Detecting Generic Network Intrusion Attacks using Tree-based Machine Learning Methods

Yazan Ahmad Alsariera
Department of Computer Science, Faculty of Science
Northern Border University, Arar 73222, Kingdom of Saudi Arabia

***Abstract*—The development Intrusion Detection System (IDS) has a solid impact in mitigating against internal and external cyber threats among other cybersecurity methods. The machine learning-based method for IDS has proven to be an effective approach to detecting either anomaly or multiple classes of intrusion. For the detection of various types of intrusion by a single IDS model, it is discovered that the overall high accuracy of the IDS model does not translate to high accuracy for each attack type. Some intrusion attacks are seen to share similarities with other attacks thereby evading detection, one of which is the generic attack. The notoriety of the generic attack is the ability of a single generic attack to compromise a whole bunch of block-ciphers. Therefore, this study proposed a machine learning framework to specifically detect generic network intrusion by implementing two (2) decision tree algorithms. The decision tree methods were developed using two distinct variants namely the J48 and Random Tree algorithms. A balanced generic network dataset was curated and used for model development. A 10-fold cross-validation technique was implemented for model development and performance evaluation, where all obtainable performance scores were extracted and presented. The performances of the decision tree methods for generic network intrusion attack detection were comparative analysis and also evaluated against existing methods. The proposed methods of this study are robust, stable and empirically seen to have outperformed existing methods.**

Keywords*—Generic attack; decision trees; cybersecurity; intrusion detection

I. Introduction

The unprecedented surge of digital users over the years had led to the expansion of the world's cyberspace [1], [2]. Technological advancements had seen the enablement and rapid growth of various digital services offered to individuals and entities across the world [3]. Cyberspace consists of billions of connected devices and users whose security is now pivotal to the existence of the modern world [4], [5]. Cybersecurity emerges as the field that ensures the security of cyberspace.

Cybersecurity ensures data, information, and devices confidentiality, availability, and integrity against cyberspace attackers through sets of systems, technologies, and processes [6]. That means cybersecurity is responsible for providing countermeasures for removing and or ameliorating security threats and breaches (internal or external intrusion attacks) [7].

Before the execution of any known and unknown threat or attack, an attacker must first intrude (i.e. gain access to) his or her target network. This made the detection of intrusion a pivotal research area in cybersecurity [8]. The development of Intrusion Detection Systems (IDS) has received enormous research spotlight and the application of machine learning algorithms has proven to be the best method of developing effective and efficient IDS among other methods [9] – [13].

A recent review of the literature identified a problem that the effectiveness of machine learning (ML) based IDS for classifying multiple types of intrusion using a single model are hampered among network attacks with similar characteristics [14]. Hence, it becomes necessary to isolate and develop specific machine learning IDS for these types of extremely dangerous attacks. One such dangerous attack is called the 'Generic' attack. The generic attack is dangerous such that one (1) generic attack can attack all block-ciphers regardless of the distinct structure of the ciphers [15], [16].

Despite known to be dangerous, countermeasures against generic intrusion are not well researched and developed in the context of applying ML algorithms. Generic network intrusions are not captured by KDDCup'99 and NSL-KDD intrusion network dataset [1]. However, the comprehensive and contemporary dataset published by [16] contains generic attack traffics. Even so, this dataset [16] is usually used for developing anomaly (i.e. normal and attack) [17] and multi-class (i.e. normal and nine (9) other attacks) IDS [18]. Hence, this study is motivated and thereby proposes an ML-based IDS framework specifically for detecting generic network intrusion attack. The contributions to knowledge made by this research are highlighted below:

1) Development of a balanced network intrusion 'Generic' attack dataset for machine learning classification process; and

2) Implementation and performance evaluation of two (2) distinct machine learning decision tree algorithms as the proposed methods for detecting generic network intrusion.

The decision trees were selected as they are seen to have a sharp distinction between their methods of learning, unlike other decision tree algorithms. More so, through this research work, answers to the following research questions are sought:

1) How well can J48 and Random Tree decision tree algorithms effectively detect generic attacks?

2) Is there any significant difference(s) in using distinct variants of decision tree algorithms for detecting generic network intrusion?

3) How good is the performance of the proposed method against related existing methods?

The proposed methods of this study, whose application lies in network security, will serve as a customized IDS for detecting generic network intrusion. The remaining part of this study is structured as follow: Section II contains a review of related works, Section II vividly reveals the method (i.e. dataset, implemented models and performance evaluation metrics), Section IV present results and discussions and finally Section V shares the conclusion and future works.

II. Review of Related Works

Although stand-alone researches on generic network intrusion detecting are very scarce, some multi-classification researches on IDS present a breakdown of their model's performances. The type of research studies that made these provisions as well as other closely related studies was sought and reviewed accordingly.

The research work of [19] presented an ensemble of sophisticated deep learning algorithms for detecting different types of network anomalies. The study implemented a majority voting ensemble of three hyper-parameter long-short-term memory deep neural network with an embedded feature extraction module. The feature extraction module composed of a Genetic Algorithm (GA). This algorithmic framework was implemented and fitted on NSL-KDD and UNSW-NB15 datasets. The published method reported an overall accuracy of 99.9%. However, the performance of the implemented framework for the detection of a generic attack dropped to 95.23% without feature selection and 97.31% with feature selection. This supports the need to develop a specified generic network IDS method with increased accuracy and lower false alarm rate.

Another study [15], presented a novel integrated rule-based multi-classification method IDS fitted on the UNSW-NB15 dataset. The proposed method is a misuse-based IDS for four type of attacks namely: DOS, Generic, Exploit, Probe and the Normal traffic in a network. The proposed method achieved an overall Average accuracy (i.e. AvgAcc) of 65.21% for all classes of attacks and a False Alarm Rate of 2.01%. From the study, an improved IDS is generally required even to detect other types of network intrusion.

A more recent study [20] published a stacked ensemble method for developing a multi-classifying IDS. Three (3) methods for stacking base classification algorithms were implemented namely: Meta Decision Tree (MDT), Multiple Model Trees (MMT) and Multi-Response Linear Regression (MLR). The base classifiers are Naïve Bayes, Decision Tree and K-Nearest Neighbour. The evaluation of the base learner (DT) for classifying all attack type achieved an overall accuracy of 75.71% without feature selection. The MMT ensemble method produced 96.89% overall accuracy, the MDT ensemble method had 98.08% and the MLR method had 97.8% accuracies based on the correlated reduced feature selected model. The performance of the method for detecting generic network intrusion was not disclosed.

Gharaee & Hosseinvand [21] reportedly developed a new feature selection IDS using the support vector machine algorithm and a genetic algorithm for feature selection which was referred to as "GF-SVM". The genetic feature selection algorithm was reportedly developed using a novel fitness function that was responsible for dimensionality reduction. The overall performance of the multi-classifier IDS was broken down and presented for each class of attack. The implemented method was able to achieve an accuracy of 97.51%, 96.69% True Positive (TP) rate, and 0.01% False Positive (FP) rate for the 'Generic' attack as related to this study. The rate at which generic network intrusion can be detected (TP) by [21] can be further improved while the FP rate can be lowered which is the intention of this study.

Succinctly, the review of related literature that provides the performance breakdown of the existing method further strengthen the need for developing stand-alone generic network intrusion attack detectors.

III. Method

In this section, details of the dataset of the study are presented as well as the machine learning algorithms used to implement the generic detector IDS and the performance evaluation metrics for the implemented models.

A. Dataset

Dataset serves as a core part of empirical research. Therefore, it is important to make use of the dataset that truly serves the study's aim, strengthens the study as well as being state-art-of-the-art. In this study, the development of ML decision trees methods for detecting 'generic' network intrusion attack is crucial. Therefore, a state-of-the-art dataset is used to conduct the study's experiment. In the research scope of developing ML methods for network intrusion detection, the UNSW-NB15 dataset is currently the best benchmarking and the openly available dataset [16]. This dataset ousted other public network attack datasets (i.e. KDDCup'99 and NSL-KDD) by providing contemporary network traffic and attacks [16].

The KDDCup'99 is the initial benchmarking dataset but was revised and led to the production of the NSL-KDD dataset. The NSL-KDD data is devoid of all redundancy in its predecessor and provides a more balanced dataset [9], [22]. However, it does not contain contemporary attacks as executed by attackers' such as the 'Generic' attack type that is been studied in this research work. More so, attackers daily carry out dynamic attacks which then require developing intelligent countermeasures from a contemporary dataset (having real and or synthetic attacks) to adequately ameliorate novel malicious network activities [23]. KDDCup'99 and NSL-KDD are not reflective of contemporary attacks and network packets, which single-out and justifies the usage of the UNSW-NB15 dataset by this study.

As mentioned in the introduction section, a high-performing multi-classifier does not usually achieve single-class discrimination when two or more class shares similar feature values [14]. Therefore, to achieve the aim of this study, the 'Generic' attack instances were extracted from the UNSW-NB15 dataset alongside adequate 'Normal' instances to create a balanced dataset. Table I gives insights into the dataset used in this study.

The original dataset contains forty-five (45) variables which were reduced to forty-three (43) in the newly created dataset (i.e. without the 'id' and 'attack_cat' variables). Forty-two (42) of all variables serves as independent variables and the forty-third (43rd) variable is the dependent variable with two values as shown in Table I. From the original dataset, there are 18,871 'Generic' attack instances. As such, 18,954 normal instances were extracted to create the benchmark dataset for this study.

B. Implemented Models

In this study, two (2) distinct variants of the decision tree (DT) machine learning algorithms were used to fit the required models for detecting generic network intrusion attacks.

Decision tree algorithms are a family of machine learning classification and regression algorithms that fits a model on a given dataset having considered the entropy of some or all attributes for making its splitting decision. Tree-based machine learning algorithms are widely used and acceptable for various research and industrial areas, even as distant as software defect prediction in the field of software engineering [24] and even for the prediction of factors in educational management [25]. Decision Tree models are known to always produce interpretable models. Additionally, the derived tree inherent in every decision tree model can be used as a rule(s) for guiding expert decision aside from its usage for prediction.

Fundamentally, all decision tree algorithm can perform both regression and classification (primarily binary classification) analyses. Decision algorithms usually fit its model through a greedy top-down method which is performed recursively on the dataset to find the most informative variable at each split decision junction [25]. Additionally, it may also include a method for producing a fine-tuned tree by the way of pruning the initial tree based on the error rate thereby removing redundant branches [26]. All decision tree algorithms begin the process of fitting a model with a root node (which is the most informative variable) and then create branches and some leaves downwardly based on the results of testing variables values Extracted from [26].

Pseudocode 1: A typical Decision Tree Algorithm.

```
1:   Create a root node R;
2:   IF (W belongs to same category C)
       {leaf node = R;
         Mark R as class C;
         Return R;
       }
3:   For i=1 to R
       {Calculate Information_gain (Ai);}
4:   ta = testing attribute;
5:   R.ta = attribute having highest information_gain;
6:   If(R.ta == continuous)
       {find threshold;}
7:   For (Each W in splitting of W)
8:     If (W is empty)
         {child of R is leaf node;}
         else
         {child of R= dtree W;}
9:   calculate the classification error rate of node R
10:  return R;
```

TABLE I. TABLE TYPE STYLES

Dataset Description		
No of Attributes		43
No. of Independent Variables		42
Dependent Feature values distribution	***Values***	
	Generic	***Normal***
	18,871	18,954

As mentioned earlier, two (2) machine learning decision trees algorithms variants were considered in this study. These are the famous J48 and Random Tree algorithms.

J48 algorithm is usually a greedy top-down approach starting from the root node through the branches down to the leaves. It can also follow a bottom-up approach. It contains decision nodes (branches) which are indicators to tested attributes and leaves which signifies class values. J48 is characterized by its ability to accept both nominal and continuous variable values. Also, it includes an imputation technique that resolves missing values in variables as well as a pruning mechanism for developing optimal but small trees that avoid over-fitting [26]. In this study, the J48 algorithm was implemented and fitted on the described dataset. The resulting model was evaluated using all obtainable performance evaluation metrics.

On the other hand, Random Tree is another variant of the decision tree algorithm family that fits various decisions trees on a given dataset using N randomly selected variables at each node. These sets of random decision trees usually form a uniform distribution which gives each tree an equal sampling chance. These uniformly distributed trees are used to develop a random tree through aggregation which produce a more robust and accurate model. In this study, the Random Tree algorithm was implemented and fitted on the dataset producing a model which was subjected to the evaluation of its performance in discriminating between 'Generic' intrusion attack and normal network traffic.

The experimental framework of this study is graphically depicted in Fig. 1 which illustrate how the data preprocessing and processing, the selected machine learning decision trees methods were developed and their respective performance evaluation.

The decision tree methods, namely J48 and Random tree decision tree algorithms, were implemented and fitted on the randomly shuffled dataset through the 10-fold cross-validation technique. The cross-validation technique is the method of fitting a robust model by splitting the dataset into user-defined value – 10 partitions. It trains the model using the first 9 splits and test on the set-aside split. This is repeated 10 times until all splits are used for training and testing. The 10 models are then aggregated to produce a robust model. The performances of the fitted models (i.e. J48 and Random Tree generic attack detectors) were measure and evaluated using widely acceptable metrics, such as confusion matrix, MCC, accuracy, True positive, True negative, kappa score and others as previously mentioned.

C. Performance Evaluation Metrics

This section discusses how the performance of the proposed ML decision trees methods for detecting generic network intrusion attack was evaluated. The models can be referred to as binary classification (i.e. two class values) methods. Thus, our proposed methods were evaluated by populating and reporting their respective performance values using the confusion matrix. More so, other performance values, which can be derived from the confusion matrix, were also reported. These are: TP Rate (i.e. Detection Rate), FP Rate (i.e. False Alarm Rate), Precision, Recall, F-Measure, Matthews Correlation Coefficient (MCC), Area Under Curve (AUC) [2], [10], [19], [27]–[30]. Also, the overall accuracy (i.e. the percentage of correctly classified 'Generic' attack and normal network traffic), as well as kappa value, were obtained for each method.

For emphasis, the MCC metric is arguably the prime metric for evaluating a binary classification as it based on all four values of the confusion matrix [19], [31]. It reveals the correlation coefficient among the detected and expected predictions, having a value ranging from 0 to 1 [30]. Therefore, a better gauge of the classification model is revealed in the MCC value. However, this does not relegate other performance metrics. MCC metric is calculated as seen in Equation 1.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (1)$$

Matthews Correlation Coefficient extracted from [31].

As illustrated in the proposed empirical framework presented in Fig. 1, all 'Generic' attack network instances from the UNSW-NB15 dataset were extracted. Additionally, enough normal network instances were also extracted to create a balanced dataset that serves as input to the decision tree methods. Before inputting the balanced dataset, it was shuffled to ensure instances of both class values were properly mixed and the model can learn from the distribution simultaneously.

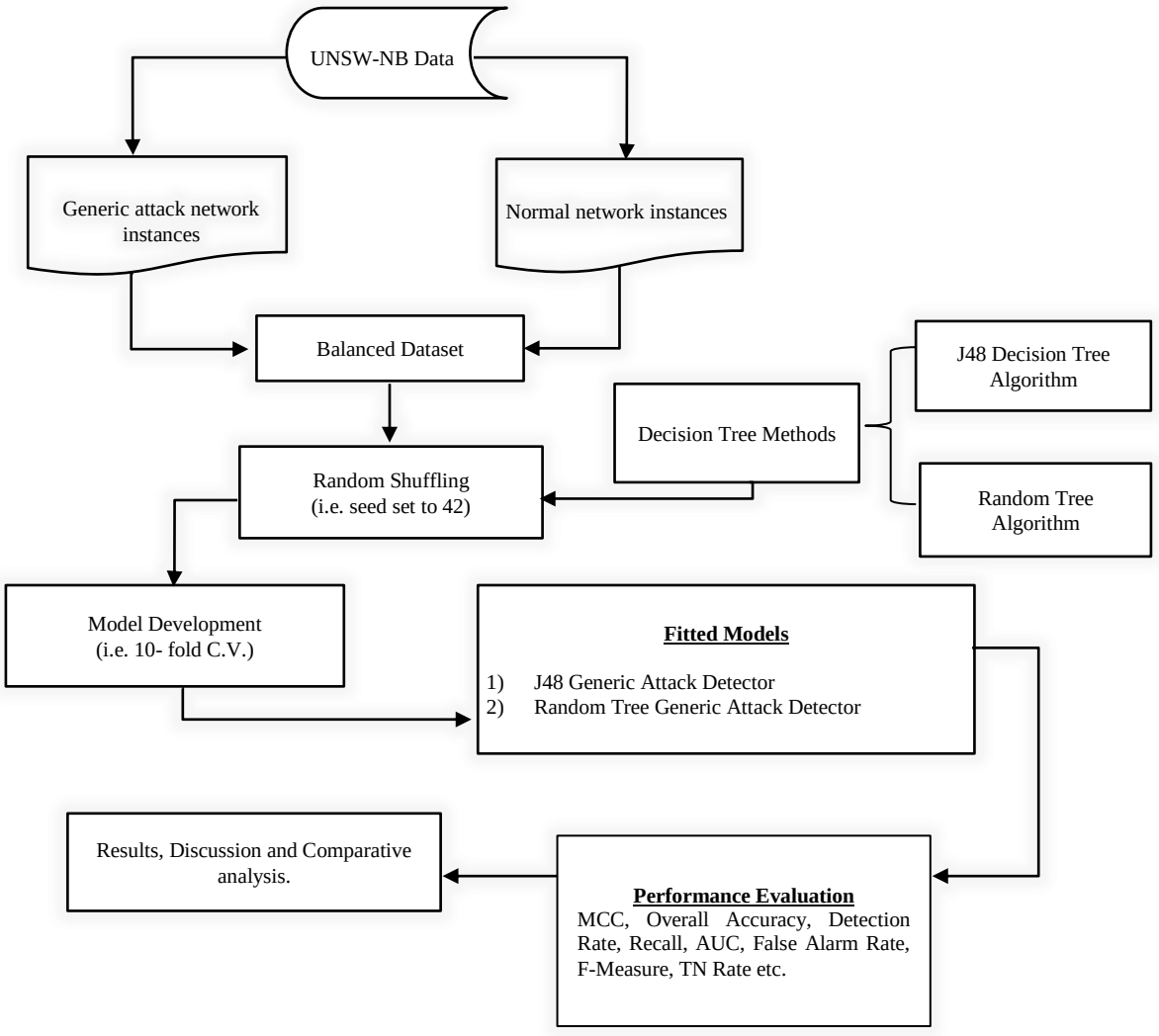

Fig. 1. Experimental Framework.

IV. Results and Discussion

This section reports the performance of the proposed ML decision trees methods in tables and charts. More so, the results are being discussed individually and a comparative analysis of the performance of the proposed methods against existing methods is provided by answering the aforementioned research questions.

A. Results

As previously discussed, the dataset has a total number of 18,871 generic instances and 18,954 normal instances. Also, the proposed methods were developed using a 10-fold cross-validation technique for classification model development.

The J48 'Generic' attack classifier performance confusion matrix is presented in Table II and other derived performance evaluation values are presented in Table III.

As seen in Table II, there were 18,830 correctly classified instances of 'generic' attack out of the total of 18,871. Similarly, there were 18,921 correctly classified normal instances out of the total of 18,954. A total of 41 generic attack instances were misclassified as normal while a total of 33 normal instances were falsely classified as generic.

The values of other derived performance measures are revealed in Table III.

From Table III, the proposed J48 classifier achieved an overall accuracy of 99.804% - an excellent performance. This is evident as revealed in the confusion matrix in Table II. Additionally, this model scores a kappa value of 0.9961 indicate the model performed higher than chance. The model achieved the TP and TN rates of 0.998 respectively while it also had FP and FN rates of 0.002 respectively. Its precision, f-measure and recall values also tallied at 0.998. Lastly, it had an AUC score of 0.999 while it had an MCC score of 0.996.

TABLE II. J48 Generic Attack Discriminator Confusion Matrix

	Generic	***Normal***
Generic	18,830	41
Normal	33	18,921

TABLE III. J48 Generic Attack Discriminator Evaluation

Evaluation Metric	**J48's Performance Value**
Accuracy	99.804%
Kappa	0.9961
TP Rate (Detection Rate)	0.998
FP Rate	0.002
TN Rate	0.998
FN Rate	0.002
Precision	0.998
Recall	0.998
F-measure	0.998
MCC	0.996
AUC	0.999

The performance of the model obtained after fitting the Random Tree algorithm on the dataset via 10-fold cross-validation was also evaluated just like its counterpart. Table IV presents the confusion matrix for the Random tree classifier.

From Table IV, 18,776 of 18,871 generic attack instances were correctly classified while 18,883 of 18,954 normal instances were also correctly classified. 95 generic instances were misclassified as normal traffic while 71 normal instances were misclassified as a generic attack.

Additionally, other performance values were derived and depicted in Table V.

This Random tree proposed method achieved an overall accuracy of 99.561% and a kappa score of 0.9912. It obtained a TP rate of 0.995, TN rate of 0.996, FP Rate of 0.004, and FN Rate of 0.005. More so, it had a precision, recall and f-measure score tallied at 0.996 respectively. The classifier scored an AUC value of 0.997 while having a 0.991 MCC score.

B. Discussion

This study aims to develop a machine learning framework specifically capable of detecting the extremely dangerous generic network intrusion attack which shares similarity with other types of attack thereby evades detection. Following the implemented of the proposed framework, the two generic network intrusion attack detectors were robustly developed and evaluated using the 10-fold cross-validation technique. Two algorithmically distinct decision tree methods were developed, and all obtainable performance evaluation scores were derived from the confusion matrix obtained produced by each of the methods.

TABLE IV. Random Tree Generic Attack Discriminator Confusion Matrix

	Generic	***Normal***
Generic	18,776	95
Normal	71	18,883

TABLE V. Random Tree Generic Attack Discriminator Evaluation

Evaluation Metric	**RT's Performance Value**
Accuracy	99.561%
Kappa	0.9912
TP Rate (Detection Rate)	0.995
FP Rate	0.004
TN Rate	0.996
FN Rate	0.005
Precision	0.996
Recall	0.996
F-measure	0.996
MCC	0.991
AUC	0.997

A comparative analysis of both proposed methods empirical results reveals that the model produced after fitting the J48 decision tree algorithm is insignificantly better than Random Tree's model as presented in Table VI.

Table VI present the empirical results of the methods using four benchmark performance metrics out of all performance metrics mentioned in the performance evaluation section. The J48 DT method is seen to produce an overall accuracy of 99.8% while the Random Tree DT method produced an overall accuracy of 99.6%. Similar trends are recorded for the detection rate (i.e. TP) and the False Alarm Rate (i.e. FP). J48 method detected a 'Generic' attack at 99.8% while Random Tree did the same at 99.5%.

More so, both methods were able to detect generic network intrusion attack with an extremely low false alarm rate. J48% false alarm rate is 0.002% while the Random Tree method's rate is at 0.004%. Both decision tree methods proved to be a viable method for detecting a generic attack. This summarized comparative analysis is depicted in Fig. 2.

C. Comparative Analysis with Existing Methods

The answers to this study's research questions, which also facilitate comparative analysis of the proposed methods even with existing methods, are provided in this section. The first research question is about the effectiveness of the machine learning decision tree (i.e. J48 and Random Tree) methods. As seen through the empirical results, the effectiveness of these methods for detecting generic network cannot be over-emphasized. Both methods are excellently effective at a detection rate not lower than 99% and with an incredible false alarm rate lower than 0.05%. The MCC scores of both methods were also not lower than 0.99 as well as their precision, recall, f-measure and ROC values. All these results indicate that both J48 and Random Tree generic network intrusion attack detector are highly effective.

TABLE VI. EMPIRICAL RESULTS

Decision Tree Algorithms	*Accuracy (%)*	*Detection Rate (%)*	*False Alarm Rate*	*MCC*
J48	99.8	99.8	0.002	0.996
Random Tree	99.6	99.5	0.004	0.991

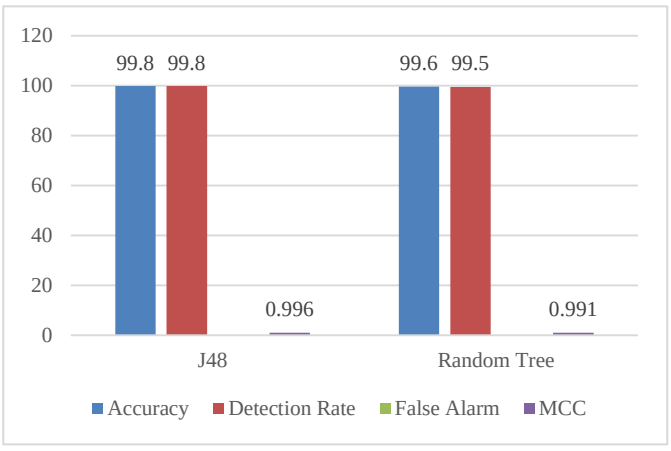

Fig. 2. Summative Comparative Analysis of Proposed Methods.

The second research question aims to investigate the comparative performance of the implemented distinct variants of the decision trees method. Also, the empirical results indicate that the methods do not significantly outperform each other. Since all other variants of the decision tree algorithms are closely related to one of these distinct variants, it is safe to infer that any decision tree method implemented for detecting generic attack will perform similarly to the proposed methods of this study.

Lastly, the answer to the third research question which is the comparative analysis of the proposed method against the existing method is provided. The published sophisticated genetic algorithm and deep-learning method [19] reported a 95.23% overall accuracy for detecting generic attack with feature selection. This reported performance [19] which is lower compared to the overall accuracy for the said method for multi-classification, is also lower than the performance of this study's proposed methods for generic network intrusion detection.

Also, the decision tree method published by [20] achieved a 75.71% generic attack detection accuracy without feature selection while its stacked ensemble methods on the correlation reduced models produced 97.8%, 96.89% and 98.08% accuracies. All these models were outperformed by the proposed methods of this study as this study's methods had at least a 99% detection rate.

Additionally, the novel integrated rule-based IDS [15] for detecting DOS, Generic, Exploit, Probe attacks and the Normal traffic in a network had an overall AvgAcc of 65.21% for all classes of attacks and a False Alarm Rate of 2.01% which is comparatively lower than the performance of this study's method even if broken down into different attack types. The existing (i.e. multi-classification) methods are low-performing machine learning methods for detecting generic network intrusion attack which justifies the importance of this research.

V. CONCLUSION AND FUTURE WORKS

This study proceeds to develop tree-based machine learning generic network intrusion detection models, having identified the problem that generic attack shares similarities with other attacks and usually evades detection from multi-classification IDS. Two (2) distinct tree-based machine learning method, J48 and Random Tree algorithms were proposed to implement this study's models.

J48 model was able to detect generic network attack at 99.8% and a false alarm rate of 0.002 while the Random Tree model detected generic network attack at a 99.6% detection rate and a false alarm rate of 0.004. The comparative analysis of the proposed methods against existing methods which are mostly multi-classification IDS reveals that the proposed method performed better than all of them in detecting generic network intrusion.

In the future, the application of other types or families of machine learning classification method will be explored. More so, the culling out of important feature (i.e. reducing the dimensionality) from the original feature space of this balanced generic attack dataset will be considered.

REFERENCES

[1] Y. Xin et al., "Machine Learning and Deep Learning Methods for Cybersecurity," IEEE Access, vol. 6, pp. 35365–35381, 2018.

[2] O. A. Sarumi, A. O. Adetunmbi, and F. A. Adetoye, "Discovering computer networks intrusion using data analytics and machine intelligence," Sci. African, vol. 9, p. e00500, 2020.

[3] H. N. Thanh and T. Van Lang, "Evaluating Effectiveness of Ensemble Classifiers When Detecting Fuzzers Attacks on the Unsw-Nb15 Dataset," J. Comput. Sci. Cybern., vol. 36, no. 2, pp. 173–185, 2020.

[4] P. Kumar, G. P. Gupta, and R. Tripathi, "A distributed ensemble design based intrusion detection system using fog computing to protect the internet of things networks," J. Ambient Intell. Humaniz. Comput., no. 0123456789, 2020.

[5] O. Faker and E. Dogdu, "Intrusion detection using big data and deep learning techniques," ACMSE 2019 - Proc. 2019 ACM Southeast Conf., pp. 86–93, 2019.

[6] M. R. Gauthama Raman et al., An efficient intrusion detection technique based on support vector machine and improved binary gravitational search algorithm, vol. 53, no. 5. Springer Netherlands, 2020.

[7] V. Dutta, M. Choraś, R. Kozik, and M. Pawlicki, "Hybrid model for improving the classification effectiveness of network intrusion detection," in Complex, Intelligent, and Software Intensive Systems, 2020, vol. Springer, pp. 405–414.

[8] W. Wei, S. Chen, Q. Lin, J. Ji, and J. Chen, "A multi-objective immune algorithm for intrusion feature selection," Appl. Soft Comput. J., vol. 95, p. 106522, 2020.

[9] M. A. Mabayoje, A. O. Balogun, A. O. Ameen, and V. E. Adeyemo, "Influence of Feature Selection on Multi-Layer Perceptron Classifier for Intrusion Detection System," Comput. Inf. Syst. Dev. Informatics Allied Res. J., vol. 7, no. 4, pp. 87–94, 2016.

[10] A. V. Elijah, A. Abdullah, N. Z. JhanJhi, M. Supramaniam, and A. O. Balogun, "Ensemble and deep-learning methods for two-class and multi-attack anomaly intrusion detection: An empirical study," Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 9, pp. 520–528, 2019.

[11] A. O. Balogun, A. M. Balogun, V. E. Adeyemo, and P. O. Sadiku, "A Network Intrusion Detection System : Enhanced Classification via Clustering," Comput. Inf. Syst. Dev. Informatics Allied Res. J., vol. 6, no. 4, pp. 53–58, 2015.

[12] P. Illy, G. Kaddoum, C. M. Moreira, K. Kaur, and S. Garg, "Securing Fog-to-Things Environment Using Intrusion Detection System Based On Ensemble Learning," no. April, pp. 15–18, 2019.

[13] M. Idhammad, K. Afdel, and M. Belouch, "Semi-supervised machine learning approach for DDoS detection," Appl. Intell., vol. 48, no. 10, pp. 3193–3208, 2018.

[14] T. Salman, D. Bhamare, A. Erbad, R. Jain, and M. Samaka, "Machine Learning for Anomaly Detection and Categorization in Multi-Cloud Environments," Proc. - 4th IEEE Int. Conf. Cyber Secur. Cloud Comput. CSCloud 2017 3rd IEEE Int. Conf. Scalable Smart Cloud, SSC 2017, pp. 97–103, 2017.

[15] V. Kumar, D. Sinha, A. K. Das, S. C. Pandey, and R. T. Goswami, "An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset," Cluster Comput., vol. 23, no. 2, pp. 1397–1418, 2020.

[16] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in 2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings IEEE, 2015, pp. 1–6.

[17] F. Feng, X. Liu, B. Yong, R. Zhou, and Q. Zhou, "Anomaly detection in ad-hoc networks based on deep learning model: A plug and play device," Ad Hoc Networks, vol. 84, pp. 82–89, 2019.

[18] M. Nawir, A. Amir, N. Yaakob, and O. N. G. B. I. Lynn, "Multi-Classification of Unsw-Nb15 Dataset for Network Anomaly Detection System," J. Theor. Appl. Inf. Technol., vol. 96, no. 15, pp. 5094–5104, 2018.

[19] I. S. Thaseen, A. K. Chitturi, F. Al‐Turjman, A. Shankar, M. R. Ghalib, and K. Abhishek, "An intelligent ensemble of long‐short‐term memory with genetic algorithm for network anomaly identification," Trans. Emerg. Telecommun. Technol., no. September, pp. 1–21, 2020.

[20] O. O. Olasehinde, "A Stacked Ensemble Intrusion Detection Approach for the Protection of Information System," Int. J. Infomation Secur. Res., vol. 10, no. 1, pp. 910–923, 2020.

[21] H. Gharaee and H. Hosseinvand, "A new feature selection IDS based on genetic algorithm and SVM," 2016 8th Int. Symp. Telecommun. IST 2016, pp. 139–144, 2017.

[22] A. Salih, X. Ma, and E. Peytchev, "Detection and Classification of Covert Channels in IPv6 Using Enhanced Machine Learning," 2015.

[23] G. Li and Z. Yan, "Data Fusion for Network Intrusion Detection : A Review," vol. 2018, 2018.

[24] A. O. Balogun et al., "SMOTE-Based Homogeneous Ensemble Methods for Software Defect Prediction," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 12254 LNCS, pp. 615–631, 2020.

[25] W. Gata et al., "Prediction of Teachers' Lateness Factors Coming to School Using C4.5, Random Tree, Random Forest Algorithm," vol. 258, no. Icream 2018, pp. 161–166, 2019.

[26] S. Aljawarneh, M. B. Yassein, and M. Aljundi, "An enhanced J48 classification algorithm for the anomaly intrusion detection systems," Cluster Comput., vol. 22, no. 5, pp. 10549–10565, 2019.

[27] S. M. Kasongo and Y. Sun, "A deep learning method with wrapper based feature extraction for wireless intrusion detection system," Comput. Secur., vol. 92, 2020.

[28] Y. A. Alsariera, A. V. Elijah, and A. O. Balogun, "Phishing Website Detection: Forest by Penalizing Attributes Algorithm and Its Enhanced Variations," Arab. J. Sci. Eng., vol. 45, no. 12, pp. 10459–10470, 2020.

[29] Y. A. Alsariera, V. E. Adeyemo, A. O. Balogun, A. K. Alazzawi, "AI Meta-Learners and Extra-Trees Algorithm for the Detection of Phishing Websites," IEEE Access, vol. 8, pp. 142532-142542, 2020.

[30] J. O. Mebawondu, O. D. Alowolodu, J. O. Mebawondu, and A. O. Adetunmbi, "Network intrusion detection system using supervised learning paradigm," Sci. African, vol. 9, 2020.

[31] N. Li, M. Shepperd, and Y. Guo, "A Systematic Review of Unsupervised Learning Techniques for Software Defect Prediction," 2019.