorithm algoWeighting.

Input:

$C := \{C_1, C_2, C_3, \ldots\ldots\ldots, C_n\}$ //*Classifiers or classifiers groups*

$P := \{P_1, P_2, P_3, \ldots\ldots\ldots, P_k\}$ //*Performance measures*

Output:

Classifiers group with weights W_i.

begin

Step 1. Decision matrix construction

$$M_d := \begin{bmatrix} C_1P_1 & C_1P_2 & \cdots & C_1P_k \\ C_2P_1 & C_2P_2 & \cdots & C_2P_k \\ \cdots & \cdots & \cdots & \cdots \\ C_nP_1 & C_nP_2 & \cdots & C_nP_k \end{bmatrix}. \quad \text{// } n \text{ = clasfiers and } k \text{ = performance outcomes}$$

Step 2. Decision matrix normalization

for i: = 0 **to** n.

begin

for j: = 0 **to** k.

begin

$r_{ij} := \frac{x_{ij} - min(x_j)}{max(x_j) - min(x_j)}$.

end

end

Step 3. Formation of weighted normalized matrix

$V_{ij} W_j r_{ij}$. // W_j= *weight allocated for performance matric j*

Step 4. Estimation of positive (A^+) and negative (A^-) ideal solution

$A^+ := \{V_1^+, \ldots, V_n^+\}$, $A' := \{V_1^-, \ldots, V_n^-\}$.

Step 5. Estimation of separation point of each classifier/classifier group

$S_i^+ \sqrt{\sum_{j=1}^{n} \left(V_{ij} - V_j^+\right)^2}$, //*positive ideal solutions*

$S_i^- \sqrt{\sum_{j=1}^{n} \left(V_{ij} - V_j^-\right)^2}$, //*negative ideal solutions*

Step 6. Weight estimation of classifiers

$W_i := \frac{S_i^-}{(S_i^- + S_i^+)}$.

end

Table 10. Weights allocated to various performance measures.

Performance Measures	Weight Allocated
Testing Time	1
Accuracy	8
Kappa	4
Mean Absolute Error (MAE)	2
False Positive Rate (FPR)	5
Precision	7
Receiver Operating Curve (ROC) Value	6
Matthews Correlation Coefficient (MCC)	3

The algorithm includes a positive and negative ideal solution to calculate the separation measure of each classifier/classifier group, which supports the calculation of each classifier or group's score. The scores are used to rank the classifiers. The procedure followed here for calculating the rank of classifiers has been presented in Table 11.

Table 11. The algorithm rankClassifiers.

Input:
$C := \{C_1, C_2, C_3, \ldots\ldots\ldots, C_p\ \}$. *//Classifiers or classifiers groups*
$W^d :=$ *//Classifiers' weight for dataset d*
$\{W_1, W_2, W_3, \ldots\ldots\ldots,\ W_q\}$.

Output:
Classifiers/Classifier group labels with rank R

begin
Step 1. Import list of classifiers
$C :=$
$\{C_1, C_2, C_3, \ldots\ldots\ldots,\ C_n\}$
Step 2. Import classifiers weights
$W :=$
$\{W_1, W_2, W_3, \ldots\ldots\ldots,\ W_n\}$
Step 3. Calculate average weight of classifiers for each dataset
$W_c^d \frac{\sum_{i=0}^{d} C_p W_q^i}{|d|}$
Step 4. Rank classifiers based on descending order of their weight
$R_c Rank^{desc}\left(W_c^d\right)$
end

4. Results and Discussion

The presented analysis to reach the best classifier was conducted through a top-to-bottom approach. Firstly, the best classifier group has been identified through intergroup analysis. Secondly, the best performing classifier of that best classifier group has been acknowledged through intragroup analysis.

4.1. Intergroup Performance Analysis

Under intergroup performance analysis, the authors have calculated the classifier group performance as a whole. The classifier's group performances for NSLKDD, ISCX-IDS2012, and CICIDS2017 datasets have been listed in Tables 12–14, respectively.

According to Table 12, decision tree classifiers present reliable results in all the fields of performance metrics, except training and testing time. On the one hand, the decision tree classifiers consume training and testing times of 4.18 s and 0.03 s, respectively. Similarly, the Bayes group of classifiers has a fast response in training and testing time but presents low-quality performance metrics. The ROC and MCC values are suitable for evaluating classifier groups' performance, considering the class imbalance learning. Therefore, by observing the average ROC and MCC of classifier groups on the NSL-KDD dataset, the authors have seen that the decision tree behaves far better than other classifier groups. The authors found a similar observation concerning the ISCXIDS2012 dataset. Table 6 shows the group performance of supervised classifiers for the ISCXIDS2012 dataset. The decision tree classifiers showed the highest amount of average accuracy of 97.3519%, but the average testing time per instance was low and on par with Bayes and Miscellaneous classifiers. Nevertheless, decision tree classifiers were far ahead of their peer classifier groups, with a higher average ROC value of 0.985. The authors have also conducted intergroup performance analysis on CICIDS2017. The average, maximum, and minimum performance reading has been outlined in Table 12. The decision tree classifiers reveal an impressive amount of accuracy and ROC values of 99.635 and 0.999, respectively.

Furthermore, the decision tree classifiers present consistent performance metrics for all three intrusion detection datasets NSLKDD, ISCXIDS2012, and CICIDS2017. However, before concluding that decision trees are best for these datasets by considering a limited number of parameters, the authors have decided to identify all these classifier groups' actual weight and rank through TOPSIS. The classifier group with the highest weight and rank will be pointed out as the best classifier for these IDS datasets. This will improve the proposed study's relevance and background to find the best classifier within the winning classifier group.

Table 12. Overall performance of classifier groups for NSLKDD dataset.

Miscellaneous	**Avg**	1.410	0.020	49.698	50.302	0.372	0.245	0.393	80.759	100.762	0.955	0.045	95.464	95.464	95.464	0.745	0.437	0.552
	Max	5.550	0.040	84.392	75.246	0.793	0.317	0.549	104.430	140.701	0.993	0.078	99.288	99.288	99.288	0.896	0.801	0.751
	Min	0.010	0.010	24.754	15.609	0.071	0.062	0.250	20.571	64.082	0.922	0.007	92.167	92.167	92.167	0.538	0.180	0.285
Decision Tree	**Avg**	4.180	0.030	95.460	4.540	0.940	0.027	0.121	8.826	31.035	0.996	0.004	99.561	99.561	99.561	0.988	0.943	0.963
	Max	39.970	0.130	97.619	13.568	0.969	0.094	0.200	30.903	51.334	0.999	0.020	99.938	99.938	99.938	0.998	0.970	0.993
	Min	0.020	0.001	86.432	2.381	0.823	0.013	0.090	4.160	23.006	0.980	0.001	98.002	98.002	98.002	0.971	0.841	0.886
Rules	**Avg**	1.590	0.040	82.121	17.859	0.763	0.081	0.200	26.846	51.314	0.988	0.012	98.754	98.754	98.754	0.898	0.767	0.799
	Max	7.620	0.200	97.241	74.339	0.964	0.304	0.433	100.000	111.128	0.999	0.100	99.935	99.935	99.935	0.993	0.965	0.976
	Min	0.001	0.001	25.661	2.759	0.000	0.012	0.102	3.807	26.088	0.900	0.001	89.991	89.991	89.991	0.500	0.000	0.240
Lazy	**Avg**	48.960	15.560	90.730	9.270	0.876	0.050	0.165	16.474	42.355	0.996	0.004	99.601	99.601	99.601	0.969	0.876	0.919
	Max	333.500	67.290	95.729	34.467	0.944	0.209	0.313	68.821	80.239	0.999	0.012	99.880	99.880	99.880	0.991	0.945	0.971
	Min	0.001	0.140	65.533	4.271	0.534	0.020	0.122	6.724	31.167	0.988	0.001	98.775	98.775	98.775	0.927	0.525	0.825
Functions	**Avg**	2.990	0.120	72.061	27.939	0.629	0.155	0.292	50.928	74.868	0.991	0.009	99.130	99.130	99.130	0.887	0.639	0.737
	Max	9.370	1.220	92.026	38.813	0.895	0.262	0.371	86.342	95.227	0.997	0.016	99.675	99.675	99.675	0.946	0.897	0.866
	Min	0.020	0.001	61.187	7.974	0.501	0.032	0.179	10.509	45.804	0.984	0.003	98.400	98.400	98.400	0.770	0.520	0.510
Bayes	**Avg**	0.040	0.010	41.043	58.957	0.266	0.258	0.369	84.979	94.696	0.966	0.034	96.596	96.596	96.596	0.694	0.282	0.479
	Max	0.080	0.020	70.824	81.519	0.610	0.322	0.405	106.223	103.920	0.987	0.064	98.664	98.664	98.664	0.889	0.628	0.745
	Min	0.010	0.001	18.481	29.176	0.000	0.176	0.309	57.835	79.140	0.936	0.013	93.591	93.591	93.591	0.500	0.000	0.240
Performance Measures		Training Time (s)	Testing Time (s)	Model Accuracy (%)	M.C.R. (%)	Kappa Statistics	M.A.E. r	R.M.S.E.	R.A.E. (%)	R.R.S.E. (%)	True Positive Rate	False Positive Rate	Precision (%)	Sensitivity (%)	F-Measure	ROC Value	MCC Value	PRC Area

Table 13. Overall performance of classifier groups for ISCXIDS2012 dataset.

Miscellaneous	**Avg**	0.740	0.010	57.548	42.452	0.145	0.428	0.543	85.557	108.576	0.995	0.005	99.540	99.540	99.540	0.590	0.176	0.573
	Max	2.940	0.030	77.356	49.090	0.545	0.499	0.701	99.813	140.098	0.997	0.006	99.699	99.699	99.699	0.771	0.570	0.717
	Min	0.001	0.001	50.910	22.645	0.000	0.226	0.476	45.281	95.152	0.994	0.003	99.406	99.406	99.406	0.500	0.000	0.500
Decision Tree	**Avg**	5.170	0.010	97.352	2.648	0.947	0.036	0.152	7.197	30.338	1.000	0.000	99.973	99.973	99.973	0.985	0.947	0.980
	Max	60.480	0.040	98.555	5.300	0.971	0.081	0.213	16.135	42.649	1.000	0.000	99.987	99.987	99.987	0.998	0.971	0.998
	Min	0.020	0.001	94.700	1.445	0.894	0.021	0.107	4.175	21.384	1.000	0.000	99.951	99.951	99.951	0.968	0.895	0.954
Rules	**Avg**	0.610	0.020	89.960	10.031	0.800	0.114	0.243	22.758	48.564	0.999	0.001	99.907	99.907	99.907	0.905	0.808	0.890
	Max	3.430	0.160	97.912	50.910	0.958	0.500	0.529	100.000	105.702	1.000	0.004	99.982	99.982	99.982	0.992	0.959	0.991
	Min	0.001	0.001	49.090	2.088	0.000	0.023	0.139	4.670	27.863	0.996	0.000	99.605	99.605	99.605	0.500	0.000	0.500
Lazy	**Avg**	14.070	9.220	92.551	7.449	0.851	0.089	0.252	17.747	50.293	0.999	0.001	99.923	99.923	99.923	0.940	0.855	0.920
	Max	92.180	29.720	97.323	17.827	0.946	0.273	0.367	54.614	73.282	1.000	0.002	99.972	99.972	99.972	0.990	0.946	0.987
	Min	0.001	0.010	82.173	2.677	0.641	0.030	0.153	5.995	30.560	0.998	0.000	99.825	99.825	99.825	0.884	0.674	0.866
Functions	**Avg**	2.340	0.170	70.873	29.127	0.413	0.343	0.471	68.686	94.124	0.997	0.003	99.730	99.730	99.730	0.739	0.451	0.731
	Max	18.720	1.860	90.364	49.090	0.807	0.491	0.701	98.163	140.098	0.999	0.005	99.906	99.906	99.906	0.929	0.807	0.924
	Min	0.010	0.001	50.910	9.636	0.000	0.170	0.302	33.986	60.396	0.995	0.001	99.498	99.498	99.498	0.500	0.000	0.500
Bayes	**Avg**	0.020	0.010	50.669	49.331	0.004	0.498	0.552	99.558	110.331	0.995	0.005	99.486	99.486	99.486	0.576	0.004	0.563
	Max	0.050	0.020	50.910	49.786	0.021	0.500	0.702	99.983	140.281	0.996	0.006	99.610	99.610	99.610	0.791	0.058	0.746
	Min	0.001	0.001	50.214	49.090	−0.005	0.493	0.500	98.603	99.969	0.994	0.004	99.373	99.373	99.373	0.500	−0.043	0.500
Performance Measures		Training Time (s)	Testing Time (s)	Model Accuracy (%)	M.C.R. (%)	Kappa Statistics	M.A.E.	R.M.S.E.	R.A.E. (%)	R.R.S.E. (%)	True Positive Rate	False Positive Rate	Precision (%)	Sensitivity (%)	F-Measure	ROC Value	MCC Value	PRC Area

Table 14. Overall performance of classifier groups for CICIDS2017 dataset.

Miscellaneous	**Avg**	0.750	0.020	98.961	1.039	0.987	0.079	0.141	33.502	41.113	1.000	0.000	99.989	99.989	99.989	0.996	0.988	0.987
	Max	2.900	0.030	99.835	1.847	0.998	0.225	0.323	95.108	93.957	1.000	0.000	99.998	99.998	99.998	1.000	0.998	0.999
	Min	0.010	0.010	98.153	0.165	0.978	0.001	0.022	0.200	6.315	1.000	0.000	99.979	99.979	99.979	0.989	0.978	0.968
Decision Tree	**Avg**	19.150	0.040	99.635	0.365	0.996	0.002	0.030	0.856	8.847	1.000	0.000	99.996	99.996	99.996	0.999	0.996	0.997
	Max	258.830	0.180	99.868	0.693	0.998	0.005	0.044	1.888	12.889	1.000	0.000	99.999	99.999	99.999	1.000	0.998	1.000
	Min	0.030	0.000	99.307	0.132	0.992	0.000	0.019	0.160	5.648	1.000	0.000	99.993	99.993	99.993	0.997	0.992	0.990
Rules	**Avg**	1.490	0.020	86.528	13.472	0.835	0.040	0.097	17.109	28.123	0.999	0.001	99.874	99.874	99.874	0.931	0.836	0.857
	Max	8.790	0.050	99.868	81.300	0.998	0.236	0.344	100.000	100.000	1.000	0.007	99.999	99.999	99.999	1.000	0.998	1.000
	Min	0.000	0.000	18.701	0.132	0.000	0.001	0.020	0.200	5.861	0.993	0.000	99.258	99.258	99.258	0.500	0.000	0.173
Lazy	**Avg**	24.600	22.380	94.973	5.027	0.938	0.022	0.064	9.513	18.547	1.000	0.000	99.953	99.953	99.953	0.998	0.938	0.993
	Max	158.190	74.390	99.802	31.860	0.998	0.148	0.254	62.516	73.749	1.000	0.003	99.998	99.998	99.998	1.000	0.998	0.999
	Min	0.000	0.030	68.140	0.198	0.609	0.001	0.024	0.239	6.918	0.997	0.000	99.703	99.703	99.703	0.991	0.606	0.982
Functions	**Avg**	18.420	0.430	86.702	13.298	0.837	0.065	0.166	27.680	48.178	0.999	0.001	99.876	99.876	99.876	0.933	0.843	0.871
	Max	115.950	4.470	99.373	73.450	0.992	0.210	0.458	88.857	133.270	1.000	0.007	99.995	99.995	99.995	0.999	0.992	0.998
	Min	0.030	0.010	26.550	0.627	0.097	0.002	0.041	0.758	12.018	0.993	0.000	99.295	99.295	99.295	0.548	0.232	0.241
Bayes	**Avg**	0.030	0.020	43.041	56.959	0.347	0.172	0.265	72.822	77.112	0.994	0.006	99.440	99.440	99.440	0.711	0.345	0.472
	Max	0.070	0.070	98.318	89.116	0.980	0.246	0.353	104.155	102.627	1.000	0.010	99.985	99.985	99.985	0.999	0.979	0.996
	Min	0.010	0.001	10.884	1.682	0.001	0.005	0.063	2.004	18.251	0.990	0.000	99.024	99.024	99.024	0.500	0.000	0.173
Performance Measures		Training Time (s)	Testing Time (s)	Model Accuracy (%)	M.C.R. (%)	Kappa Statistics	M.A.E.	R.M.S.E.	R.A.E. (%)	R.R.S.E. (%)	True Positive Rate	False Positive Rate	Precision (%)	Sensitivity (%)	F-Measure	ROC Value	MCC Value	PRC Area

Figure 4 presents the weights and ranks of classifier groups for all three IDS datasets. The decision tree classifier presents the highest performance. Moreover, the decision trees present a consistent performance for all the IDS datasets. Therefore, the decision tree can be considered as the best method for the development of reliable IDSs.

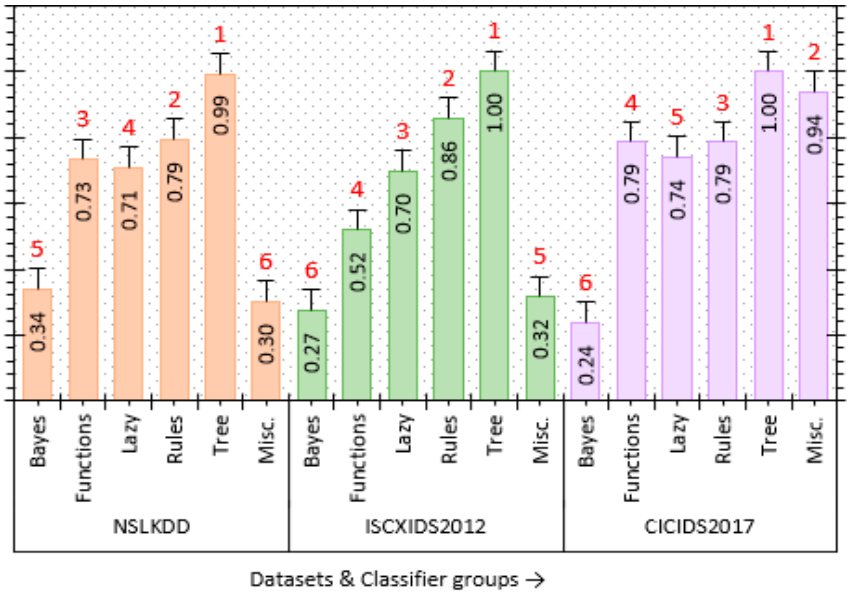

Figure 4. Weights and ranks of supervised classifier groups.

4.2. Intragroup Performance Analysis

In the intergroup analysis, the authors conclude that decision tree classifiers reveal the best performance for imbalanced IDS datasets. The authors have decided to conduct an intragroup analysis of decision trees for NSLKDD, ISCXIDS2012, and CICIDS2017 datasets. The intragroup study aims to identify the best decision tree within the decision tree group of classifiers for the concerned datasets. Several performance outcomes of decision tree classifiers for NSLKDD, ISCXIDS2012, and CICIDS2017 datasets have been analyzed through Figures 5–7.

Figure 5. Performance of decision tree classifiers for NSLKDD dataset.

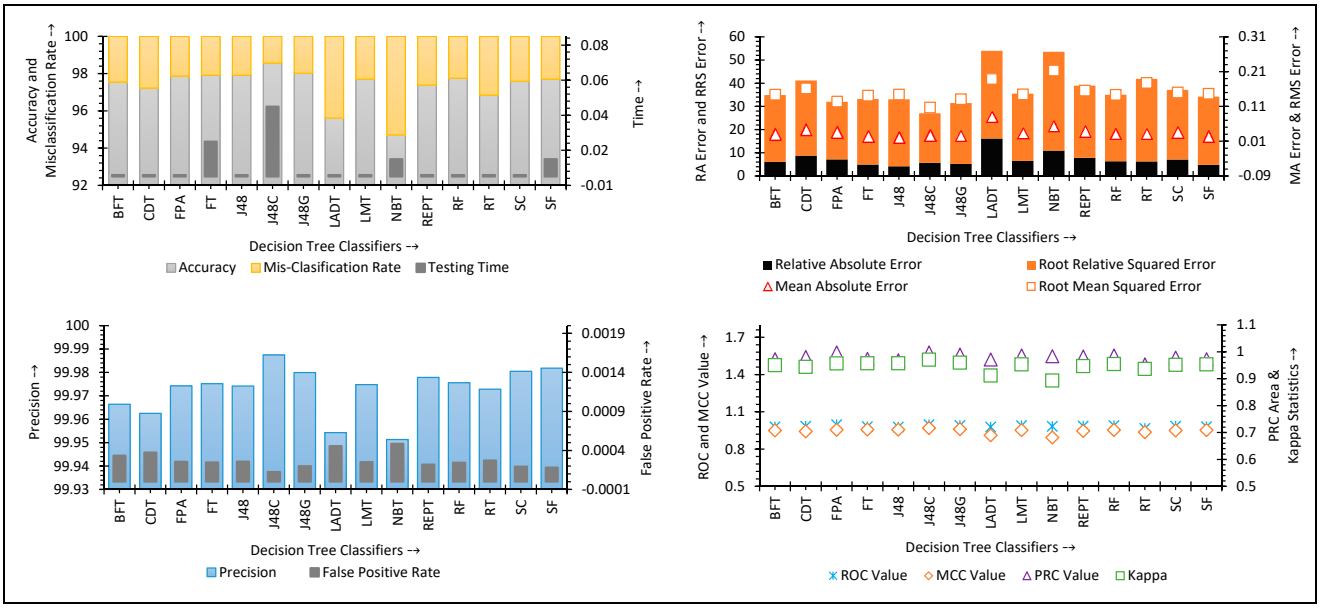

Figure 6. Performance of decision tree classifiers for ISCXIDS2012 dataset.

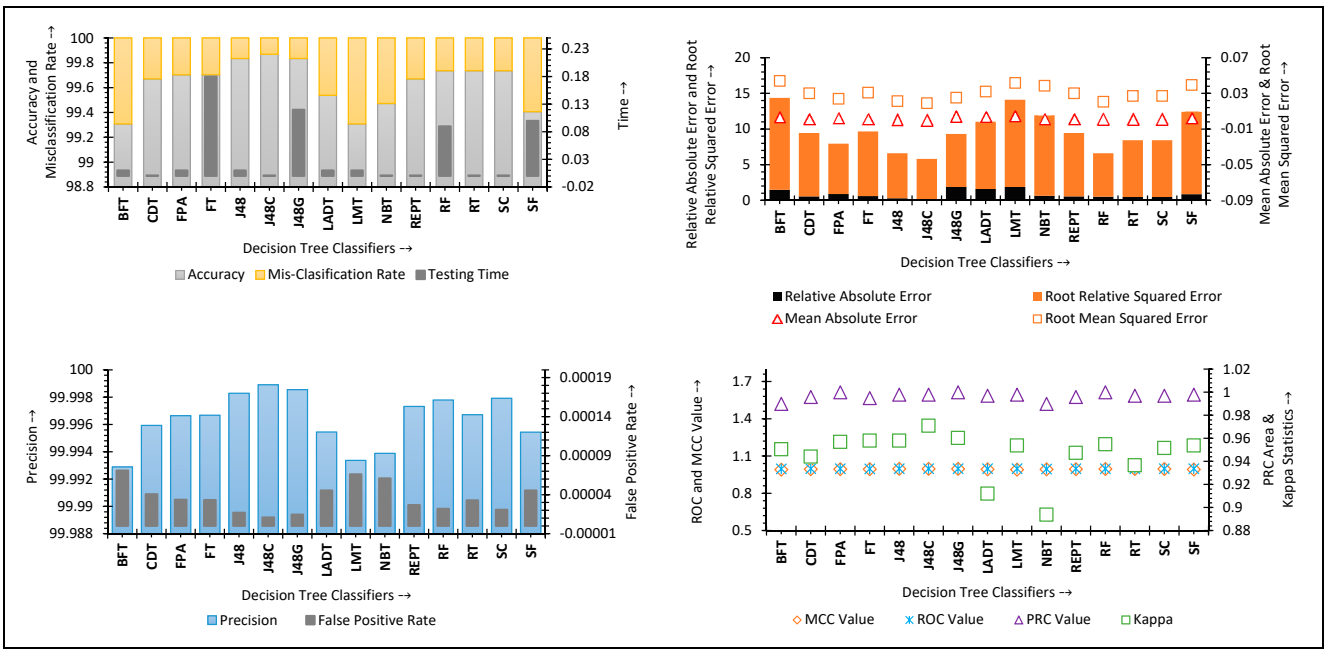

Figure 7. Performance of decision tree classifiers for CICIDS2017 dataset.

The J48Consolidated classifier shows better accuracy for the NSL-KDD dataset. The sample size of NSLKDD here is an imbalance in nature. Therefore, these measures play a significant role in finding the best classifier. Considering the ROC value, the ForestPA performs better as compared to J48Consolidated. Additionally, both ForestPA and J48Consolidated show similar performance in terms of the MCC value. Consequently, the authors did not find sufficient scope for deciding an ideal decision tree classifier for the NSLKDD dataset.

Furthermore, the decision tree classifiers' performance on a sample of the ISCXIDS2012 dataset is presented in Figure 6. The Functional Trees (FT), J48Consolidated, NBTree, and SysFor classifiers consumed a significant amount of computational time. Nevertheless, the rest of the decision trees consumed 0.001 s of testing time per instance. The J48Consolidated algorithm was limited by presenting the longest amount of time to detect an anomalous instance. However, this computation time consumption supports

the fact that J48Consolidated provides the highest accuracy of 98.5546%, which leads to the lowest misclassification rate of 1.4454%. Moreover, J48Consolidated seems to lead the decision trees group with the best Kappa value (0.9711).

The test results of decision trees on a CICIDS2017 dataset are presented in Figure 7. The J48Consolidated algorithm provides high-quality results in the class imbalance instances of the CICIDS2017 dataset. J48Consolidated scores the highest accuracy with a low misclassification rate. However, considering the ROC and MCC values, the J48 presents better performance than the J48Consolidated. Therefore, it is not clear about the best classifiers, which can be considered as the base learner for future IDS.

In the case of ISCXIDS2012, J48Consolidated also presents consistent results in all performance measures. However, in the case of NSL-KDD and CICIDS2017, it was not possible to find the best classifier. Therefore, the authors have also considered TOPSIS to allocate individual decision tree classifiers' weight and rank. The average weight and rank of decision tree classifiers for all datasets have also been calculated to find the best classifier for all the datasets. The average weight and rank across all the datasets are not significant in identifying a suitable classifier because an IDS is designed considering a specific dataset or environment. However, average weight and rank will play a relevant role in the conclusion concerning the most versatile classifier conducted in this study. The average ranks and weights of all the classifiers for all the three IDS datasets are represented in Figure 8.

The J48Consolidated classifier has the highest rank across all the datasets. Moreover, J48Consolidated presents the highest weight of 0.964 for the ISCXIDS2012 dataset. The J48Consolidated decision tree classifier is best for the high-class imbalance NSLKDD and CICIDS2017 and ISCXIDS2012 datasets. Therefore, J48Consolidated will be a suitable classifier for designing IDS base learners using either NSLKDD, ISCXIDS2012, and CICIDS2017 datasets.

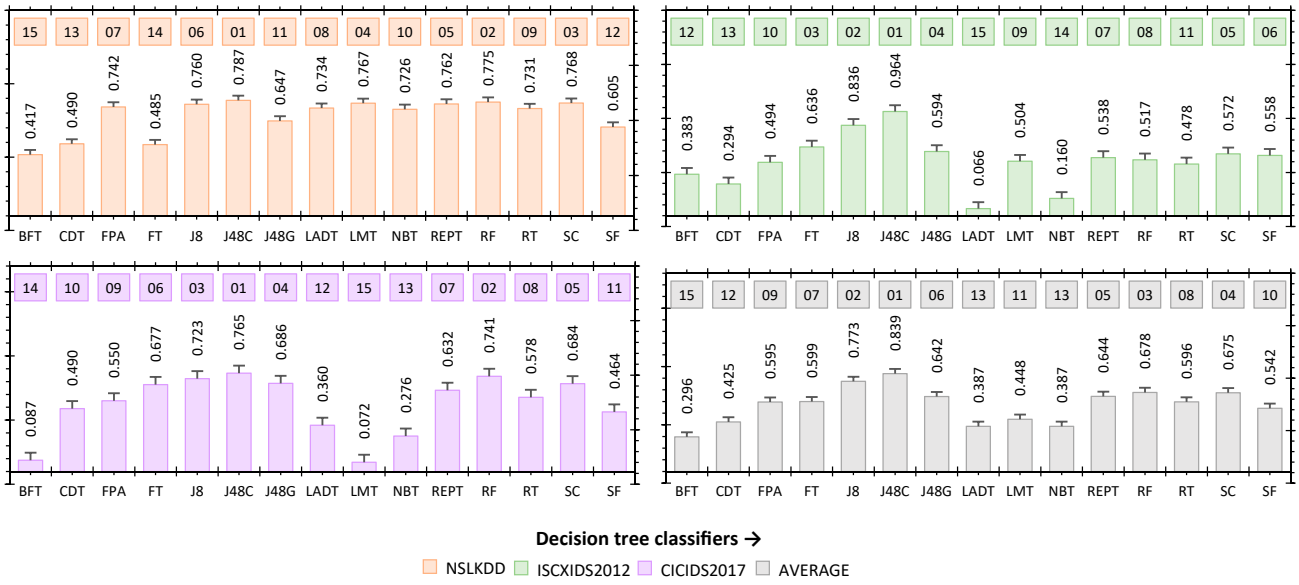

Figure 8. Techniques for Order Preference by Similarity to the Ideal Solution (TOPSIS) weights and ranks of decision tree classifiers for NSLKDD, ISCXIDS2012 and CICIIDS2017 dataset.

4.3. Detailed Performance Reading of All the Classifiers

Tables 15–17 provide a detailed insight of all the supervised classifiers in six distinct groups. These tables outlined thirteen performance metrics. However, the authors have identified the best classifier group (decision tree) and the best classifier (J48Consolidated). Nevertheless, other classifiers can have different performances considering other datasets. Therefore, while designing IDSs, the authors suggest further evaluation of supervised classifiers based on specific computing and network environments.

Table 15. Performance outcome of supervised classifiers on NSL-KDD dataset.

PRC		0.69	0.24	0.75	0.24		0.75	0.51	0.87	0.84	0.69	0.84	0.73	0.71	0.76	0.84	0.58		0.91	0.94	0.97	0.97	0.9	0.83	0.91		0.43	0.97	0.98	0.97
MCC		0.63	0	0.5	0		0.54	0.52	0.9	0.72	0.57	0.75	0.53	0.57	0.65	0.71	0.56		0.93	0.94	0.94	0.95	0.93	0.53	0.93		0.28	0.93	0.93	0.95
ROC		0.89	0.5	0.89	0.5		0.89	0.77	0.95	0.94	0.87	0.93	0.88	0.88	0.91	0.94	0.83		0.97	0.98	0.99	0.99	0.96	0.93	0.97		0.78	0.99	0.99	0.99
PRE		95.82 31889	93.59 08483	98.66 36249	98.30 7638		98.39 9922	98.55 60253	99.67 48494	99.19 72477	98.89 33377	99.37 02418	98.97 77229	99.20 17755	99.35 70459	99.52 77488	99.27 14257		99.63 10669	99.6 6243	99.68 28219	99.79 18892	99.78 39184	98.77 48687	99.88 02779		89.99 10501	99.01 83819	99.28 2802	99.64 97204
FPR		0.041 76811	0.064 09152	0.013 36375	0.016 92362		0.016 00078	0.014 43975	0.003 25151	0.008 02752	0.011 06662	0.006 29758	0.010 22277	0.007 98225	0.006 42954	0.004 72251	0.007 28574		0.003 68933	0.00 33757	0.003 17178	0.002 08111	0.002 16082	0.012 25131	0.001 19722		0.10 00895	0.009 81618	0.007 17198	0.00 35028
RRSE		79.1399	102.586	93.1367	103.92		81.8842	95.2269	45.804	66.1501	75.5902	62.0564	91.8191	76.9766	72.4047	66.2061	89.427		36.6378	36.0546	36.0659	31.1672	38.7481	80.2386	37.5756		87.5368	39.2184	36.9341	28.0692
RAE		70.4239	105.433	57.8345	106.223		61.3275	45.4247	10.5094	47.2252	59.8444	29.678	56.3749	62.3526	53.7548	47.3792	86.3418		6.7241	7.1712	7.1279	10.8788	7.521	68.8212	7.0727		76.422	22.9807	19.5085	5.2622
RMSE		0.309	0.4	0.363	0.405		0.319	0.371	0.179	0.258	0.295	0.242	0.358	0.3	0.282	0.258	0.349		0.143	0.141	0.141	0.122	0.151	0.313	0.147		0.341	0.153	0.144	0.109
MAE		0.214	0.32	0.176	0.322		0.186	0.138	0.032	0.143	0.182	0.09	0.171	0.189	0.163	0.144	0.262		0.02	0.022	0.022	0.033	0.023	0.209	0.022		0.232	0.07	0.059	0.016
KV		0.61	0	0.452	0		0.502	0.536	0.895	0.708	0.571	0.743	0.501	0.564	0.649	0.706	0.543		0.933	0.934	0.934	0.944	0.925	0.534	0.929		0.316	0.922	0.927	0.946
MCR		29.176	81.519	43.613	81.519		37.188	34.467	7.9743	21.958	31.859	19.539	38.813	32.351	26.493	22.071	34.618		5.102	4.9887	4.9887	4.2706	5.7067	34.467	5.3666		50.718	5.9335	5.5178	4.1194
ACC		70.824	18.481	56.387	18.481		62.812	65.533	92.026	78.042	68.141	80.461	61.187	67.649	73.507	77.929	65.382		94.898	95.011	95.011	95.729	94.293	65.533	94.633		49.282	94.067	94.482	95.881
TT		0.001	0.01	0.02	0.01		0.02	0.01	1.22	0.01	0.01	0.01	0.01	0.01	0.02	0.001	0.01		0.9	0.56	0.53	27.26	67.29	12.27	0.14		0.001	0.01	0.01	0.02
Name of Classifiers	**Bayes Group**	DMNB	HMM	NB	SGM	**Function-based**	LDA	LLNR	LSVM	LR	MLP	MLP	QDA	RBF	RBFN	SLR	SMO	**Lazy Groups**	IB1	IBK	IBKLG	KSTAR	LKNN	LWL	RLKNN	**Rule-based**	CR	DTBL	DTNB	FURIA
PRC	0.96	0.91	0.91	0.48	0.69	0.97	0.93	0.95	0.24		0.97	0.97	0.99	0.94	0.96	0.99	0.97	0.89	0.96	0.98	0.97	0.97	0.95	0.97	0.97		0.75	0.29	0.44	0.73
MCC	0.96	0.94	0.93	0.46	0.74	0.97	0.95	0.96	0		0.95	0.94	0.95	0.93	0.97	0.97	0.96	0.84	0.96	0.96	0.95	0.95	0.96	0.92	0.94		0.8	0.18	0.18	0.59
ROC	0.99	0.97	0.97	0.71	0.86	0.99	0.97	0.98	0.5		0.99	0.99	1	0.98	0.99	1	0.99	0.97	0.99	0.99	0.99	0.99	0.98	0.99	0.99		0.9	0.54	0.65	0.9
PRE	99.82 79894	99.83 07215	99.841 0367	98.674 7335	99.41 443	99.924 9049	99.904 8517	99.934 5042	98.511 9678		98.001 9589	98.47 03883	99.63 1803	99.47 24522	99.93 12084	99.93 82843	99.81 64556	99.39 36835	99.86 18703	99.89 96154	99.89 90746	99.92 15602	99.62 81227	99.64 02261	99.90 69906		95.94 22283	92.16 69683	94.45 9749	99.287 9644
FPR	0.001 72011	0.001 69279	0.0015 8963	0.0132 5267	0.005 8557	0.000 75095	0.000 95148	0.000 65496	0.0148 8032		0.0199 8041	0.0152 9612	0.0036 8197	0.0052 7548	0.000 68792	0.000 61716	0.0018 3544	0.0060 6317	0.001 3813	0.001 00385	0.001 00926	0.000 7844	0.003 71877	0.003 59774	0.000 93009		0.040 57772	0.0783 3032	0.055 40251	0.0071 2036
RRSE	29.1536	36.2284	37.9705	111.128	74.42	26.0876	32.7698	27.5715	100		29.3488	31.0348	26.7965	35.1255	26.726	23.0061	27.3993	51.3341	29.6896	28.125	31.0852	28.9906	29.0771	37.2188	30.567		64.0822	140.701	101.664	96.6
RAE	7.1175	6.5746	7.2221	61.8613	27.7429	5.1261	5.3792	3.8067	100		6.1856	8.6064	7.66	9.4388	4.16	5.3558	5.6998	30.9031	5.9875	9.7965	8.5333	6.4623	4.838	12.3021	6.4675		20.5706	99.1673	104.43	98.8697
RMSE	0.114	0.141	0.148	0.433	0.29	0.102	0.128	0.107	0.39		0.114	0.121	0.105	0.137	0.104	0.09	0.107	0.2	0.116	0.11	0.121	0.113	0.113	0.145	0.119		0.25	0.549	0.396	0.377
MAE	0.022	0.02	0.022	0.188	0.084	0.016	0.016	0.012	0.304		0.019	0.026	0.023	0.029	0.013	0.016	0.017	0.094	0.018	0.03	0.026	0.02	0.015	0.037	0.02		0.062	0.301	0.317	0.3
KV	0.956	0.934	0.928	0.401	0.72	0.964	0.946	0.962	0		0.949	0.942	0.952	0.925	0.964	0.969	0.96	0.823	0.954	0.956	0.944	0.953	0.958	0.917	0.941		0.793	0.071	0.071	0.553
MCR	3.3636	4.9887	5.48	46.939	21.051	2.7589	4.0816	2.8723	74.339		3.8549	4.4218	3.6281	5.7067	2.7211	2.381	3.0612	13.568	3.5147	3.3636	4.2328	3.5903	3.2124	6.3492	4.4974		15.609	75.246	75.246	35.11
ACC	96.636	95.011	94.52	53.061	78.949	97.241	95.918	96.863	25.661		96.145	95.578	96.372	94.293	97.279	97.619	96.939	86.432	96.485	96.636	95.767	96.41	96.788	93.651	95.503		84.392	24.754	24.754	64.89
TT	0.01	0.07	0.19	0.01	0.001	0.01	0.001	0.2	0.001		0.001	0.001	0.01	0.13	0.001	0.08	0.001	0.01	0.001	0.05	0.001	0.001	0.001	0.01	0.1		0.04	0.01	0.01	0.01
Name of Classifiers	JRIP	MODLEM	NNGE	OLM	ONER	PART	RIDOR	ROUGHS	ZEROR	**Decision Trees**	BFT	CDT	FPA	FT	J48	J48C	J48G	LADT	LMT	NBT	REPT	RF	RT	SC	SF	**Miscellaneous**	CHIRP	FLR	HP	VFI

Table 16. Performance outcome of supervised classifiers on ISCXIDS2012 dataset.

PRC		0.51	0.5	0.75	0.5		0.88	0.5	0.54	0.89	0.92	0.58	0.88	0.68	0.76	0.89	0.52		0.89	0.89	0.93	0.99	0.96	0.87	0.92		0.78	0.96	0.99	0.98
MCC		0.06	0	−0.04	0		0.69	0	0.25	0.71	0.81	0.18	0.7	0.31	0.54	0.71	0.08		0.85	0.85	0.85	0.95	0.94	0.67	0.89		0.69	0.9	0.94	0.96
ROC		0.51	0.5	0.79	0.5		0.9	0.5	0.56	0.9	0.93	0.54	0.89	0.69	0.78	0.9	0.53		0.92	0.92	0.94	0.99	0.97	0.88	0.94		0.83	0.97	0.99	0.98
PRE		99.37 28674	99.42 99533	99.53 15303	99.60 97789		99.82 43482	99.49 77819	99.55 7592	99.84 30158	99.90 63948	99.5 618	99.85 16314	99.70 34607	99.79 1891	99.87 48053	99.62 05372		99.91 29747	99.91 38788	99.91 47643	99.97 18208	99.96 99616	99.82 54369	99.95 32906		99.77 76012	99.93 58449	99.96 51549	99.97 54771
FPR		0.006 27133	0.005 70047	0.00 46847	0.003 90221		0.001 75652	0.005 02218	0.004 42408	0.001 56984	0.000 93605	0.00 4382	0.001 48369	0.002 96539	0.002 08109	0.001 25195	0.003 79463		0.0008 7025	0.0008 6121	0.00085 236	0.0002 8179	0.0003 0038	0.001 74563	0.0004 6709		0.0022 2399	0.0006 4155	0.0003 4845	0.0002 4523
RRSE		99.9692	99.9786	140.281	101.097		71.0846	140.098	131.834	69.0002	60.3961	96.3492	82.1845	97.0951	80.9554	68.5047	137.865		55.3244	55.3091	55.3226	30.5601	34.6212	73.2824	47.6323		72.0015	41.3656	31.888	27.8625
RAE		99.971	99.983	98.603	99.675		50.171	98.163	86.923	51.026	34.658	92.162	33.986	95.649	64.472	53.276	95.059		15.308	15.355	15.313	6.3002	5.9947	54.614	11.347		52.167	18.041	11.966	4.6704
RMSE		0.5	0.5	0.702	0.506		0.356	0.701	0.659	0.345	0.302	0.482	0.411	0.486	0.405	0.343	0.69		0.277	0.277	0.277	0.153	0.173	0.367	0.238		0.36	0.207	0.16	0.139
MAE		0.5	0.5	0.493	0.499		0.251	0.491	0.435	0.255	0.173	0.461	0.17	0.478	0.322	0.266	0.475		0.077	0.077	0.077	0.032	0.03	0.273	0.057		0.261	0.09	0.06	0.023
KV		0.021	0	−0.005	0		0.658	0	0.116	0.68	0.807	0.066	0.659	0.306	0.506	0.687	0.06		0.847	0.847	0.847	0.946	0.94	0.641	0.887		0.656	0.899	0.944	0.958
MCR		49.786	49.09	49.358	49.09		17.024	49.09	43.469	15.899	9.636	45.932	16.97	34.743	24.518	15.578	47.538		7.6552	7.6552	7.6552	2.6767	2.9979	17.827	5.6745		17.077	5.0321	2.7837	2.0878
ACC		50.214	50.91	50.642	50.91		82.976	50.91	56.531	84.101	90.364	54.069	83.03	65.257	75.482	84.422	52.463		92.345	92.345	92.345	97.323	97.002	82.173	94.326		82.923	94.968	97.216	97.912
TT		0.001	0.001	0.02	0.01		0.01	0.001	1.86	0.001	0.01	0.001	0.001	0.01	0.01	0.001	0.001		0.43	0.42	0.43	25.96	29.72	7.6	0.01		0.001	0.001	0.01	0.001
Name of Classifiers	**Bayes Group**	DMNB	HMM	NB	SGM	**Function-based**	LDA	LLNR	LSVM	LR	MLP	MLP	QDA	RBF	RBFN	SLR	SMO	**Lazy Groups**	IB1	IBK	IBKLG	KSTAR	LKNN	LWL	RLKNN	**Rule-based**	CR	DTBL	DTNB	FURIA
PRC	0.98	0.96	0.95	0.67	0.88	0.98	0.96	0.96	0.5		0.97	0.98	1	0.97	0.97	1	0.99	0.97	0.99	0.98	0.98	0.99	0.95	0.98	0.97		0.72	0.5	0.51	0.56
MCC	0.95	0.94	0.94	0.5	0.84	0.96	0.95	0.94	0		0.95	0.94	0.96	0.96	0.96	0.97	0.96	0.91	0.95	0.9	0.95	0.96	0.94	0.95	0.95		0.57	0	0	0.13
ROC	0.99	0.97	0.97	0.72	0.92	0.99	0.97	0.97	0.5		0.98	0.99	1	0.98	0.98	1	0.99	0.98	0.99	0.98	0.99	0.99	0.97	0.99	0.98		0.77	0.5	0.52	0.57
PRE	99.97 1268	99.97 31938	99.97 12323	99.74 88586	99.92 63852	99.98 15088	99.97 67136	99.97 6457	99.60 49811		99.966 4172	99.962 5352	99.97 42563	99.97 51918	99.97 41804	99.98 7454	99.97 99563	99.95 42549	99.97 4761	99.95 13088	99.97 78768	99.97 56092	99.97 28172	99.98 0487	99.98 18513		99.69 91743	99.40 61919	99.43 8694	99.61 67652
FPR	0.0002 8732	0.0002 6806	0.0002 8768	0.0025 1141	0.0007 3615	0.0001 8491	0.0002 3286	0.0002 3543	0.0039 5019		0.0003 3583	0.0003 7465	0.0002 5744	0.0002 4808	0.000 2582	0.0001 2546	0.0002 0044	0.0004 5745	0.0002 5239	0.0004 8691	0.0002 2123	0.0002 4391	0.0002 7183	0.0001 9513	0.0001 8149		0.0030 0826	0.0059 3808	0.0056 1306	0.0038 3235
RRSE	30.8651	33.3618	35.5365	105.702	57.4128	28.2636	33.3618	33.7169	100		28.84	32.3926	24.852	28.2722	28.8922	21.3836	26.2021	37.8522	29.0201	42.6491	31.1256	28.6718	35.5365	30.0486	29.3342		95.1522	140.098	99.8206	99.2345
RAE	7.8292	5.5665	6.3158	55.879	16.485	5.6777	5.5665	5.6857	100		6.0636	8.7887	7.1535	4.9106	4.1749	5.6436	5.213	16.135	6.501	10.926	7.8099	6.3565	6.3158	7.0733	4.8823		45.281	98.163	99.813	98.97
RMSE	0.154	0.167	0.178	0.529	0.287	0.141	0.167	0.169	0.5		0.144	0.162	0.124	0.141	0.145	0.107	0.131	0.189	0.145	0.213	0.156	0.143	0.178	0.15	0.147		0.476	0.701	0.499	0.496
MAE	0.039	0.028	0.032	0.279	0.082	0.028	0.028	0.028	0.5		0.03	0.044	0.036	0.025	0.021	0.028	0.026	0.081	0.033	0.055	0.039	0.032	0.032	0.035	0.024		0.226	0.491	0.499	0.495
KV	0.946	0.944	0.937	0.436	0.835	0.958	0.944	0.943	0		0.951	0.944	0.957	0.958	0.958	0.971	0.96	0.912	0.954	0.894	0.948	0.955	0.937	0.952	0.954		0.545	0	0	0.037
MCR	2.6767	2.7837	3.1585	27.944	8.2441	2.0878	2.7837	2.8373	50.91		2.4625	2.7837	2.1413	2.0878	2.0878	1.4454	1.9807	4.3897	2.3019	5.2998	2.6231	2.2484	3.1585	2.409	2.3019		22.645	49.09	49.09	48.983
ACC	97.323	97.216	96.842	72.056	91.756	97.912	97.216	97.056	49.09		97.538	97.216	97.859	97.912	97.912	98.555	98.019	95.61	97.698	94.7	97.377	97.752	96.842	97.591	97.698		77.356	50.91	50.91	51.017
TT	0.001	0.01	0.04	0.001	0.001	0.001	0.001	0.16	0.01		0.001	0.001	0.001	0.02	0.001	0.04	0.001	0.001	0.001	0.01	0.001	0.001	0.001	0.001	0.01		0.03	0.001	0.001	0.001
Name of Classifiers	JRIP	MODLEM	NNGE	OLM	ONER	PART	RIDOR	ROUGHS	ZEROR	**Decision Trees**	BFT	CDT	FPA	FT	J48	J48C	J48G	LADT	LMT	NBT	REPT	RF	RT	SC	SF	**Miscellaneous**	CHIRP	FLR	HP	VFI

Table 17. Performance outcome of supervised classifiers on CICIDS2017 dataset.

PRC	0.55	0.17	1	0.17	0.99	0.79	0.24	1	0.7	1	1	0.91	0.98	1	0.98	1	1	1	0.99	1	0.98	1	0.35	1	1	1
MCC	0.4	0	0.98	0	0.93	0.84	0.23	0.99	0.62	0.99	0.99	0.75	0.96	0.99	0.98	1	1	1	0.97	1	0.61	1	0.19	1	1	1
ROC	0.84	0.5	1	0.5	1	0.91	0.55	1	0.85	1	1	0.97	1	1	1	1	1	1	1	1	0.99	1	0.78	1	1	1
PRE	99.43 51798	99.02 42634	99.98 482	99.31 4295	99.94 20017	99.86 27102	99.29 48247	99.99 34914	99.75 22321	99.99 12904	99.99 46994	99.84 89866	99.97 43239	99.99 45883	99.98 82371	99.99 75286	99.99 75547	99.99 75803	99.97 77682	99.99 78099	99.70 33716	99.99 84293	99.25 80448	99.99 60665	99.99 84546	99.99 67141
FPR	0.005 6482	0.0097 5737	0.000 1518	0.006 85705	0.0005 7998	0.001 3729	0.0070 5175	0.000 0651	0.002 47768	0.000 0871	0.000 053	0.001 51013	0.0002 5676	0.000 0541	0.0001 1763	0.000 0247	0.000 0245	0.000 0242	0.0002 2232	0.000 0219	0.0029 6628	0.000 0157	0.007 41955	0.000 0393	0.000 0155	0.000 0329
RRSE	85.7682	101.801	18.2509	102.627	34.5891	58.5609	133.27	12.0178	74.8515	13.9419	12.3098	65.869	24.7756	12.2351	87.5349	7.4718	7.4683	7.4708	19.2768	7.4718	73.7487	6.9175	89.0538	10.2775	5.8612	7.8759
RAE	81.4361	103.694	2.0038	104.155	9.4823	17.157	88.8573	1.593	46.5799	1.9336	0.7581	43.2699	6.4697	1.8685	86.5127	0.2793	0.4021	0.2968	2.5804	0.2793	62.5164	0.2394	79.2697	4.2854	0.3721	0.3516
RMSE	0.295	0.35	0.063	0.353	0.119	0.201	0.458	0.041	0.257	0.048	0.042	0.226	0.085	0.042	0.301	0.026	0.026	0.026	0.066	0.026	0.254	0.024	0.306	0.035	0.02	0.027
MAE	0.192	0.245	0.005	0.246	0.022	0.041	0.21	0.004	0.11	0.005	0.002	0.102	0.015	0.004	0.204	7E-04	9E-04	7E-04	0.006	7E-04	0.148	6E-04	0.187	0.01	9E-04	8E-04
KV	0.409	0	0.98	0	0.928	0.828	0.097	0.992	0.669	0.988	0.992	0.777	0.962	0.992	0.982	0.997	0.997	0.997	0.973	0.997	0.609	0.998	0.236	0.996	0.998	0.996
MCR	47.922	89.116	1.6821	89.116	5.9367	14.182	73.45	0.6926	26.781	0.9565	0.6266	18.305	3.1332	0.6926	1.5172	0.2309	0.2309	0.2309	2.2098	0.2309	31.86	0.1979	61.643	0.3298	0.1319	0.2968
ACC	52.078	10.884	98.318	10.884	94.063	85.818	26.55	99.307	73.219	99.044	99.373	81.695	96.867	99.307	98.483	99.769	99.769	99.769	97.79	99.769	68.14	99.802	38.358	99.67	99.868	99.703
TT	0.001	0.001	0.07	0.01	0.03	0.01	4.47	0.03	0.01	0.01	0.02	0.01	0.07	0.01	0.02	1.53	0.75	0.63	74.39	50.56	28.76	0.03	0.001	0.01	0.03	0.01

Name of Classifiers	**Bayes Group**	DMNB	HMM	NB	SGM	**Function-based**	LDA	LLNR	LSVM	LR	MLP	MLP	QDA	RBF	RBFN	SLR	SMO	**Lazy Groups**	IB1	IBK	IBKLG	KSTAR	LKNN	LWL	RLKNN	**Rule-based**	CR	DTBL	DTNB	FURIA
PRC	1	0.99	1	0.69	0.97	1	0.99	1	0.17		0.99	1	1	1	1	1	1	1	1	0.99	1	1	1	1	1	1	0.97	0.98	1
MCC	1	1	1	0.73	0.98	1	1	1	0		0.99	1	1	1	1	1	1	0.99	0.99	0.99	1	1	1	1	0.99	1	0.98	0.98	1
ROC	1	1	1	0.84	0.99	1	1	1	0.5		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.99	1	1
PRE	99.99 6313	99.99 6363	99.99 85577	99.75 83622	99.98 40857	99.99 80274	99.99 73401	99.99 81529	99.38 43856		99.99 28895	99.99 59325	99.99 66399	99.99 66774	99.99 82895	99.99 89026	99.99 8545	99.99 54447	99.99 33673	99.99 38898	99.99 73185	99.997 7868	99.99 67055	99.99 79222	99.99 54358		99.99 7991	99.97 92353	99.98 00186	99.99 74836
FPR	0.000 0369	0.000 0364	0.000 0144	0.0024 1638	0.0001 5914	0.000 0197	0.000 0266	0.000 0185	0.0061 5614		0.000 0711	0.000 0407	0.000 0336	0.000 0332	0.000 0171	0.00 0011	0.000 0145	0.000 0456	0.000 0663	0.000 0611	0.000 0268	0.000 0221	0.000 0329	0.000 0208	0.000 0456		0.000 0201	0.000 20765	0.0001 9981	0.000 0252
RRSE	9.3308	9.7828	6.3148	82.0922	21.1333	7.4737	8.9305	7.4718	100		12.8889	8.8872	7.055	9.075	6.3063	5.6481	7.4142	9.4212	12.2351	11.2962	8.8872	6.0871	7.9743	7.9743	11.5493		6.3148	20.9438	93.9574	43.2378
RAE	0.538	0.4788	0.1995	33.7155	2.2344	0.2964	0.399	0.2793	100		1.4792	0.5601	0.8876	0.5957	0.2992	0.1596	1.888	1.5977	1.8685	0.6384	0.5601	0.5199	0.4642	0.4642	0.8583		0.1995	2.1945	95.1079	36.5055
RMSE	0.032	0.034	0.022	0.282	0.073	0.026	0.031	0.026	0.344		0.044	0.031	0.024	0.031	0.022	0.019	0.026	0.032	0.042	0.039	0.031	0.021	0.027	0.027	0.04		0.022	0.072	0.323	0.149
MAE	0.001	0.001	5E-04	0.08	0.005	7E-04	9E-04	7E-04	0.236		0.004	0.001	0.002	0.001	7E-04	4E-04	0.005	0.004	0.004	0.002	0.001	0.001	0.001	0.001	0.002		5E-04	0.005	0.225	0.086
KV	0.996	0.995	0.998	0.671	0.978	0.997	0.996	0.997	0		0.992	0.996	0.996	0.996	0.998	0.998	0.998	0.994	0.992	0.994	0.996	0.997	0.997	0.997	0.993		0.998	0.978	0.978	0.996
MCR	0.3628	0.3958	0.1649	27.869	1.847	0.2309	0.3298	0.2309	81.3		0.6926	0.3298	0.2968	0.2968	0.1649	0.1319	0.1649	0.4617	0.6926	0.5277	0.3298	0.2639	0.2639	0.2639	0.5937		0.1649	1.814	1.847	0.3298
ACC	99.637	99.604	99.835	72.131	98.153	99.769	99.67	99.769	18.701		99.307	99.67	99.703	99.703	99.835	99.868	99.835	99.538	99.307	99.472	99.67	99.736	99.736	99.736	99.406		99.835	98.186	98.153	99.67
TT	0.001	0.02	0.05	0.04	0.001	0.01	0.001	0.02	0.001		0.01	0.001	0.01	0.18	0.01	0.001	0.12	0.01	0.01	0.001	0.001	0.09	0.001	0.001	0.1		0.02	0.01	0.01	0.03

Name of Classifiers	JRIP	MODLEM	NNGE	OLM	ONER	PART	RIDOR	ROUGHS	ZEROR	**Decision Trees**	BFT	CDT	FPA	FT	J48	J48C	J48G	LADT	LMT	NBT	REPT	RF	RT	SC	SF	**Miscellaneous**	CHIRP	FLR	HP	VFI

5. J48Consolidated—A C4.5 Classifier Based on C4.5

J48Consolidated has been presented as the best classifier considering the decision tree group. Therefore, this section provides an in-depth analysis of J48Consodated.

5.1. Detection Capabilities of J48Consolidated

In this section, the J48Consolidated classifier is analyzed, considering the classification of the attack detection process. The classification threshold and the percentage of detection have been taken into consideration while analyzing attack classes. The attack-wise classification output for NSLKDD, ISCXIDS, and CICIDS2017 datasets has been presented in Figures 9–11, respectively.

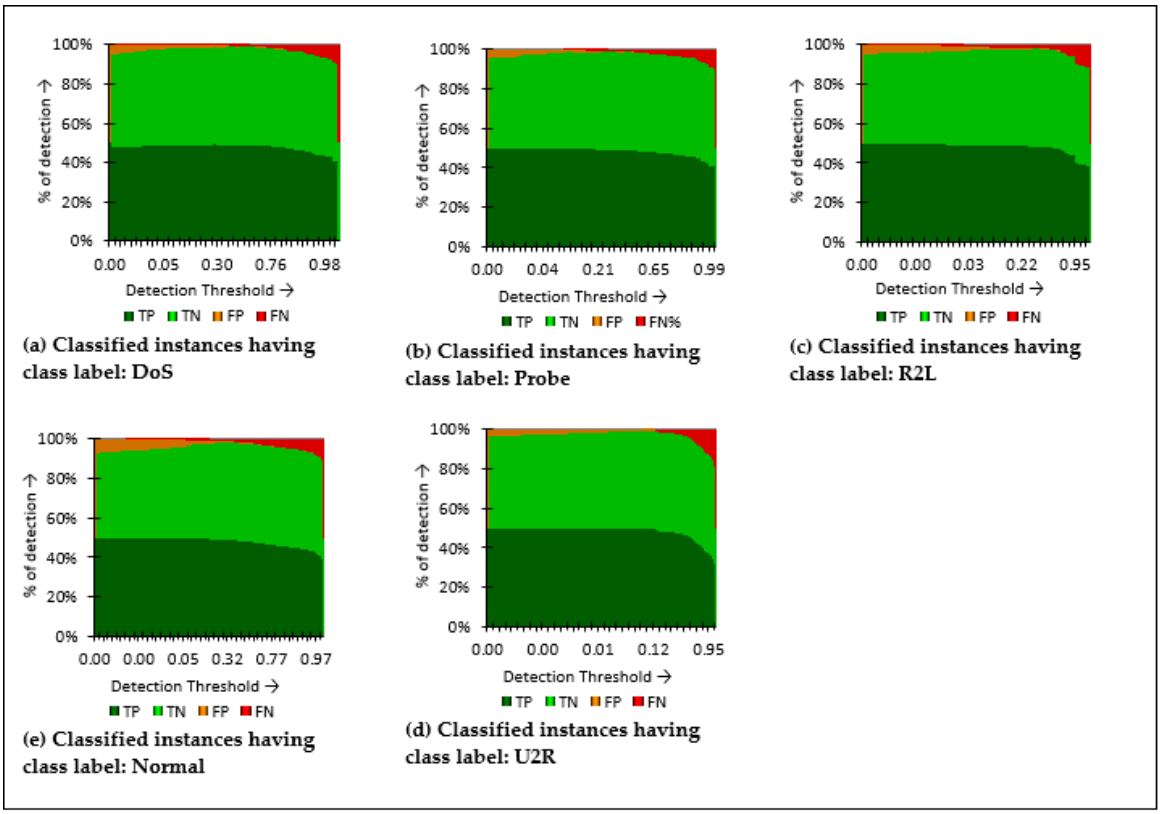

Figure 9. Detection (%) of attacks and normal class labels of NSL-KDD multi-class dataset.

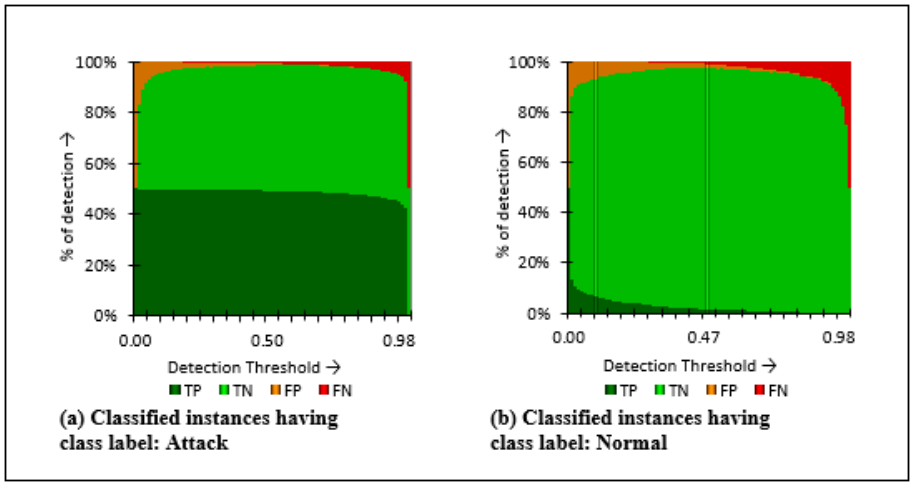

Figure 10. Detection (%) of attacks and normal class labels of ISCXIDS2012 binary class dataset.

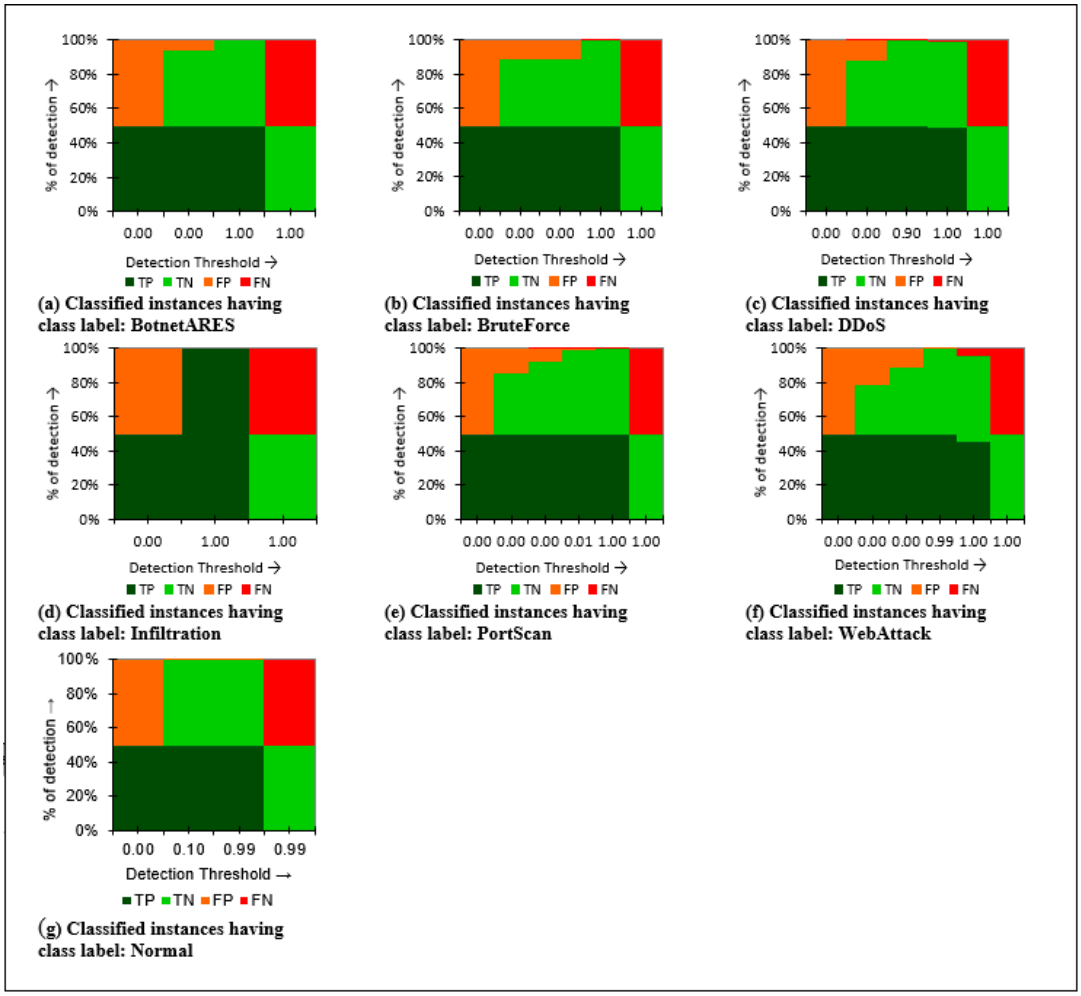

Figure 11. Detection (%) of attacks and normal class labels of CICIDS2017 multi class dataset.

The detection output for the NSLKDD dataset remains consistently good for DoS, Probe, R2L, U2R, and Normal classes with the increase in detection threshold. The U2R attack class shows low false positives, whereas few regular instances are misclassified during the classification process. Overall, the J48Consolidated classifier exhibited satisfactory performance for the NSLKDD dataset.

ISCXIDS2012 is a binary class dataset; therefore, J48Consolidated seems to generate false alarms. However, the presented results are low compared to the number of correctly classified instances (true positives and true negatives).

Finally, the individual J48Consolidated evaluation presents an effective classification considering six attack groups of the CICIDS2017 dataset. The classifier also differentiates regular instances with attack instances during the classification process.

5.2. Classification Output of J48Consolidated

The three IDS datasets are considered for a specific environment. The correlation of attributes, attacks, and benign instances varied from dataset to dataset. Therefore, J48Consolidated shows a different classification performance considering different IDS datasets. The classification output of J48Consolidated for NSLKDD, ISCXIDS2012, and CICIDS2017 datasets has been outlined in Figures 12–14, respectively.

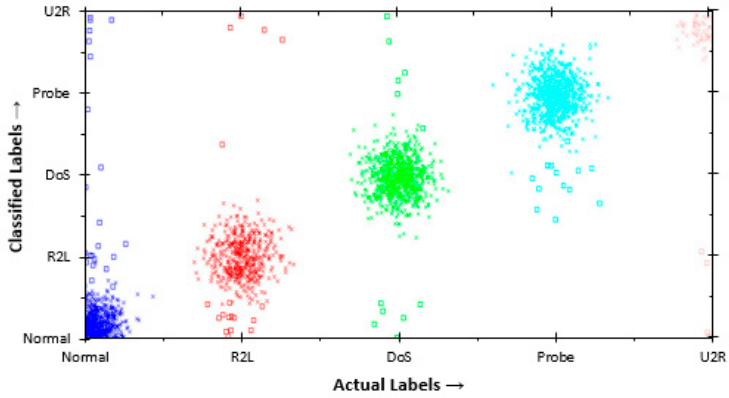

Figure 12. Classification of J48Consolidated on NSL-KDD dataset.

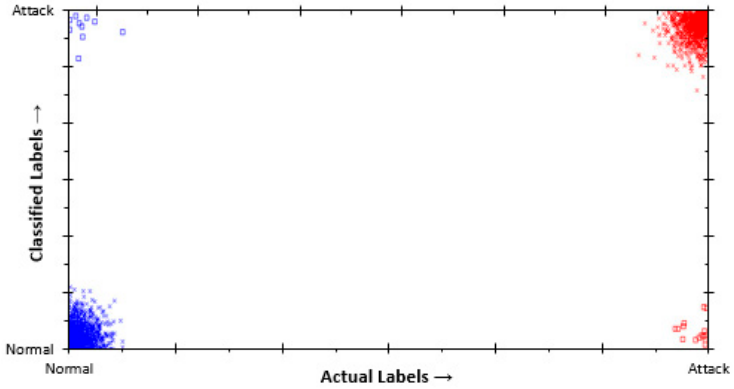

Figure 13. Classification of J48Consolidated on ISCXIDS2012 dataset.

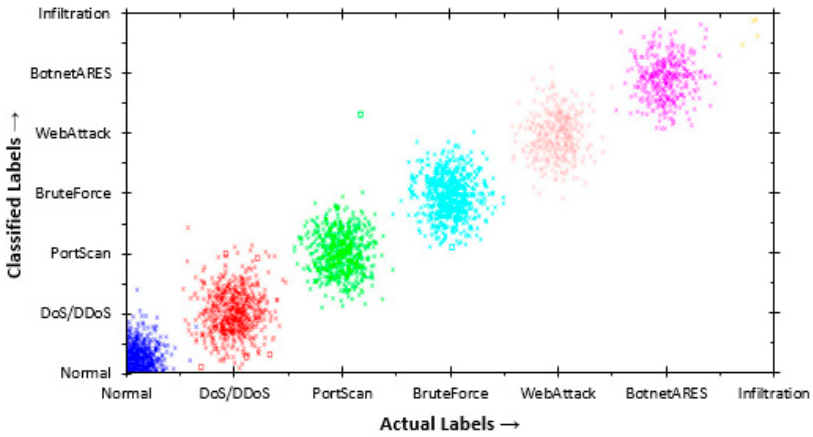

Figure 14. Classification of J48Consolidated on CICIDS2017 dataset.

Figure 12 shows that the J48Consolidated classifier presents a reliable classification in the NSLKDD dataset. Nevertheless, J48Consolidated also produced false alarms for positive and negative instances. Therefore, the authors recommend incorporating filter components such as data standardization and effective feature selection while designing IDSs using J48Considated. A filter component not only smooths the underlying data, but will also improve classification performance.

On the one hand, for the ISCXIDS2012 dataset, J48Consolidated dramatically showed improvement in classification. The classifier showed few false alarms. On the other hand, J48Consolidated successfully detected almost all the instances of the ISCXIDS2012 binary dataset. Therefore, the classifier achieved the highest TOPSIS score of 0.964 (Figure 8); thus, contributing to the highest average rank.

Finally, for the CICIDS2017 dataset, the J48Consolidated classifier presented a low number of false alarms. The six attack groups of the CICIDS2017 dataset presented a consistent classification with a detection accuracy of 99.868% (Table 17) and a low false positive of 0.000011.

A reliable IDS benchmark dataset must fulfill 11 criteria [122], such as complete network configuration, attack diversity, overall traffic, thorough interaction, labeled dataset, full capture, existing protocols, heterogeneity, feature set, anonymity, and metadata. The CICIDS2017 [123] dataset fulfills these criteria. Furthermore, CICIDS2017 is recent and focuses on the latest attack scenarios. The J48Consolidated classifier presented the best results for the CICIDS2017 dataset with an accuracy of 99.868%. Consequently, the J48Consolidated classifier can be assumed as an effective IDS with the CICIDS2017 dataset. Nevertheless, the authors recommend the incorporation of feature selection procedures at the preprocessing stage to extract the most relevant features of the dataset and promote system performance.

6. Conclusions

This paper analyzed fifty-four widely used classifiers spanning six different groups. These classifiers were evaluated on the three most popular intrusion detection datasets, i.e., NSLKDD, ISCXIDS2012 and CICIDS2017. The authors have extracted a sufficient number of random samples from these datasets, which retained the same class imbalance property of the original datasets. Consequently, multi-criteria decision-making has been used to allocate weight to these classifiers for different datasets. The rank of the classifiers was also finalized using those weights. First, an intragroup analysis has been conducted to find the best classifier group. Secondly, an intragroup analysis of the best classifier group has been made to find the best classifiers for the intrusion detection datasets. The authors analyzed thirteen performance metrics. Therefore, the best classifier has been selected impartially. On the one hand, the intergroup analysis presented the decision tree group of classifiers as the best classifier group, followed by the Rule-based classifiers, whereas the intragroup study identified J48Consolidated as the best classifier for high-class imbalance considering NSLKDD, CICIDS2017 and ISCXIDS2012 datasets. The J48Consolidated classifier provided the highest accuracy of 99.868%, a misclassification rate of 0.1319%, and a Kappa value of 0.998.

This study presented an in-depth analysis that provides numerous outcomes for IDS designers. Comparing fifty-four classifiers on intrusion detection datasets through thirteen performance matrices and ranking them is the main contributory work of this article. Nevertheless, the present study has limitations. Further investigation is required considering other datasets and other specific application domains. Moreover, the number of classes, class-wise performance observation, and classifiers' performance based on varying sample sizes should be carried out to understand the detailed aspects of the classifiers. The scalability and robustness of the classifiers were not tested. As a future work, many other IDS datasets can be used for ascertaining performance of the classifiers. Many recent ranking algorithms can be used as voting principle to obtain exact ranks of classifiers. Many other recent rule-based, decision forest classifiers were covered in this article; those classifiers can be analyzed to understand the real performance of the classifiers and classifier groups. Finally, J48Consolidated, which evolved as an ideal classifier out of this analysis, can be used along with a suitable feature selection technique to design robust intrusion detection systems.

Author Contributions: The individual contributions for this research are specified below: Conceptualization, R.P., S.B. and M.F.I.; Data curation, R.P. and M.F.I.; Formal analysis, A.K.B., M.P., C.L.C. and R.H.J.; Funding acquisition, R.H.J., C.L.C. and M.F.I.; Investigation; R.P., S.B., M.F.I. and A.K.B.; Methodology, R.P., S.B., C.L.C., M.F.I. and M.P.; Project administration, S.B., R.H.J., C.L.C. and A.K.B.; Resources, S.B., A.K.B., C.L.C. and M.P.; Software, R.P., C.L.C., M.F.I. and M.P.; Supervision, S.B., A.K.B., R.H.J.; Validation, R.P., M.F.I., C.L.C. and M.P.; Visualization, R.P., S.B., M.F.I., R.H.J. and A.K.B.; Writing—Review and editing, R.P., M.F.I., S.B., C.L.C., M.P., R.H.J. and A.K.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Sejong University research fund.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: NSL-KDD—https://www.unb.ca/cic/datasets/nsl.html (accessed on 1 February 2021), ISCXIDS2012—https://www.unb.ca/cic/datasets/ids.html (accessed on 1 February 2021), CICIDS2017—https://www.unb.ca/cic/datasets/ids-2017.html (accessed on 1 February 2021).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

Abbreviation	Description
TT	Testing Time
ACC	Accuracy
KV	Kappa Value
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
RAE	Relative Absolute Error
RRSE	Root Relative Squared Error
FPR	False Positive Rate
PRE	Precession
ROC	Receiver Operating Curve
MCC	Matthews Correlation Coefficient
PRC	Precision Recall Curve
TOPSIS	Techniques for Order Preference by Similarity to the Ideal Solution
IDS	Intrusion Detection System
IoT	Internet of Things
LWL	Locally Weighted Learning
RLKNN	Rseslib K-Nearest Neighbor
CR	Conjunctive Rule
DTBL	Decision Table
DTNB	Decision Table Naïve Bayes
FURIA	Fuzzy Rule Induction
NNGE	Nearest Neighbor with Generalization
OLM	Ordinal Learning Method
RIDOR	RIpple-DOwn Rule learner
BFT	Best-First Decision Tree
CDT	Criteria Based Decision Tree
LADT	Logit Boost based Alternating Decision Tree
LMT	Logistic Model Trees
NBT	Naïve Bayes based Decision Tree
REPT	Reduces Error Pruning Tree
RF	Random Forest
RT	Random Tree
SC	Simple Cart
CHIRP	Composite Hypercubes on Iterated Random Projections
FLR	Fuzzy Lattice Reasoning
HP	Hyper Pipes
VFI	Voting Feature Intervals
TP	True Positives
TN	True Negatives
TPR	True Positive Rate
TNR	True Negative Rate
FT	Functional Trees

References

1. Chavhan, S.; Gupta, D.; Chandana, B.N.; Khanna, A.; Rodrigues, J.J.P.C. IoT-based Context-Aware Intelligent Public Transport System in a metropolitan area. *IEEE Internet Things J.* **2019**, *7*, 6023–6034. [CrossRef]
2. Chen, Q.; Bridges, R.A. Automated behavioral analysis of malware: A case study of wannacry ransomware. In Proceedings of the 16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017, Cancun, Mexico, 18–21 December 2017; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, December 2017; Volume 2017, pp. 454–460.
3. Liang, W.; Li, K.C.; Long, J.; Kui, X.; Zomaya, A.Y. An Industrial Network Intrusion Detection Algorithm Based on Multifeature Data Clustering Optimization Model. *IEEE Trans. Ind. Inform.* **2020**, *16*, 2063–2071. [CrossRef]
4. Jiang, K.; Wang, W.; Wang, A.; Wu, H. Network Intrusion Detection Combined Hybrid Sampling with Deep Hierarchical Network. *IEEE Access* **2020**, *8*, 32464–32476. [CrossRef]
5. Zhang, Y.; Li, P.; Wang, X. Intrusion Detection for IoT Based on Improved Genetic Algorithm and Deep Belief Network. *IEEE Access* **2019**, *7*, 31711–31722. [CrossRef]
6. Yang, H.; Wang, F. Wireless Network Intrusion Detection Based on Improved Convolutional Neural Network. *IEEE Access* **2019**, *7*, 64366–64374. [CrossRef]
7. Lever, J.; Krzywinski, M.; Altman, N. Model selection and overfitting. *Nat. Methods* **2016**, *13*, 703–704. [CrossRef]
8. Krawczyk, B. Learning from imbalanced data: Open challenges and future directions. *Prog. Artif. Intell.* **2016**, *5*, 221–232. [CrossRef]
9. Pes, B. Learning from high-dimensional biomedical datasets: The issue of class imbalance. *IEEE Access* **2020**, *8*, 13527–13540. [CrossRef]
10. Wang, S.; Yao, X. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2012**, *42*, 1119–1130. [CrossRef] [PubMed]
11. Ho, T.K.; Basu, M. Complexity measures of supervised classification problems. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 289–300.
12. Kelly, M.G.; Hand, D.J.; Adams, N.M. Supervised classification problems: How to be both judge and jury. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin, Germany, 1999; Volume 1642, pp. 235–244.
13. Kuncheva, L.I. *Combining Pattern Classifiers: Methods and Algorithms: Second Edition*; Wiley: Hoboken, NJ, USA, 2014; Volume 9781118315, ISBN 9781118914564.
14. Jain, A.K.; Duin, R.P.W.; Mao, J. Statistical pattern recognition: A review. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 4–37. [CrossRef]
15. Lashkari, A.H.; Gil, G.D.; Mamun, M.S.I.; Ghorbani, A.A. Characterization of tor traffic using time based features. In Proceedings of the ICISSP 2017 3rd International Conference on Information Systems Security and Privacy, Porto, Portugal, 19–21 February 2017; SciTePress: Setúbal, Portugal, 2017; Volume 2017-Janua, pp. 253–262.
16. Robert, C. Machine Learning, a Probabilistic Perspective. *CHANCE* **2014**, *27*, 62–63. [CrossRef]
17. Maindonald, J. *Pattern Recognition and Machine Learning*; Journal of Statistical Software: Los Angeles, CA, USA, 2007; Volume 17.
18. Frasca, T.M.; Sestito, A.G.; Versek, C.; Dow, D.E.; Husowitz, B.C.; Derbinsky, N. A Comparison of Supervised Learning Algorithms for Telerobotic Control Using Electromyography Signals. In Proceedings of the 30th AAAI Conference on Artificial Intelligence, AAAI 2016, Phoenix, AZ, USA, 12–17 February 2016; pp. 4208–4209. Available online: www.aaai.org (accessed on 12 May 2020).
19. Soru, T.; Ngomo, A.C.N. A comparison of supervised learning classifiers for link discovery. *ACM Int. Conf. Proceeding Ser.* **2014**, 41–44. [CrossRef]
20. Arriaga-Gómez, M.F.; De Mendizábal-Vázquez, I.; Ros-Gómez, R.; Sánchez-Ávila, C. A comparative survey on supervised classifiers for face recognition. In Proceedings of the International Carnahan Conference on Security Technology, Hatfield, UK, 13–16 October 2014; Volume 2014, pp. 1–6.
21. Shiraishi, T.; Motohka, T.; Thapa, R.B.; Watanabe, M.; Shimada, M. Comparative assessment of supervised classifiers for land use-land cover classification in a tropical region using time-series PALSAR mosaic data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 1186–1199. [CrossRef]
22. Micó, L.; Oncina, J. Comparison of fast nearest neighbour classifiers for handwritten character recognition. *Pattern Recognit. Lett.* **1998**, *19*, 351–356. [CrossRef]
23. Sianaki, O.A. Intelligent Decision Support System for Energy Management in Demand Response Programs and Residential and Industrial Sectors of the Smart Grid. Ph.D. Thesis, Curtin University, Bentley, WA, Australia, 2015.
24. Hwang, C.; Masud, A. *Multiple Objective Decision Making—Methods and Applications: A State-Of-The-Art Survey*; Springer: New York, NY, USA, 2012.
25. Radanliev, P.; De Roure, D.; Page, K.; Van Kleek, M.; Santos, O.; Maddox, L.; Burnap, P.; Anthi, E.; Maple, C. Design of a dynamic and self-adapting system, supported with artificial intelligence, machine learning and real-time intelligence for predictive cyber risk analytics in extreme environments—Cyber risk in the colonisation of Mars. *Saf. Extrem. Environ.* **2021**, 1–12. [CrossRef]
26. Wu, X.; Kumar, V.; Ross, Q.J.; Ghosh, J.; Yang, Q.; Motoda, H.; McLachlan, G.J.; Ng, A.; Liu, B.; Yu, P.S.; et al. Top 10 algorithms in data mining. *Knowl. Inf. Syst.* **2008**, *14*, 1–37. [CrossRef]
27. Kotsiantis, S.B.; Zaharakis, I.; Pintelas, P. Supervised machine learning: A review of classification techniques. *Emerg. Artif. Intell. Appl. Comput. Eng.* **2007**, *160*, 3–24.

28. Demusar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
29. Chand, N.; Mishra, P.; Krishna, C.R.; Pilli, E.S.; Govil, M.C. A comparative analysis of SVM and its stacking with other classification algorithm for intrusion detection. In Proceedings of the 2016 International Conference on Advances in Computing, Communication and Automation, ICACCA, Dehradun, India, 8–9 April 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–6.
30. Htike, K.K.; Khalifa, O.O. Comparison of supervised and unsupervised learning classifiers for human posture recognition. In Proceedings of the International Conference on Computer and Communication Engineering (ICCCE 2010), Kuala Lumpur, Malaysia, 11–13 May 2010. [CrossRef]
31. Tuysuzoglu, G.; Yaslan, Y. Gözetimli Siniflandiricilar ve Topluluk Temelli Sözlükler ile Biyomedikal Veri Siniflandirilmasi. In Proceedings of the 25th Signal Processing and Communications Applications Conference, SIU 2017, Antalya, Turkey, 15–18 May 2017; pp. 1–4. [CrossRef]
32. Gu, S.; Jin, Y. Multi-train: A semi-supervised heterogeneous ensemble classifier. *Neurocomputing* **2017**, *249*, 202–211. [CrossRef]
33. Labatut, V.; Cherifi, H. Accuracy Measures for the Comparison of Classifiers. *arXiv* **2012**, arXiv:abs/1207.3790.
34. Caruana, R.; Niculescu-Mizil, A. An empirical comparison of supervised learning algorithms. In Proceedings of the 23rd International Conference on Machine Learning, Hong Kong, China, 18–22 December 2006; Volume 148, pp. 161–168.
35. Amancio, D.R.; Comin, C.H.; Casanova, D.; Travieso, G.; Bruno, O.M.; Rodrigues, F.A.; Da Fontoura Costa, L. A systematic comparison of supervised classifiers. *PLoS ONE* **2014**, *9*, e94137. [CrossRef]
36. Araar, A.; Bouslama, R. A comparative study of classification models for detection in ip networks intrusions. *J. Theor. Appl. Inf. Technol.* **2014**, *64*, 107–114.
37. Gharibian, F.; Ghorbani, A.A. Comparative study of supervised machine learning techniques for intrusion detection. In Proceedings of the Fifth Annual Conference on Communication Networks and Services Research (CNSR 2007), Fredericton, NB, Canada, 14–17 May 2007; pp. 350–355.
38. Panda, M.; Patra, M.R. A comparative study of data mining algorithms for network intrusion detection. In Proceedings of the 1st International Conference on Emerging Trends in Engineering and Technology, ICETET 2008, Maharashtra, India, 16–18 July 2008; pp. 504–507.
39. Srinivasulu, P.; Nagaraju, D.; Kumar, P.R.; Rao, K.N. Classifying the network intrusion attacks using data mining classification methods and their performance comparison. *Int. J. Comput. Sci. Netw. Secur.* **2009**, *9*, 11–18.
40. Wu, S.Y.; Yen, E. Data mining-based intrusion detectors. *Expert Syst. Appl.* **2009**, *36*, 5605–5612. [CrossRef]
41. Jalil, K.A.; Kamarudin, M.H.; Masrek, M.N. Comparison of machine learning algorithms performance in detecting network intrusion. In Proceedings of the 2010 International Conference on Networking and Information Technology, Manila, Philippines, 11–12 June 2010; pp. 221–226. [CrossRef]
42. Amudha, P.; Rauf, H.A. Performance analysis of data mining approaches in intrusion detection. In Proceedings of the 2011 International Conference on Process Automation, Control and Computing, Coimbatore, India, 20–22 July 2011.
43. China Appala Naidu, R.; Avadhani, P.S. A comparison of data mining techniques for intrusion detection. In Proceedings of the IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), Ramanathapuram, India, 23–25 August 2012; pp. 41–44.
44. Kalyani, G. Performance Assessment of Different Classification Techniques for Intrusion Detection. *IOSR J. Comput. Eng.* **2012**, *7*, 25–29. [CrossRef]
45. Thaseen, S.; Kumar, C.A. An analysis of supervised tree based classifiers for intrusion detection system. In Proceedings of the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering, Salem, India, 21–22 February 2013; pp. 294–299.
46. Revathi, S.; Malathi, A. A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection. *Int. J. Eng. Res. Technol.* **2013**, *2*, 1848–1853.
47. Robinson, R.R.R.; Thomas, C. Ranking of machine learning algorithms based on the performance in classifying DDoS attacks. In Proceedings of the 2015 IEEE Recent Advances in Intelligent Computational Systems, RAICS 2015, Trivandrum, Kerala, 10–12 December 2015; pp. 185–190.
48. Choudhury, S.; Bhowal, A. Comparative analysis of machine learning algorithms along with classifiers for network intrusion detection. In Proceedings of the 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Avadi, India, 6–8 May 2015; pp. 89–95.
49. Jain, A.; Rana, J.L. Classifier Selection Models for Intrusion Detection System (Ids). *Inform. Eng. Int. J.* **2016**, *4*, 1–11.
50. Bostani, H.; Sheikhan, M. Modification of supervised OPF-based intrusion detection systems using unsupervised learning and social network concept. *Pattern Recognit.* **2017**, *62*, 56–72. [CrossRef]
51. Belavagi, M.C.; Muniyal, B. Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection. *Procedia Comput. Sci.* **2016**, *89*, 117–123. [CrossRef]
52. Almseidin, M.; Alzubi, M.; Kovacs, S.; Alkasassbeh, M. Evaluation of Machine Learning Algorithms for Intrusion Detection System. In Proceedings of the 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, 14–16 September 2017.
53. Amira, A.S.; Hanafi, S.E.O.; Hassanien, A.E. Comparison of classification techniques applied for network intrusion detection and classification. *J. Appl. Log.* **2017**, *24*, 109–118.

54. Aksu, D.; Üstebay, S.; Aydin, M.A.; Atmaca, T. Intrusion detection with comparative analysis of supervised learning techniques and fisher score feature selection algorithm. In *Communications in Computer and Information Science*; Springer: Berlin, Germany, 2018; Volume 935, pp. 141–149.
55. Nehra, D.; Kumar, K.; Mangat, V. Pragmatic Analysis of Machine Learning Techniques in Network Based IDS. In *Proceedings of the International Conference on Advanced Informatics for Computing Research*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 422–430.
56. Mahfouz, A.M.; Venugopal, D.; Shiva, S.G. *Comparative Analysis of ML Classifiers for Network Intrusion Detection*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 193–207.
57. Rajagopal, S.; Siddaramappa Hareesha, K.; Panduranga Kundapur, P. Performance analysis of binary and multiclass models using azure machine learning. *Int. J. Electr. Comput. Eng.* **2020**, *10*, 978. [CrossRef]
58. Ahmim, A.; Ferrag, M.A.; Maglaras, L.; Derdour, M.; Janicke, H. *A Detailed Analysis of Using Supervised Machine Learning for Intrusion Detection*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 629–639.
59. Frank, E.; Hall, M.A.; Witten, I.H. *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques."*; Morgan Kaufmann: Burlington, VT, USA, 2016; p. 128.
60. Su, J.; Zhang, H.; Ling, C.X.; Matwin, S. Discriminative parameter learning for Bayesian networks. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 1016–1023.
61. Yu, S.-Z. Hidden semi-Markov models. *Artif. Intell.* **2010**, *174*, 215–243. [CrossRef]
62. Ghahramani, Z. An introduction to Hidden Markov Models and Bayesian Networks. In *Hidden Markov Models*; World Scientific: Singapore, 2001; Volume 15.
63. Zhang, H. Exploring conditions for the optimality of naïve bayes. *Proc. Int. J. Pattern Recognit. Artif. Intell.* **2005**, *19*, 183–198. [CrossRef]
64. John, G.H.; Langley, P. Estimating Continuous Distributions in Bayesian Classifiers George. *Proc. Elev. Conf. Uncertain. Artif. Intell.* **1995**, *42*, 338–345.
65. Puurula, A. Scalable Text Classification with Sparse Generative Modeling. In *Lecture Notes in Computer Science*; Springer: New York, NY, USA, 2012; pp. 458–469.
66. Balakrishnama, S.; Ganapathiraju, A. *Linear Discriminant Analysis—A Brief Tutorial*; Institute for Signal and information Processing: Philadelphia, PA, USA, 1998; Volume 18, pp. 1–8.
67. Fan, R.E.; Chang, K.W.; Hsieh, C.J.; Wang, X.R.; Lin, C.J. LIBLINEAR: A library for large Linear Classification. *J. Mach. Learn. Res.* **2008**, *9*, 1871–1874.
68. Chang, C.C.; Lin, C.J. LIBSVM: A Library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 1–27. [CrossRef]
69. Kleinbaum, D.G.; Klein, M. *Introduction to Logistic Regression*; Springer: New York, NY, USA, 2010; pp. 1–39.
70. Windeatt, T. Accuracy/diversity and ensemble MLP classifier design. *IEEE Trans. Neural Netw.* **2006**, *17*, 1194–1211. [CrossRef]
71. Hertz, J.; Krogh, A.; Palmer, R.G. *Introduction to the Theory of Neural Computation*; Elsevier Science Publishers: Amsterdam, The Netherlands, 2018; ISBN 9780429968211.
72. Yang, Q.; Cheng, G. Quadratic Discriminant Analysis under Moderate Dimension. *Stat. Theory*. 2018. Available online: http://arxiv.org/abs/1808.10065 (accessed on 12 May 2020).
73. Frank, E. *Fully Supervised Training of Gaussian Radial Basis Function Networks in WEKA*; University of Waikato: Hamilton, New Zealand, 2014; Volume 04.
74. Schwenker, F.; Kestler, H.A.; Palm, G. Unsupervised and Supervised Learning in Radial-Basis-Function Networks. In *Self-Organizing Neural Networks*; Physica Verlag: Heidelberg, Germany, 2002; pp. 217–243.
75. Kyburg, H.E. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference by Judea Pearl. *J. Philos.* **1991**, *88*, 434–437. [CrossRef]
76. Kecman, V. *Support Vector Machines—An Introduction*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 1–47.
77. Keerthi, S.S.; Shevade, S.K.; Bhattacharyya, C.; Murthy, K.R.K. Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Comput.* **2001**, *13*, 637–649. [CrossRef]
78. Aha, D.W.; Kibler, D.; Albert, M.K. Instance-Based Learning Algorithms. In *Machine Learning*; Springer: Berlin, Germany, 1991; Volume 6, pp. 37–66.
79. Cleary, J.G.; Trigg, L.E. *K**: An Instance-based Learner Using an Entropic Distance Measure. In *Machine Learning Proceedings 1995*; Morgan Kaufmann: New York, NY, USA, 1995.
80. Wojna, A.; Latkowski, R. Rseslib 3: Library of rough set and machine learning methods with extensible architecture. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 10810 LNCS, pp. 301–323.
81. Frank, E.; Hall, M.; Pfahringer, B. *Locally Weighted Naive Bayes*; University of Waikato: Hamilton, New Zealand, 2012.
82. Atkeson, C.G.; Moore, A.W.; Schaal, S. Locally Weighted Learning. *Artif. Intell. Rev.* **1997**, *11*, 11–73. [CrossRef]
83. Zimek EM (Documentation for extended WEKA including Ensembles of Hierarchically Nested Dichotomies). Available online: http://www.dbs.ifi.lmu.de/~{}zimek/diplomathesis/implementations/EHNDs/doc/weka/clusterers/FarthestFirst.html (accessed on 12 May 2020).
84. Kohavi, R. The power of decision tables. In *Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 1995; Volume 912, pp. 174–189.

85. Hall, M.A.; Frank, E. Combining naive bayes and decision tables. In Proceedings of the FLAIRS Conference, Coconut Grove, FL, USA, 15–17 May 2008; Volume 2118, pp. 318–319.
86. Hühn, J.; Hüllermeier, E. FURIA: An algorithm for unordered fuzzy rule induction. *Data Min. Knowl. Discov.* **2009**, *19*, 293–319. [CrossRef]
87. Cohen, W.W. Fast Effective Rule Induction. In Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, CA, USA, 9–12 July 1995.
88. Stefanowski, J. Rough set based rule induction techniques for classification problems. In *Rough Set Based Rule Induction Techniques for Classification Problems*; Intelligent Techniques & Soft Computing: Aachen, Germany, 1998; Volume 1, pp. 109–113.
89. Sylvain, R. *Nearest Neighbor with Generalization*; University of Canterbury: Christchurch, New Zealand, 2002.
90. Martin, B. *Instance-Based Learning: Nearest Neighbor with Generalization*; University of Waikato: Hamilton, New Zealand, 1995.
91. Ben-David, A. Automatic Generation of Symbolic Multiattribute Ordinal Knowledge-Based DSSs: Methodology and Applications. *Decis. Sci.* **1992**, *23*, 1357–1372. [CrossRef]
92. Holte, R.C. Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. *Mach. Learn.* **1993**, *11*, 63–90. [CrossRef]
93. Frank, E.; Wang, Y.; Inglis, S.; Holmes, G.; Witten, I.H. Using model trees for classification. *Mach. Learn.* **1998**, *32*, 63–76. [CrossRef]
94. Thangaraj, M. Vijayalakshmi Performance Study on Rule-based Classification Techniques across Multiple Database Relations. *Int. J. Appl. Inf. Syst.* **2013**, *5*, 1–7.
95. Pawlak, Z. Rough set theory and its applications to data analysis. *Cybern. Syst.* **1998**, *29*, 661–688. [CrossRef]
96. Frank, E. ZeroR. Weka 3.8 Documentation. 2019. Available online: https://weka.sourceforge.io/doc.stable-3-8/weka/classifiers/rules/ZeroR.html (accessed on 12 May 2020).
97. Suthaharan, S. Decision Tree Learning. In *Machine Learning Models and Algorithms for Big Data Classification. Integrated Series in Information Systems*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 237–269.
98. Abellán, J.; Moral, S. Building Classification Trees Using the Total Uncertainty Criterion. *Int. J. Intell. Syst.* **2003**, *18*, 1215–1225. [CrossRef]
99. Adnan, M.N.; Islam, M.Z. Forest PA: Constructing a decision forest by penalizing attributes used in previous trees. *Expert Syst. Appl.* **2017**, *89*, 389–403. [CrossRef]
100. Gama, J. Functional trees. *Mach. Learn.* **2004**, *55*, 219–250. [CrossRef]
101. Salzberg, S.L. C4.5: Programs for Machine Learning by J. Ross Quinlan. In *Machine Learning*; Morgan Kaufmann Publishers, Inc.: New York, NY, USA, 1993; Volume 16, pp. 235–240.
102. Ibarguren, I.; Pérez, J.M.; Muguerza, J.; Gurrutxaga, I.; Arbelaitz, O. Coverage-based resampling: Building robust consolidated decision trees. *Knowledge-Based Syst.* **2015**, *79*, 51–67. [CrossRef]
103. Hayashi, Y.; Tanaka, Y.; Takagi, T.; Saito, T.; Iiduka, H.; Kikuchi, H.; Bologna, G.; Mitra, S. Recursive-rule extraction algorithm with J48graft and applications to generating credit scores. *J. Artif. Intell. Soft Comput. Res.* **2016**, *6*, 35–44. [CrossRef]
104. Holmes, G.; Pfahringer, B.; Kirkby, R.; Frank, E.; Hall, M. Multiclass alternating decision trees. In *Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2002; Volume 2430, pp. 161–172.
105. Landwehr, N.; Hall, M.; Frank, E. Logistic model trees. *Mach. Learn.* **2005**, *59*, 161–205. [CrossRef]
106. Sumner, M.; Frank, E.; Hall, M. Speeding up Logistic Model Tree induction. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Samos, Greece, 18–20 July 2005; Volume 3721 LNAI, pp. 675–683.
107. Jiang, L.; Li, C. Scaling up the accuracy of decision-tree classifiers: A naive-bayes combination. *J. Comput.* **2011**, *6*, 1325–1331. [CrossRef]
108. Kalmegh, S. Analysis of WEKA Data Mining Algorithm REPTree, Simple Cart and RandomTree for Classification of Indian News. *Int. J. Innov. Sci. Eng. Technol.* **2015**, *2*, 438–446.
109. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
110. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
111. Witten, I.H.; Frank, E.; Hall, M.A.; Pal, C.J. *Data Mining: Practical Machine Learning Tools and Techniques*; Elsevier: Amsterdam, The Netherlands, 2016; ISBN 9780128042915.
112. Islam, Z.; Giggins, H. Knowledge Discovery through SysFor: A Systematically Developed Forest of Multiple Decision Trees kDMI: A Novel Method for Missing Values Imputation Using Two Levels of Horizontal Partitioning in a Data set View project A Hybrid Clustering Technique Combining a Novel Genetic Algorithm with K-Means View project Knowledge Discovery through SysFor-a Systematically Developed Forest of Multiple Decision Trees. 2011. Available online: https://www.researchgate.net/publication/236894348 (accessed on 11 May 2020).
113. Wilkinson, L.; Anand, A.; Tuan, D.N. CHIRP: A new classifier based on composite hypercubes on iterated random projections. In Proceedings of the Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011; pp. 6–14.

114. Athanasiadis, I.N.; Kaburlasos, V.G.; Mitkas, P.A.; Petridis, V. Applying Machine Learning Techniques on Air Quality Data for Real-Time Decision Support 1 Introduction 2 Decision support systems for assessing air quality in real—time 3 The σ—FLNMAP Classifier. *First Int. Symp. Inf. Technol. Environ. Eng.* **2003**, 2–7. Available online: http://www.academia.edu/download/53083886/Applying_machine_learning_techniques_on_20170511-3627-1jgoy73.pdf (accessed on 11 May 2020).
115. Deeb, Z.A.; Devine, T. *Randomized Decimation HyperPipes*; Penn State University: University Park, PA, USA, 2010.
116. Demiröz, G.; Altay Güvenir, H. Classification by voting feature intervals. In *Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 1997; Volume 1224, pp. 85–92.
117. Ingre, B.; Yadav, A. Performance analysis of NSL-KDD dataset using ANN. In Proceedings of the 2015 International Conference on Signal Processing and Communication Engineering Systems, Guntur, India, 2–3 January 2015; pp. 92–96. [CrossRef]
118. Ibrahim, L.M.; Taha, D.B.; Mahmod, M.S. A comparison study for intrusion database (KDD99, NSL-KDD) based on self organization map (SOM) artificial neural network. *J. Eng. Sci. Technol.* **2013**, *8*, 107–119.
119. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6.
120. Shiravi, A.; Shiravi, H.; Tavallaee, M.; Ghorbani, A.A. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Comput. Secur.* **2012**, *31*, 357–374. [CrossRef]
121. Gharib, A.; Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. An Evaluation Framework for Intrusion Detection Dataset. In Proceedings of the ICISS 2016—2016 International Conference on Information Science and Security, Jaipur, India, 19–22 December 2016; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2017.
122. Panigrahi, R.; Borah, S. *Design and Development of a Host Based Intrusion Detection System with Classification of Alerts*; Sikkim Manipal University: Sikkim, India, 2020.
123. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proceedings of the ICISSP 2018, Madeira, Portugal, 22–24 January 2018; pp. 108–116.