OPEN ACCESS
International Journal of Innovations in Science & Technology
IJIST
P ISSN : 2618-1630
E ISSN : 2909-6130
Original Article

Performance Evaluation of Classification Algorithms for Intrusion Detection on NSL-KDD Using Rapid Miner

Zoraez Ashfaq Malik[1], Marriam Siddiqui[1], Azhar Imran[1], Aman Ullah Yasin[1], Abdul Haleem Butt[1], Zahir Javed Paracha[2]

[1]Department of Creative Technologies, Faculty of Computing & AI (Air University, Islamabad)

[2]Department of Electrical Engineering, Pakistan Institute of Electrical Engineering, Multan

* Correspondence: Dr. Azhar Imran Mudassir (azharimran63@gmail.com)

Citation | Malik, Zoraez Ashfaq, Marriam Siddique, Zahir Javed Paracha, Azhar Imran, Amanullah Yasin, and Abdul Hameed Butt. 2022. "Performance Evaluation of Classification Algorithms for Intrusion Detection on NSL-KDD Using Rapid Miner". International Journal of Innovations in Science & Technology 4 (1):135-146.

https://journal.50sea.com/index.php/IJIST/article/view/101

DOI |https://doi.org/10.33411/IJIST/2022040110.

Received | Dec 7, 2021; **Revised** | Dec 18, 2021 **Accepted** | Dec 22, 2021; **Published** | Feb 19, 2022.

The rapid advancement of the internet and its exponentially increasing usage has also exposed it to several vulnerabilities. Consequently, it has become an extremely important that can prevent network security issues. One of the most commonly implemented solutions is Intrusion Detection System (IDS) that can detect unusual attacks and unauthorized access to a secured network. In the past, several machine learning algorithms have been evaluated on the KDD intrusion dataset. However, this paper focuses on the implementation of the four machine learning algorithms: KNN, Random Forest, gradient boosted tree and decision tree. The models are also implemented through the Auto Model feature to determine its convenience. The results show that Gradient Boosted trees have achieved the highest accuracy (99.42%) in comparison to random forest algorithm that achieved the lowest accuracy (93.63%).

Keywords: Intrusion detection system; Machine learning; RapidMiner; NSL-KDD and Gradient boosted tree.

Author's Contribution. All the authors contributed equally.

Conflict of interest. We declare no conflict of interest for publishing this manuscript in IJIST.

Project details. Nil

1. Introduction

The complexities of cyber-attacks are increasing with time and consequently, their malice too. In today's world, every networked environment must take high-level security measures to ensure secure and reliable communication between several organizations. Software or device that inspects traffic of a network for any violation or malicious activity is termed as an intrusion detection system (IDS). This safeguard system is placed at one or more strategic points in a network to detect suspicious activity [1]. All the traffic from and to the devices, connected to the network is analyzed and the activities are matched to the known attacks. Any violation or malicious activity is collected centrally and reported through a piece of Security Information and Event Management System (SIEM).

The need for an intrusion detection system is undeniable; thus, an accurate model must be developed. In this field, machine learning has proven to be an effective investigation device that can detect any irregular event taking place in any system's traffic [2]. Various models based on machine learning algorithms have been used for the detection of cyber-attacks [3]. Many researchers have studied machine learning algorithms and implemented them on NSL-KDD data set to enhance the performance of cyber-attack detection mechanisms. These include Artificial Neural Networks, Naïve Bayesian algorithms, self-organizing maps, Support Vector Machines, Random Forests, and much more [4]. Salama et al. 2019 applied a restricted Boltzmann machine (RBM) on the same dataset for feature reduction and then implemented a support vector machine (SVM) as a classifier with an approximate accuracy of 87% [5].

Amreen et al. [6] proposed an Intelligent Network IDS using Average One Dependence Estimators (AODE) which is an improved variant of Naïve Bayes. The researchers also established a network anomaly detection system with the help of discriminative RBM in combination with generative models. This system provided reasonable accuracy for gathering information from training data[7]. Solane Duque et al. built a model with a k-means machine learning algorithm and observed a high-efficiency rate along with low false negative and positive rates. It was implied that this algorithm could be implemented on a signature-based approach to lessen the false-negative rate. It was observed that random forest classifier provided better performance as compared to other algorithms [8]. Furthermore, the researchers evaluated eight tree-based classification algorithms to predict network attacks[9]. In addition to this, the researchers have also worked on the spots where the performance of IDS can be improved using the deep learning models. Shone presented a non-symmetric deep autoencoder (NDAE) based on a deep learning technique for unsupervised feature-based learning [10]. Another novel approach was proposed by Tao Ma et al. known as SCDNN in which a deep neural network (DNN) was implemented along with Spectral Clustering (SC) [11].

A recent approach to the implementation of the machine learning algorithms is integrated into environments like WEKA, Knime, Orange, Keel, Azure, IBM SPSS Modeler, and Scikit-Learn[12].

This research explores the usability of Rapid Miner for the implementation of machine learning algorithms for IDS. This paper aims to implement the machine learning algorithms on Rapid-Miner. This method will also determine the ease of utilizing a built platform for data science tools as well as to evaluate five different classifiers on NSLKDD as given in Figure 1.

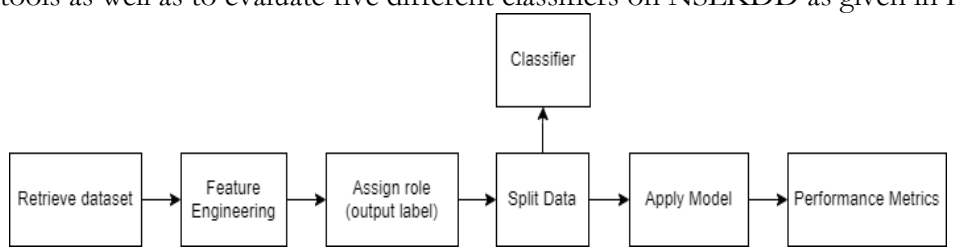

Figure 1. General Flowchart for Classification algorithms

2. Research Methodology

Data Mining Tool

In this paper, models for the classification algorithms were developed on RapidMiner using the NSL-KDD dataset to evaluate their performance. Rapid-Miner is a platform that offers an integrated environment that allows to perform the data preparation and pre-processing, text mining, predictive analytics, machine learning, and deep learning. The ROI-centric approach-based software allows the users to conveniently build and test models using block coding [13]. The end-to-end data science platform offers a wide range of processes that work together seamlessly.

Dataset

The classification algorithms were evaluated on the NSL-KDD dataset. The dataset can be downloaded from the website of the University of New Brunswick. One of the earlier datasets used to develop an IDS and a predictive model to differentiate between intrusions and normal connections is KDD'99 that is obtained as a result of the Knowledge Discovery and Data Mining Tools Competition (KDD cup) [14]. A cleaned up and revised version of KDD'99 has been built, known as Network Security Laboratory- Knowledge Discovery and Data Mining Tools (NSL-KDD) data set.

The new dataset has resolved some intrinsic issues of its predecessor, yet it is not a perfect depiction of current real networks[15]. It is due to insufficient public data sets for IDSs built for networks. However, the dataset can be employed as an effectual standard data set by the researchers to classify different intrusion detection methods. It is mainly because the records for the train and test datasets have reasonable examples, providing comparable and consistent research evaluation results. Moreover, the training set does not include redundant examples and thus, the bias for frequent examples can be avoided. In addition to this, duplicate examples are not included, preventing the bias towards the methods that offer better rates of detection on the frequent examples. From each difficulty level group, some examples are selected which are inversely proportional to the percentage of examples in the previous KDD dataset mentioned in Table 1. As a result, NSL-KDD accurately evaluates different learning algorithms.

Table 1. Dataset Description

Classes		Subclasses
DoS	1.	apache2
	2.	back
	3.	land
	4.	Nep tune
	5.	mailbomb
	6.	pod
	7.	process table
	8.	Smurf
	9.	teardrop
	10.	Udp storm
	11.	worm
Probe	1.	Ips weep
	2.	M scan
	3.	N map
	4.	Ports weep
	5.	saint
	6.	Satan
U2R	1.	Buffer over flow

	2.	Load module
	3.	Perl
	4.	ps
	5.	rootkit
	6.	Sql attack
	7.	X term
R2L	1.	ftp write
	2.	Guess passwd
	3.	http tunnel
	4.	I map
	5.	Multi hop
	6.	named
	7.	phf
	8.	Send mail
	9.	Snmp get attack
	10.	spy
	11.	Snmp guess
	12.	Warez client
	13.	Warez master
	14.	X lock
	15.	X snoop

Classification Algorithms

The machine algorithms implemented in this paper are described below:

Random Forest.

The random forest contains a significant number of decision trees that could perform classification individually and a most voted class is deemed as a prediction of the model. Using random forest, the accuracy can be improved as it utilizes the classification power of several trees. However, the key point is to form low correlated decision trees within the random forest. Otherwise, the error of the individual decision trees can add up and classify inaccurately. For this purpose, feature randomness and bagging to build uncorrelated decision trees that can provide high accuracy as given in Table 2.

Table 2. Model Parameters for RF

Parameters	No. of trees	Criterion	Maximal Depth
Values	100	Gini index	10

KNN.

Another commonly used machine learning algorithm is K-Nearest Neighbor that uses multiclass data to predict the class for a new sample. The classifier calculates the distance of the new sample point to all other existing points. It classifies the new sample point based on its closest neighbor in the dataset is given in Table 3.

Table 3. Model Parameters for KNN

Parameters	K
Values	5

Gradient Boost Tree (GBT).

Gradient Boost tree is an ensemble method that uses decision trees that are linked in series where every tree tries to minimize e the error generated by the previous three. Gradient boost tree is a greedy ML algorithm, so to reduce overfitting. Regularization methods are employed to penalize different components of the algorithm [16]. Even though, the

sequential algorithm takes longer to learn, it offers high accuracy for classification problems. The model parameters of GBT are shown in Table 4.

Table 4. Model Parameters for GBT

Parameters	No. of trees	Maximal Depth	Min rows	Learning rate	Sample rate
Values	50	10	10	0. 01	1.0

Decision Tree.

One of the supervised machine learning algorithms is a decision tree that is used for both classification and regression problems. The algorithm utilizes tree representation where a leaf node denotes a class label, while the attributes are signified on the tree's internal node (Table 5).

Table 5. Model Parameters for DT

Parameters	Criterion	Maximal Depth
Values	Gini index	10

Performance Matrices

The performance matrices used in this research include:

Accuracy.

Accuracy is the percentage of right predictions made after the model being tested. The accuracy of a classification model is determined based on its confusion matrix. It is used to obtain a general evaluation of the model given a balanced dataset.

Classification error.

It is the percentage of incorrect predictions made by a classifier where the incorrect predictions are the sum of true positives and false positives.

Weighted mean recall.

The recall is the measure of positive instances that are predicted as positively corrected. The weighted mean of recall is the average of recall with weights that are equal to a class's probability.

Weighted mean precision

Precision is measured to determine confidence in the performance of the applied model. It is the probability of correctly predicting a positive instance. Weighted mean precision considers the weight equal to the class probability into consideration.

Kappa.

Cohen's kappa is used to measure how closely instances classified by a machine learning algorithm are identical to the ground truth. The value for this statistic ranges from 0 to 1 where 0 represents total disagreement while 1 represents complete agreement. [17]. Generally, it is considered a more robust gauge as compared to basic percent agreement measurement.

Logistic Loss

It is the negative average of a log of accurately predicted probabilities and indicates the extent to which the prediction probability is similar to its respective actual value.

Root Mean Squared Error (RMSE).\

RMSE is an absolute measure of fit that indicates the success of the prediction of a model.

Auto model in Rapid Miner

RapidMiner Studio also facilities the accelerated method of developing and validating models through its extension known as the Auto Model. This feature can address three main problem classes: Prediction (classification and regression), clustering, and outlier detection. After the preprocessing and data mapping of the NSLKDD, Auto Model provides a selection of DT, RF, and GBT. Suitable parameters for each model were selected along with feature engineering and optimization techniques.

3. Results

The performance matrices values of each classifier are separately mentioned in Tables 6 to 7, for KNN, RF, GBT, and DT, respectively. The class precision and the recall for normal, DoS, R2L, probe, and U2R evaluated on KNN, RFF, GBT, and DT are illustrated in Figure 2 to Figure 3, respectively.

Table 6. Performance Evaluation of KNN

Performance matrices	Values for KNN
Accuracy (%)	98.70
classification error (%)	1.30
Weighted mean recall (%)	81.43
Weighted mean precision (%)	91.63
Kappa	0.978
Logistic loss	0.319
RMSE	0.1

	true normal	true DoS	true R2L	true Probe	true U2R	class precision
pred. normal	30577	45	64	82	30	99.28%
pred. DoS	59	21227	2	188	1	98.84%
pred. R2L	65	2	1471	11	5	94.66%
pred. Probe	117	81	15	5350	3	96.12%
pred. U2R	4	0	0	0	9	69.23%
class recall	99.21%	99.40%	94.78%	95.01%	18.75%	

Figure 2. Recall and Precision percentages for each class for KNN

Table 7. Performance Evaluation of RF

Performance matrices	Values for RF
Accuracy (%)	93.63
Classification error (%)	6.37
Weighted mean recall (%)	55.91
Weighted mean precision (%)	57.20
Kappa	0.889
Logistic loss	0.364
RMSE	0.256

	true normal	true DoS	true R2L	true Probe	true U2R	class precision
pred. normal	30700	1262	1513	672	48	89.78%
pred. DoS	8	20090	1	123	0	99.35%
pred. R2L	0	0	0	0	0	0.00%
pred. Probe	114	3	38	4836	0	96.89%
pred. U2R	0	0	0	0	0	0.00%
class recall	99.60%	94.08%	0.00%	85.88%	0.00%	

Figure 3. Recall and Precision percentages for each class for RF.

Table 8. Performance Evaluation of GBT

Performance matrices	Values for GBT
Accuracy (%)	99.42

Classification error (%)	0.58
Weighted mean recall (%)	89.28
Weighted mean precision (%)	93.39
Kappa	0.990
Logistic loss	0.456
RMSE	0.453

	true normal	true DoS	true R2L	true Probe	true U2R	class precision
pred. normal	34504	23	30	62	8	99.64%
pred. DoS	26	23989	2	27	0	99.77%
pred. R2L	47	0	1690	10	16	95.86%
pred. Probe	97	11	17	6234	2	98.00%
pred. U2R	0	1	7	2	28	73.68%
class recall	99.51%	99.85%	96.79%	98.41%	51.85%	

Figure 4. Recall and Precision percentages for each class for GBT.

Table 9. Performance Evaluation of DT

Performance matrices	Values for DT
Accuracy (%)	97.26
Classification error (%)	2.74
Weighted mean recall (%)	85.84
Weighted mean precision (%)	88.56
Kappa	0.954
Logistic loss	0.329
RMSE	0.163

	true normal	true DoS	true R2L	true Probe	true U2R	class precision
pred. normal	29997	22	44	6	0	99.76%
pred. DoS	104	21051	0	256	0	98.32%
pred. R2L	336	0	1464	125	24	75.12%
pred. Probe	385	282	39	5244	2	88.10%
pred. U2R	0	0	5	0	22	81.48%
class recall	97.32%	98.58%	94.33%	93.13%	45.83%	

Figure 5. Recall and Precision percentages for each class for DT

Figure 6 shows the comparison of classification accuracies for the four models.

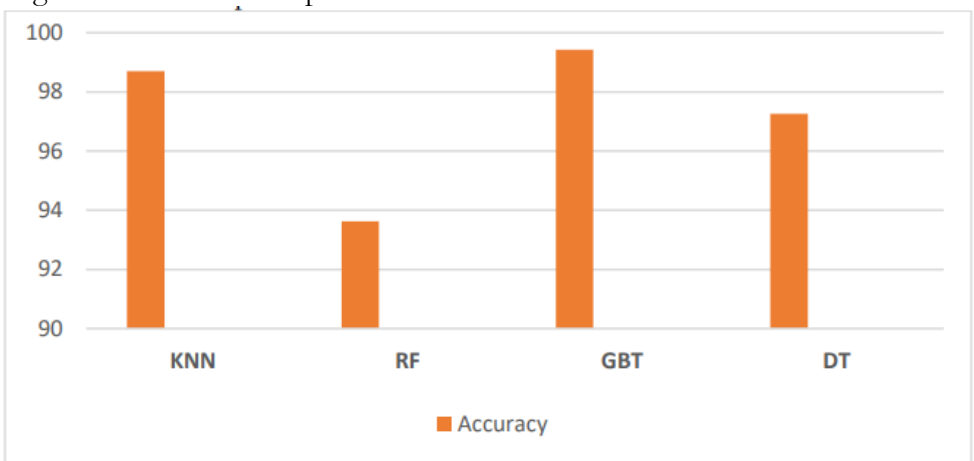

Figure 6. Comparison of accuracies for ML Algorithms

Figure 7 and 8 presents the class-wise comparison of precision and recall for ML models.

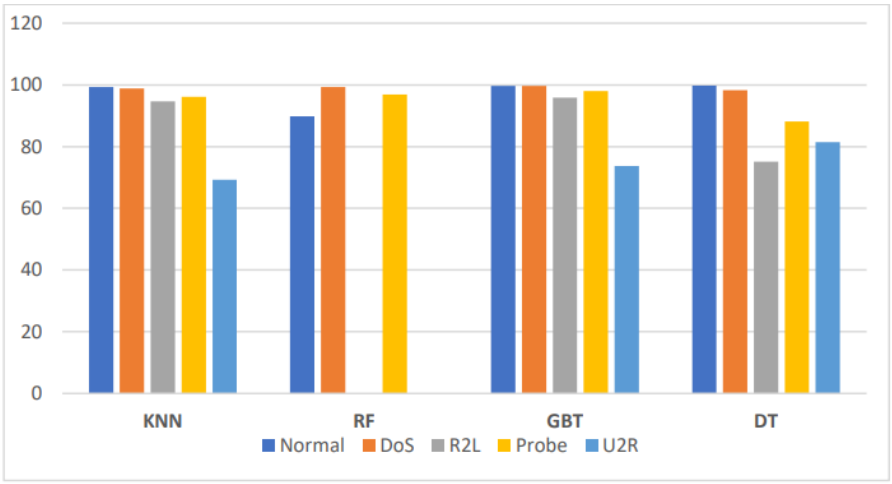

Figure 7. Comparison of Class wise Precision for ML Algorithms

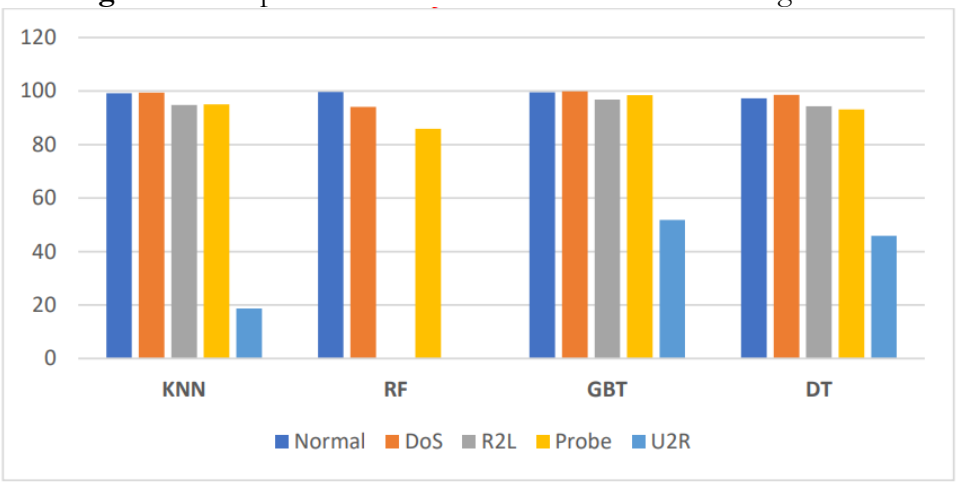

Figure 8. Comparison of Class wise Recall for ML Algorithms

Results from Auto Model

Figure 9 shows the overall performance of the three machine algorithms: DT, RF and GBT.

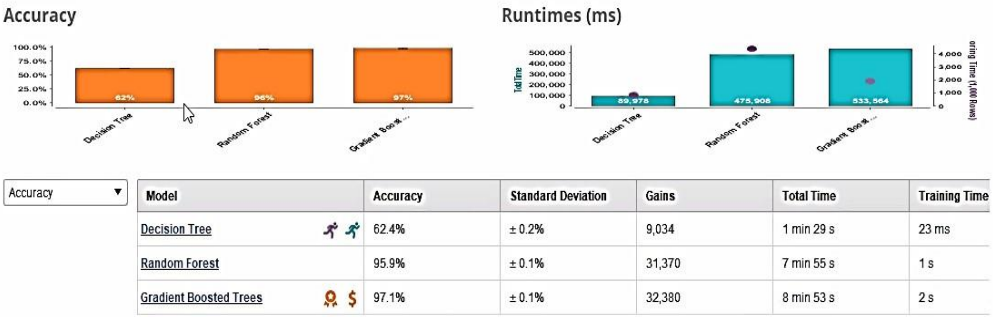

Figure 9. Overview of Classifier's performance in Auto Model

Figure 10 to 11 show the values of performance matrices and confusion matrix for DT, RF and GBT implemented in Auto Model.

Insights on recall and prediction for each class can be easily extracted from the confusion matrices generated by the Auto Model.

Decision Tree - Performance

Accuracy	62.4%	± 0.2%
Classification Error	37.6%	± 0.2%

Confusion Matrix

	true normal	true DoS	true R2L	true Probe	true U2R	class precision
pred. normal	20981	11111	663	1886	30	60.51%
pred. DoS	917	4105	24	1111	1	66.66%
pred. R2L	37	16	409	45	2	80.35%
pred. Probe	39	27	34	996	1	90.79%
pred. U2R	0	0	0	0	0	0.00%
class recall	95.48%	26.90%	36.19%	24.67%	0.00%	

Figure 10. Performance Matrices and Confusion Matrix for DT in Auto Model

Random Forest - Performance

Accuracy	95.9%	± 0.1%
Classification Error	4.1%	± 0.1%

Confusion Matrix

	true normal	true DoS	true R2L	true Probe	true U2R	class precision
pred. normal	17967	183	76	82	1	98.13%
pred. DoS	226	12345	18	54	1	97.64%
pred. R2L	193	254	834	149	20	57.52%
pred. Probe	185	4	6	3103	0	94.09%
pred. U2R	3	0	0	0	10	76.92%
class recall	96.73%	96.55%	89.29%	91.59%	31.25%	

Figure 11. Performance Matrices and Confusion Matrix for RF in Auto Model

Gradient Boosted Trees - Performance

Accuracy	97.1%	± 0.1%
Classification Error	2.9%	± 0.1%

Confusion Matrix

	true normal	true DoS	true R2L	true Probe	true U2R	class precision
pred. normal	18357	43	123	516	6	96.39%
pred. DoS	83	12817	2	41	1	99.02%
pred. R2L	40	0	779	110	9	83.05%
pred. Probe	25	4	12	2728	1	98.48%
pred. U2R	0	1	2	0	14	82.35%
class recall	99.20%	99.63%	84.86%	80.35%	45.16%	

Figure 12. Performance Matrices and Confusion Matrix for GBT in Auto Model

4. Discussions

For each model, accuracy, absolute error, weighted mean recall, weighted mean precision kappa values, logistic loss, and RMSE are calculated. The comparison of these values aids in evaluating the performance of the machine learning algorithm. The confusion matrix provides a summary of the prediction results for all the classes by a classification model.

Statistical Findings

The weighted mean recall for GBT (89.28%) is also the highest while the weighted mean recall for the RF is the lowest (55.91%). The weighted mean precision is highest in the

case of GBT (93.39%) while lowest for RF (57.20%). The value for Cohen's kappa coefficient (K) is also highest for GBT (0.990) but lowest for RF (0.889%). The highest logistic loss is observed in the GBT model (0.456) while KNN has the lowest logistic loss (0.319). The lowest RMSE is observed for GBT (0.453) while the KNN has generated the highest value of RMSE (0.1).

It has been observed that the DoS attack has been precisely identified by all the classifiers. However, the RF classifier did not identify the R2L or U2R attacks. The class wise precision and recall comparison for each machine learning algorithm is shown in Figure 3.6, and Figure 3.7, respectively. In addition to this, all the classifiers have low accuracy and precision score for the U2R class. It is potentially because only 2% of the dataset contains instances of R2l, U2R and PROBE make up 2% of the dataset collectively.

Comparing the accuracies for the machine learning algorithms, it is observed that the highest accuracy rate (99.42%) has been attained through the GBT model while RF provides the lowest accuracy (93.63%).

Findings from Auto Model

The models built on Auto Model show similar results as GBT has outperformed RF and DT with the highest accuracy of 97.1%. DT offers accuracy of 62.4% and classification error of 37.6%. Meanwhile, RF has an accuracy of 95.9% and classification error of 4.1%.

The model shows that DT has highest class precision for Probe (90.79%) and highest class recall for DoS (99.63%). RF has highest class precision for normal (98.13%) and highest class recall for normal (96.13%). GBT has highest class precision for Probe (98.48%) and highest class recall for normal (99.20%). The class precision and recall for U2R is lowest among all classes for all models, except for class precision for RF, where it is second lowest.

5. Conclusion And Future Recommendations

As a result of increased connectivity between computers, the implementation of intrusion detection has become vital for secure networks. Researchers have designed models like classification and clustering through machine learning algorithms like Naïve Bayes, logistic regression, RF, and SVM on NSL-KDD. The KDD dataset approximately includes 9% DOS attacks and 19% normal packets, while R2l, U2R, and PROBE make up 2% of the dataset collectively. This research paper discusses the classification performance of the four machine learning algorithms that include KNN, Random Forest, gradient boosted tree, and decision tree on the NSL-KDD dataset. Based on this research, GBT has outperformed all the other classification algorithms in the designs built and the Auto Model feature. GBT provided the highest accuracy (99.42%) while random forest algorithms achieved the lowest accuracy (93.63%). Moreover, it is found that it is more convenient to implement the machine learning models, especially on Rapid-Miner through Auto Model. This method is not only time-efficient and compact but also reduces the burden of implementing models via complex syntax. Different matrices including Accuracy, Absolute error, weighted mean recall, weighted mean precision, and Kappa are computed. All of these machine learning classifiers offer an accuracy on the NSL-KDD up to an acceptable extent. In the future, the latest available datasets like the CIC-Bell-DNS-EXF-2021 dataset can be used to evaluate the machine learning algorithms. Other ensemble models and deep learning algorithms can also be tested on the newest dataset.

References

[1] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," J. Netw. Comput. Appl., 2013, vol. 36, no. 1, pp. 16–24.

[2] T. T. Bhavani, M. K. Rao, and A. M. Reddy, "Network Intrusion Detection System Using Random Forest and Decision Tree Machine Learning Techniques," in First International Conference on Sustainable Technologies for Computational Intelligence, Singapore, 2020, pp. 637–643. doi: 10.1007/978-981-15-0029-9_50.

[3] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," Comput. Secur., 2009, vol. 28, no. 1–2, pp. 18–28.

[4] C. Zhang, F. Ruan, L. Yin, X. Chen, L. Zhai, and F. Liu, "A deep learning approach for network intrusion detection based on NSL-KDD dataset," in 2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID), 2019, pp. 41–45.

[5] M. A. Salama, H. F. Eid, R. A. Ramadan, A. Darwish, and A. E. Hassanien, "Hybrid Intelligent Intrusion Detection Scheme," in Soft Computing in Industrial Applications, Berlin, Heidelberg, 2011, pp. 293–303. doi: 10.1007/978-3-642-20505-7_26.

[6] A. Sultana and M. A. Jabbar, "Intelligent network intrusion detection system using data mining techniques," in 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), 2016, pp. 329–333.

[7] U. Fiore, F. Palmieri, A. Castiglione, and A. De Santis, "Network anomaly detection with the restricted Boltzmann machine," Neurocomputing, vol. 122, pp. 13–23, 2013.

[8] M. C. Belavagi and B. Muniyal, "Performance evaluation of supervised machine learning algorithms for intrusion detection," Procedia Comput. Sci., vol. 89, pp. 117–123, 2016.

[9] S. Thaseen and C. A. Kumar, "An analysis of supervised tree-based classifiers for intrusion detection system," in 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering, 2013, pp. 294–299.

[10] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," IEEE Trans. Emerg. Top. Comput. Intell., 2018, vol. 2, no. 1, pp. 41– 50.

[11] T. Ma, F. Wang, J. Cheng, Y. Yu, and X. Chen, "A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks," Sensors, 2016, vol. 16, no. 10, p. 1701.

[12] S. Hosseini and S. R. Sardo, "Data mining tools -a case study for network intrusion detection," Multimed. Tools Appl., 2021, vol. 80, no. 4, pp. 4999–5019, doi: 10.1007/s11042-020-09916-0.

[13] J. Arunadevi, S. Ramya, and M. R. Raja, "A study of classification algorithms using Rapidminer," Int. J. Pure Appl. Math., vol. 119, no. 12, pp. 15977–15988, 2018.

[14] K. Siddique, Z. Akhtar, F. A. Khan, and Y. Kim, "KDD cup 99 data sets: A perspective on the role of data sets in network intrusion detection research," Computer, 2019, vol. 52, no. 2, pp. 41–51.

[15] R. Bala and R. Nagpal, "A review on kdd cup99 and nsl nsl-kdd dataset.," Int. J. Adv. Res. Comput. Sci., 2019, vol. 10, no. 2.

[16] S. Si, H. Zhang, S. S. Keerthi, D. Mahajan, I. S. Dhillon, and C.-J. Hsieh, "Gradient boosted decision trees for high dimensional sparse output," in International Conference on Machine Learning, 2017, pp. 3182–3190.

[17] T. Wan, H. U. Jun, P. W. Hui ZHANG, and H. E. Hua, "Kappa coefficient: a popular measure of rater agreement ," Shanghai Arch. Psychiatry, 2015, vol. 27, no. 1, p. 62.