Performance Comparison of Ensemble Learning and Supervised Algorithms in Classifying Multi-label Network Traffic Flow

Mwita Machoke
School of Computational and Communication Science and
Engineering, Department of Information Technology System
Development and Management, NM-AIST, Arusha, Tanzania
machokem@nm-aist.ac.tz

Jimmy Mbelwa
University of Dar es Salaam
Dar es Salaam, Tanzania
jimmymbelwa@gmail.com

Johnson Agbinya
School of Information Technology and Engineering
Melbourne Institute of Technology
Melbourne, Victoria, Australia
jagbinya@mit.edu.au

Anael Elikana Sam
School of Computational and Communication Science and
Engineering, Department of Communication Science and
Engineering, NM-AIST Arusha, Tanzania
anael.sam@nm-aist.ac.tz

Received: 16 February 2022 | Revised: 21 March 2022 | Accepted: 22 March 2022

Abstract-Network traffic classification is of significant importance. It helps identify network anomalies and assists in taking measures to avoid them. However, classifying network traffic correctly is a challenging task. This study aims to compare ensemble learning methods with normal supervised classification to come up with improved classification methods. Three types of network traffic were classified (Benign, Malicious, and Outliers). The data were collected experimentally by using Paessler Router Traffic Grapher software and online and were analyzed by R software. The datasets were used to train five supervised models (k-nearest neighbors, mixture discriminant analysis, Naïve Bayes, C5.0 classification model, and regularized discriminant analysis). The models were trained by 70% of the samples and the rest 30% were used for validation. The same samples were used separately in predicting individual accuracy. The results were compared to the ensemble learning models which were built with the use of the same datasets. Among the five supervised classifiers, k-nearest neighbors and C5.0 classification scored the highest accuracy of 0.868 and 0.761. The ensemble learning classifiers Bagging (Random Forest) and Boosting (eXtreme Gradient Boosting) had accuracy of 0.904 and 0.902 respectively. The results show that the ensemble learning method has higher accuracy compared to the normal supervised classifiers. Therefore, it can be used to detect malicious activities in network traffic as well as anomalies with improved accuracy.

Keywords-ensemble; malicious; anomalies; security

I. INTRODUCTION

The rapid development of Information and Communication Technologies (ICTs) including hardware, the Internet, data science techniques, and services such as online transactions, edge, and cloud computing has changed the way many societies communicate, work, and learn [1]. Because of the variety of computer software available as well as the growing number of users, large-volume data processing has become more complex. The trends show that an increase in mobile data traffic will exceed zettabyte (200^{10}) by 2025 [2]. The total network traffic pattern has increased exponentially the last few years [3, 4]. Furthermore, it has also been reported that such rapid growth in data traffic is likely to cause security breaches, risks, and lowering of network performance, especially in communication network systems [5, 6]. It also provides room for potential demand for re-designing network architectures to overcome security threats [7]. Through the use of the Internet, there is a possibility of security attacks occurring at any given time in the communication network system including software, hardware, and attached accessories or communication devices.

Data flows from one point to another as unidirectional packets in Internet Protocol (IP) communication networks. The flows depend on the hardware performance and network architecture [8]. A flow is a traffic stream with a common set of identifiers that has the same source IP, destination IP, protocol, source, and destination ports [9]. Monitoring data traffic in connected devices provides useful information that would be of importance in the timely understanding of the behavior of the flows and in predicting bandwidth usage. Monitoring data flows is crucial in that, any Denial of Service attacks (DoS) and other network security threats and vulnerabilities within the network can be easily identified for timely interventions [5, 10-12]. It helps system administrators and security experts to understand and monitor all activities in the given computer network.

High online service demand is embedded in our daily life. Detecting anomalies in the network can be very difficult [13].

Corresponding author: Mwita Machoke

It is very difficult to detect, identify and prevent malicious activities in computer networks, especially in a distributed computing environment. Priority is given to the process of identifying the characteristics of applications that generate high traffic due to malicious activities in communicating devices. These facts reveal that network traffic management and monitoring for the smooth running of an information system is a demanding task [14]. Hence, in order to avoid the occurrence of such an unpreferred situation in a computer network, high-performing models are needed for both hardware and software solutions, therefore, the study of Machine Learning (ML) is of importance to supplement both hardware and software-based solutions [15]. ML is the ability of computer algorithms to learn from a large amount of data through experience and provide a predicted output. ML can be applied in different fields such as data science, ICT, health care, finance, etc. [16].

There are four types of ML schemes: supervised, semi-supervised, unsupervised, and reinforcement learning. They both use classification algorithms. Examples of supervised learning classifications algorithms are Decision Tree (DT), k-Nearest Neighbors (k-NN), Naïve Bayes (NB), and logistic regression [6]. They use labeled training data as input features to generate the output [17]. Examples of unsupervised classification algorithms include k-means and hierarchical clustering [18]. They use unlabeled input features to generate the outputs.

The supervised learning classification relies on an ensemble learning technique to generate multiple models. Ensemble learning technique is a collection of classifiers used to build very sophisticated models with higher accuracy compared to single estimator classifiers [19, 50]. It is a machine learning approach in classifying datasets with high dimensions and its training process is not very complex [12]. Its output is based on the training data by aggregating them to generate a strong model. It thus fuses the results from several different models. This improves performance and prediction by stacking different models.

There are many studies related to the use of supervised and unsupervised ML in flow-based network traffic classification. For instance, in [20], the authors used ML in the classification of end users' applications. Different methods were used, such as k-NN, Random Forest (RF), and J48. The k-NN technique provided the best results followed by RF with accuracy of 93.94 % and 90.87% respectively. In [21], the authors compared Principal Component Analysis (PCA) with the Gaussian NB method. The mean accuracy of PCA was about 86% during the validation process in network intrusion detection compared to the 74% of Gaussian NB. A comparison study of DT, k-NN, Support Vector Machine (SVM), and RF was conducted in [22], resulting in higher accuracy of RF (96.87%) in comparison to the 48.56% of the SVM. This study shows that the ensemble learning method RF is very useful in network traffic data analytics compared to other ML techniques. In this study, SVM which is a supervised classifier, failed to separate network traffic based on feature classes used. Data mining approaches were applied in [18] in finding the dynamic patterns of network traffic. The study applied the clustering method by portioning the data from different domains and characterization the traffic in the time series data set. Two feature classes (benign and malicious) were considered. The authors in [23] compared the data mining approach using ML and ensemble learning method to forecast water flow. It was shown that the use of ensemble learning provided the best performance. Some of the performance metrics were not generated, for example recall, sensitivity, and kappa. When compared to other supervised algorithms, the use of ensemble learning in evaluating intrusion detection by using different data from network traffic tracing shows an improvement in network traffic classification [19].

The development of network infrastructure hardware has resulted in the use of the Deep Packet Inspection (DPI) method in classifying network attacks and threats. However, this approach needs a lot of memory as well as resources during computation. Another weakness of the method is that it is very difficult for database maintenance, especially for zero-day attacks and protocols [24]. In the field of ICT, there are three main approaches in classifying network traffic, namely flow, payload-inspection, and port-based methods [25]. Regarding the port-based and flow-based classification, it was shown in [26] that port-based classification has higher accuracy. However, ML classification provides both higher accuracy and performance results [27] compared to port and flow-based classifications. Based on the above study, this paper aimed at comparing the accuracy and performance of ML approaches.

Since the focus of the study for this article was to develop a learning classification model that can identify and detect network anomalies with high accuracy, especially zero-day attacks, we opted for supervised learning classification algorithms. The supervised learning classification algorithms provide higher accuracy than the unsupervised ones [24]. There are two types of ensemble learning methods in supervised learning classification [28]: Boosting and Bootstrap Aggregating (Bagging), both with a potential of being used in classification and regression. Boosting is an ensemble learning method which combines several weak learners to build a strong learner by using supervised classification [29]. Bagging methods divide the training data set into small samples for training the models.

In Tanzania, only a few studies have been conducted on the evaluation of computer system network traffic by using data mining and the ML approach. Authors in [30] compared network traffic classification and packet detection, showing that both computational performance and classification accuracy can be used for the management of computer network systems. This article thus aims to compare the performance of ensemble learning techniques (Bagging and Boosting) with the normal supervised classification algorithms, particularly k-NN, NB, LDA, MDA, and C5.0. The comparison is focused on three metrics (accuracy, kappa, and logloss). The question is whether using ensemble methods improves the accuracy and value of kappa. To fulfill this objective, we computed model sensitivity and specificity, precision and recall metrics from the models, and generated both positive and negative predictions from the models. This article contributes to the development of models, hardware or software, to detect network traffic anomalies in a much more effective and efficient way. It also contributes to the literature related to the ML approach in the computer network security field.

II. Materials and Methods

A. Data Capture and Classification Process

This section illustrates the structure of network traffic dataset classification step by step from data capture up to model evaluation as shown in Figure 1.

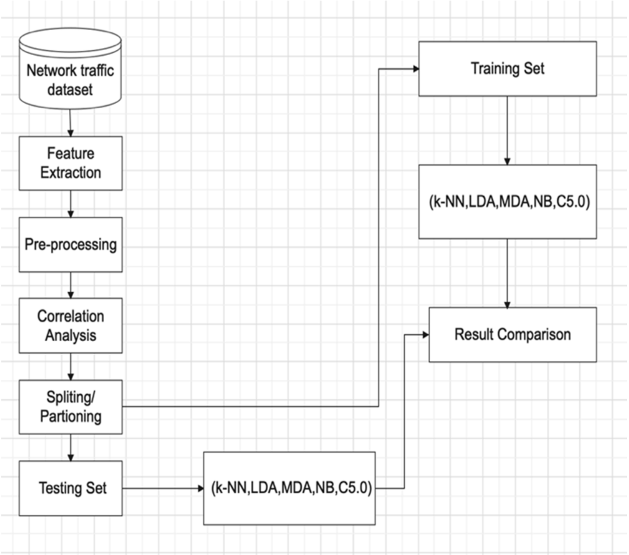

Fig. 1. Diagram illustrating data capture and classification.

B. Network Traffic Data Collection

We set experiments for network traffic data capture by using Paessler Router Traffic Grapher (PRTG) software and Cisco flow software in a Cisco 3900 router series hardware. The data captured at this stage were used to test the models. Online data donated by Mills [31] were downloaded in April 2021 from the Kaggle website (www.kaggle.com). These data were used for training the models with the variables shown in Table II. The experiment of the training dataset was set at Lancaster University's network address space. The data set contained robust ground truth through the correlation of malicious behavior in the network. The data were then stored in a computer and external hard disks for backup in packet capture (pcap) file format.

C. Feature Selection from Datasets

In supervised learning, after data capture, the next step is to select features from the data set collected from the network intended for testing the models. We used Joy software [32] which is a BSD-licensed libpcap-based software package for extracting features from live network traffic or pcap files. Sixteen variables with three feature class labels were generated in Comma Separated Values (CSV) and MS excel format (Table I).

D. Data Pre-processing

Data pre-processing was done in R software (Version. 4.1.2 named Bird Hippie) [33] by using RStudio editor Integrated Development Environment (IDE) [34]. Data pre-processing was performed to transform the data to a useful format for import and manipulation by ML algorithms. A total of 191,223 datasets were extracted, followed by feature selection and were labeled as benign (86,762), malicious (74,110), and outliers (30,351). The pre-processing stage generated 133,971 labeled samples. Out of these samples, 70% (n = 93,780) and 30% (n = 40,191) were used for training and model testing respectively. The dataset was then scaled and centered by using median imputation for every variable.

TABLE I. DESCRIPTION OF VARIABLES

Complete name	Abbreviation
Average inter packet time	Avgipt
Bytes in	Bytesin
Bytes out	Bytesout
Destination IP address	Destip
Destination port number	Destport
Entropy	Entropy
Number of packets out	Numpktsout
Number of packets in	Numpktsin
Protocol	Proto
Source IP address	Srcip
Source port number	Srcport
Time end	Timeend
Time start	Timestart
Total entropy	Totalentropy
Duration	Duration
Feature class label	Label

E. Variable Multicollinearity Test

After the data pre-processing stage, we looked for variable multicollinearity by using the Spearman correlation coefficient test [35]. The Variance Inflation Factor (VIF) method was applied to detect and remove highly correlated variables based on VIF interpretation as shown in Table II.

TABLE II. VIF INTERPRETATION

VIF value	Conclusion
VIF = 1	Not correlated
1 < VIF < 6	Moderate correlated
VIF > 6	Highly correlated

To identify variables to remove or to retain in the model, R software [33] was used and RStudio IDE [34]. Model development and data analysis were conducted in a laptop with Quad Intel Core i5, 8GB of RAM, and 560 SSD running macOS Big Sur. VIF is the measure of how the variance is inflated by the correlation of the predictors which leads to the variance increase of predictors [36]. Variables with higher correlation were removed from the list while those with VIF $1 \leq \text{VIF} < 6$ were kept for model development as shown in Table III.

F. Model Development

Models were developed by using the classification and regression training (Caret) packages [21] in R software with R programming language. Other packages (e.g. ggplot2, randomForest, and xgboost) were used for calculations, data manipulation, and visualization. A total of nine predictors, with three classes, namely benign, malicious, and outlier from

133,971 samples were used. Re-sampling was done by using the 10-fold cross-validation method to validate the outcome of the classifier, whereby the classification was repeated once. In this paper, different models were developed by using normal supervised classifications: C5.0, k-NN, Mixture Discriminant Analysis (MDA) algorithm, Regularized Discriminant Analysis (RDA), and NB. For comparison purposes ensemble learning classifiers, namely RF from boosting techniques and eGB from bagging techniques were used as indicated in Table IV.

TABLE III. MULTI-COLINEARITY TEST

Variables	Tolerance	VIF	Eigenvalue	Condition index
avgipt	0.998	1.002	3.421	1.000
bytesin	0.660	1.516	1.470	1.525
bytesout	0.351	2.852	1.096	1.767
entropy	0.816	1.225	0.990	1.859
numpktsout	0.300	3.338	0.691	2.224
numpktsin	0.268	3.738	0.645	2.303
totalentropy	0.198	5.051	0.372	3.031
distance	0.807	1.238	0.186	4.292

TABLE IV. CLASSIFIERS AND LEARNING METHOD

Classifier	Learning method
eGB	Ensemble (Boosting)
RF	Ensemble (Bagging)
C5.0	Normal Supervised
k-NN	Normal Supervised
RDA	Normal Supervised
MDA	Normal Supervised
NB	Normal Supervised

A serial process for RF and eGB as sequential and parallel categories of ensemble learning classifiers respectively was completed as indicated in Figure 2. RF algorithm depends on aggregating the output from several trees. Trees are modified, pruned, and an average of the results and predictions is done by using the estimation of the dependent variables on new observations. The eGB a popular ensemble learner' method which is used in ML with AdaBoost in DTs. It avoids over-fitting challenges and its accuracy is higher than AdaBoost's.

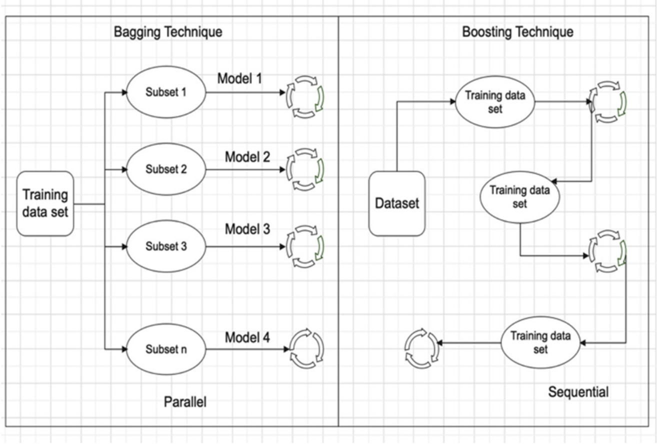

Fig. 2. Flow chart for ensemble learning classifiers.

The normal classifiers and learning methods are described below.

1) C5.0 Classification Model

This model is an extension of C4.5 which establishes a DT where every feature is considered during classification [37]. The trees constructed by C50 have high accuracy and a minimum breakdown which makes the classifier reliable and faster. The model is used to handle non-numerical features such as factor, character, etc., therefore, the model was used as a DT classifier or boosted classifier following [38]. In most studies, C5.0 performs higher than CART and C4.5 [39].

2) k-Nearest Neighbors Algorithm

In ML, k-NN is considered as a lazy learning classifier and it is used to classify objects that are closely related in training data samples based on instance learning. It uses similarity and distance between two points and categorizes the dataset based on the distance or similarities from other categories as shown in (1) and Figure 2. By calculating the Euclidean distance [40], the New Class in the figure will belong to Class C and not in Class B whereas, by using similarities, this occurs when we choose k = 4 as the number of neighbors. Alternatively, these can be done by using the Euclidean distance as indicated in (1) from P1 to P2.

$$\text{Euclidean Distance}=\sqrt{(P2-P1)^2+(K2-K1)^2} \quad (1)$$

In this study, we used k = 29 because it produced the optimal results for the acquired data.

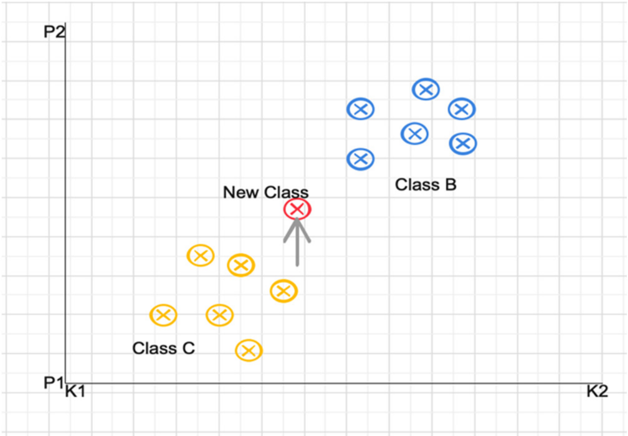

Fig. 3. Diagram illustrating the k-NN classification technique.

3) Mixture Discriminant Analysis

Discriminant analysis is used to predict the probability of belonging to a given class (or category) based on one or multiple predictor variables. It works with continuous and/or categorical predictor variables. In MDA, each class is assumed to be a Gaussian mixture of subclasses. It is the extension of Linear Discriminant Analysis (LDA) which can be used as supervised classification. The method is nonparametric because it minimizes within-group variability. LDA can be used in multi-class classification methods that follow the Gaussian theorem to model classes. In our dataset we had three classes denoted by "P" and our training sample was denoted by (yl……. yn) with classes (w1…… wn), where wi ∈ {1...P}. The prior probability jk of each class follows the Gaussian model θ(y |μp, Σ). The estimation from the model (a_p) can be generated mathematically by:

$$a_p = \sum_{i=1}^{n} \frac{I(z_i = p)}{n} \quad (2)$$

where a_p is the model estimate, p is the number of classes in the data sets, n the number of samples, and z_i a constant for normal distribution.

4) Regularized Discriminant Analysis

RDA uses multivariate means as well as a covariance matrix. The properties are generated from the data and used in the predictions. RDA data use Gaussian assumptions whereby each variable when plotted is like a bell curve. The model generates variance and means of each class from the data as illustrated in (3):

$$mean_p = \frac{1}{np} \sum_{i=1}^{n} xi \quad (3)$$

The variance of the samples was computed using (4):

$$Variance = \frac{1}{n-P} \sum_{i=1}^{n} [\![(xi]\!] - mean_p) \quad (4)$$

where n is the number of instances, *P* the number of classes, *x* the input values, and *np* is the number of classes in the instance.

For model prediction in RDA, we used the classes with the highest probability of the classes (*h*) with *x* as input through the Bayesian theorem as illustrated in (5):

$$P(Y = h\,|X = x) = \frac{P(h) \times P(x|h)}{\sum_{l=1}^{K} (P(l) \times P(x|l))} \quad (5)$$

where *P*(*x*|*k*) is the estimated probability of *x* belonging to the class *k*, *P*(*Y* = *k*|*X* = *x*) is the probability of the class (*Y* = *k*) given the input data *x*, and *P*(*k*) is the base probability of a given class *k*. We are considering (Y = k).

5) Naïve Bayes

The use of NB classifiers in ML especially in anomaly detection has been widely applied in filtering spam emails. The accuracy of separating spam in the email is limited because its strength depends on the independence between the features [41]. The model also suffers from the heavy overhead computation which makes the mode use more resources during execution [42]. Therefore, we used this model with others for comparison due to simplicity and efficiency. NB uses the concept of the Bayesian theorem with the assumption of prior knowledge of a given hypothesis to classify features. The theorem state as:

$$P(h|d) = \frac{P(d|h) \times P(h)}{P(d)} \quad (6)$$

where *P*(*d*) is the probability of the data, *P*(*h*) is the probability of hypothesis *h* being true, *P*(*h*|*d*) the posterior probability, *P*(*d*|*h*) the probability of data *d* given that the hypothesis *h* was true. Likewise, the maximum posterior (MAP) hypothesis can be calculated by applying (7):

$$MAP(h) = \max(\frac{P(d|h) \times P(h)}{P(d)}) \quad (7)$$

6) Model Evaluation Metrics

The proposed classification techniques used two ensemble learning methods versus five normal supervised ML. Model Accuracy, Precision, Recall, F1 score metrics were used for model evaluations. Recall, F1 score, Precision, Accuracy can be mathematically computed by using the equations from Table V [11, 28]. True Positive (TP) occurs when the values of both predicted class and actual class are 1. True Negative (TN) occurs when the values of the predicted actual classes are both 0. False negatives (FN) and False Positives (FP) occur when the predicted class changes the actual class.

TABLE V. MODEL EVALUATION METRICS

Metrics	**Formula**
Accuracy	(TP + TN) / (TP+TN+FP+FN)
True positive rate	TP / (TP + FN)
False positive rate	FP / (FP + TN)
Precision	TP / (TP + FP)
Recall	TP / (TP + FN)
F1 score	2 Precision × Recall / (Precision + Recall)

III. RESULTS AND DISCUSSION

The study's main objective was to evaluate the performance of different models in network traffic classifications. To achieve this objective, the current study used ensemble learning methods and normal supervised classifications for comparison. Multilabel data features were classified by using different models. The following evaluation metrics were applied for both ensemble and normal supervised learning: Accuracy, Under the Curve (AUC), Precision, Recall, Sensitivity, Specificity, Positive and Negative predictions.

A. Model Accuracy

1) Normal Supervised Learning Methods

Results from the five algorithms that were developed before subjecting individual models to ensemble learning techniques showed that k-NN had the highest accuracy followed by C5.0 and MDA with accuracy of 0.868, 0.761, and 0.741 respectively. NB classifier scored the lowest accuracy of 0.696 as shown in Table VI. These results are in accordance with the findings in [25, 43, 44].

TABLE VI. EVALUATION METRICS OF NORMAL SUPERVISED LEARNING METHODS

Method	**Accuracy**	**Kappa**
k-NN	0.868	0.833
C5.0	0.761	0.585
MDA	0.742	0.558
RDA	0.738	0.561
NB	0.696	0.491

2) Ensemble Models

Results from the two higher-performing ensemble models in higher dimensional dataset techniques showed that RF had higher accuracy compared to eGB as presented in Table VII. Our study is supported by [45], in which ensemble methods (xGB and RF) were used with accuracy of 89.09% and 85.49%

respectively. Another study that supports our result was [13] as indicated in the comparison in Table XIII.

TABLE VII. EVALUATION METRICS OF ENSEMBLE LEARNING METHODS

Method	Accuracy	Kappa	Logloss
RF (bagging)	0.904	0.841	2.369
eGB (boosting)	0.902	0.830	0.223

3) *Comparison of Normal Supervised and Ensemble Models*

The results from the comparison done after developing the models by using ensemble learning and supervised algorithms revealed that in normal supervised algorithms, k-NN had the highest accuracy as shown in Table VI. Both the ensemble learning methods had higher accuracy, with RF having the highest.

B. Evaluation of the Normal Supervised Learning Processed by Ensemble Classifier

After processing supervised algorithms by using ensemble learning methods, there was an improvement of accuracy and Kappa values as shown in Table VIII. C5.0 had the highest accuracy (0.902) as compared to the previous accuracy of 0.761 (Table V). On the other hand, k-NN also improved with a small margin from 0.868 to 0.898.

TABLE VIII. ACCURACY AND KAPPA FOR NORMAL SUPERVISED LEARNING PROCESSED BY ENSEMBLE CLASSIFIERS

Normal supervised learning	Accuracy	Kappa	Logloss
C5.0	0.902	0.839	0.229
k-NN	0.898	0.832	2.699
RDA	0.738	0.561	0.653
MDA	0.739	0.560	0.747
NB	0.698	0.470	0.471

C. Model AUC

AUC was the highest in eGB, followed by C5.0, whereas RDA had the least as shown in Table IX. The value of F1 score metrics, as the measure of the test's accuracy, was highest in C50, followed by k-NN and eGB. RDA scored the least F1-score.

TABLE IX. AUC AND F1 SCORE

Classifier	AUC	F1-scores
eGB	0.979	0.865
C5.0	0.978	0.870
RF	0.944	0.854
k-NN	0.916	0.868
NB	0.910	0.521
MDA	0.888	0.589
RDA	0.863	0.489

D. Precision and Recall

The results showed that eGB has the highest Precision followed by k-NN and RF, while RDA scored the least Precision as presented in Table X. RF exhibited the highest Recall followed by C50 and k-NN, while NB scored the lowest value.

TABLE X. PRECISION AND RECALL RESULTS

Classifier	Precision	Recall
eGB	0.906	0.843
k-NN	0.900	0.849
RF	0.900	0.867
C50	0.897	0.865
NB	0.777	0.553
MDA	0.670	0.601
RDA	0.558	0.593

E. Model Sensitivity and Specificity

RF scored the highest sensitivity, followed by C50 and k-NN, while NB had the lowest sensitivity as shown in Table XI. Furthermore, RF attained the highest specificity, followed by k-NN and eGB, while NB scored the lowest.

TABLE XI. SENSITIVITY AND SPECIFICITY RESULTS

Classifier	Sensitivity	Specificity
RF	0.867	0.946
C50	0.865	0.897
k-NN	0.849	0.942
eGB	0.843	0.943
MDA	0.601	0.859
RDA	0.593	0.859
NB	0.553	0.822

F. Prediction

Both Positive and Negative predictions generated from the models are presented in Table XII. eGB attained the highest positive prediction followed by RF and k-NN. RDA scored the lowest positive prediction. NB scored the highest negative prediction, followed by MDA.

TABLE XII. POSITIVE AND NEGATIVE PREDICTIONS

Classifier	Positive	Negative
eGB	0.906	0.954
k-NN	0.900	0.954
RF	0.900	0.955
C50	0.897	0.955
NB	0.777	0.874
MDA	0.670	0.883
RDA	0.558	0.891

G. Discussion

The result from this study has been compared with the results from [13] as shown in Table XIII and Figure 4. The acquired results show that the proposed techniques achieved better accuracy, AUC, Recall, and Precision, but not F1 score. The study which [13] shows that eGB, C5.0, and RF had an accuracy of 0.901, 0.886, and 0.885 respectively. The findings of this paper also show an accuracy of 0.902, 0.902, and 0.904 for C5.0, eGB, and RF. Therefore, our results are higher considering model accuracies. The study conducted in [46] was looking at anomaly detection by using ML techniques by using RF. One of the performance metrics was the accuracy of RF which was 99.7 which is higher compared to this paper results. Another study [47] was utilized the RF classifier and scored an accuracy of 0.893, Recall 0.890, F1 score of 0.896, and precision of 0.92. In [45], eGB obtained the following metrics scores; Accuracy of 0.926, Precision 0.927, Recall 0.926, and

F1 score 0.924 which are similar or less than the same respective scores of the current study (Table XIII).

TABLE XIII. COMPARISON WITH [13]

Classifier	Accuracy	AUC	Recall	Precision	F1 score	Kappa
[13]						
eGB	0.901	0.995	0.75	0.901	0.899	0.875
RF	0.885	0.948	0.741	0.882	0.885	0.855
C5.0	0.886	0.985	0.725	0.883	0.884	0.856
k-NN	0.85	0.966	0.675	0.846	0.847	0.811
NB	0.297	0.633	0.128	0.23	0.228	0.096
Current study						
MDA	0.739	0.888	0.601	0.67	0.56	0.56
RDA	0.742	0.86	0.593	0.558	0.561	0.738
RF	0.904	0.979	0.867	0.9	0.854	0.841
eGB	0.902	0.944	0.843	0.906	0.865	0.83
NB	0.698	0.91	0.553	0.777	0.521	0.47
k-NN	0.898	0.916	0.849	0.9	0.868	0.32
C5.0	0.902	0.978	0.865	0.897	0.87	0.839

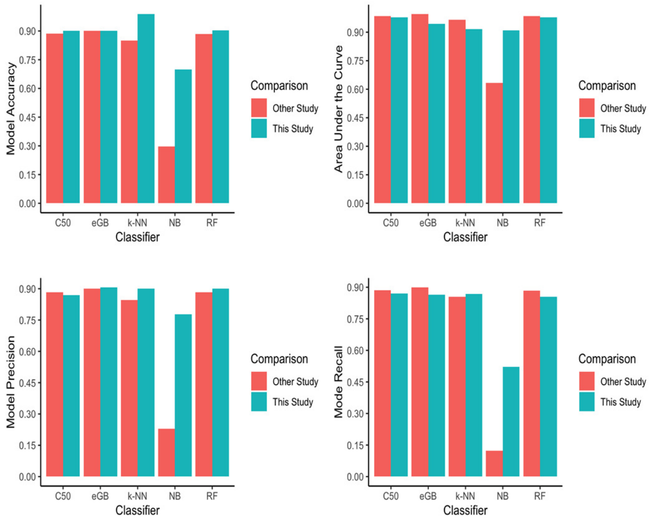

Fig. 4. Graphs showing comparisons of this study with others.

The results of this paper are also equivalent and sometimes above the results of [48, 49]. One can conclude that the results of the current study are supported by other studies, however, with slight variations in some parameters.

IV. CONCLUSION, RECOMMENDATIONS, AND FUTURE WORK

This article presented and compared the results of normal supervised algorithms and ensemble learning techniques, namely RF and eGB. The individual classifiers were compared with ensemble learners by using a real experimental dataset with little correlation. The overall accuracy of the ensemble methods was higher than the accuracy of normal classifiers. Therefore, we can conclude that the ensemble learning techniques can be used to classify the multilabel network traffic.

This study contributes to the knowledge of network traffic classification by using supervised and ensemble learning and multilabel datasets. To the best of our knowledge there are no similar studies regarding the network traffic classification in Tanzania.

The current study can be extended to new emerging technologies (edge computing, cyber security, e-commerce, fog computing, and distributed databases such as Blockchain). Also, the use of emerging ML approaches like reinforcement and deep learning could be applied with new experimental datasets. The performance comparison of ensemble learning with other learning methods in classifying network traffic in emerging technologies is very important. Application of higher processing speeds and distributed systems such as H2O, Apache Spark, etc. to facilitate the application of big data in the massive network traffic data can be also considered.

FUNDING AND ACKNOWLEDGMENT

The current study was funded by the European Union's Horizon 2020 research and innovation grant number [641918] through the AFRICANBIOSERVICES PROJECT in Tanzania under Tanzanian Wildlife Research Institute (TAWIRI). The authors extend their appreciation to Tanzania Communications Regulatory Authority (TCRA), TAWIRI, and Tanzania National Parks (TANAPA) for their support during data collection, and NM-AIST for the conducive working and research environment.

REFERENCES

[1] G. Aceto, V. Persico, and A. Pescape, "The role of Information and Communication Technologies in healthcare: taxonomies, perspectives, and challenges," *Journal of Network and Computer Applications*, vol. 107, pp. 125–154, Apr. 2018, https://doi.org/10.1016/j.jnca.2018.02.008.

[2] S. Morgan, "The 2020 Data Attack Surface Report," Arcserve, 2020.

[3] J. Shi, C. Pan, W. Zhang, and M. Chen, "Performance Analysis for User-Centric Dense Networks With mmWave," *IEEE Access*, vol. 7, pp. 14537–14548, 2019, https://doi.org/10.1109/ACCESS.2019.2893403.

[4] TCRA, "A: TELECOM SERVICES," 2021

[5] G. Ali, M. Ally Dida, and A. Elikana Sam, "Two-Factor Authentication Scheme for Mobile Money: A Review of Threat Models and Countermeasures," *Future Internet*, vol. 12, no. 10, Oct. 2020, Art. no. 160, https://doi.org/10.3390/fi12100160.

[6] N. B. Amor, S. Benferhat, and Z. Elouedi, "Naive Bayes vs decision trees in intrusion detection systems," in *ACM Symposium on Applied Computing*, Nicosia, Cyprus, Mar. 2004, pp. 420–424, https://doi.org/10.1145/967900.967989.

[7] X. Liu *et al.*, "Attention-based bidirectional GRU networks for efficient HTTPS traffic classification," *Information Sciences*, vol. 541, pp. 297–315, Dec. 2020, https://doi.org/10.1016/j.ins.2020.05.035.

[8] P. Barford and D. Plonka, "Characteristics of network traffic flow anomalies," in *1st ACM SIGCOMM Workshop on Internet measurement*, San Francisco, CA, USA, Nov. 2001, pp. 69–73, https://doi.org/10.1145/505202.505211.

[9] L. Machlica, K. Bartos, and M. Sofka, "Learning detectors of malicious web requests for intrusion detection in network traffic," *arXiv:1702.02530 [cs, stat]*, Feb. 2017, Accessed: Apr. 20, 2022.

[10] S. Manaseer, O. Al-Nahar, and A. Hyassat, "Network Traffic Modeling, Case Study: The University of Jordan," *International Journal of Recent Technology and Engineering*, vol. 7, no. 5, pp. 13–16, Jan. 2019.

[11] K. Aldriwish, "A Deep Learning Approach for Malware and Software Piracy Threat Detection," *Engineering, Technology & Applied Science Research*, vol. 11, no. 6, pp. 7757–7762, Dec. 2021, https://doi.org/10.48084/etasr.4412.

[12] Z. Liu, N. Su, Y. Qin, J. Lu, and X. Li, "A Deep Random Forest Model on Spark for Network Intrusion Detection," *Mobile Information Systems*, vol. 2020, Dec. 2020, Art. no. e6633252, https://doi.org/10.1155/2020/6633252.

[13] K. Demertzis, K. Tsiknas, D. Takezis, C. Skianis, and L. Iliadis, "Darknet Traffic Big-Data Analysis and Network Management for Real-

Time Automating of the Malicious Intent Detection Process by a Weight Agnostic Neural Networks Framework," *Electronics*, vol. 10, no. 7, Jan. 2021, Art. no. 781, https://doi.org/10.3390/electronics10070781.

[14] A. D'Alconzo, I. Drago, A. Morichetta, M. Mellia, and P. Casas, "A Survey on Big Data for Network Traffic Monitoring and Analysis," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 800–813, Sep. 2019, https://doi.org/10.1109/TNSM.2019.2933358.

[15] R. de O. Schmidt, R. Sadre, and A. Pras, "Gaussian traffic revisited," in *IFIP Networking Conference*, Brooklyn, NY, USA, Dec. 2013, pp. 1–9.

[16] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, Jul. 2015, https://doi.org/10.1126/science.aaa8415.

[17] C. Dong, C. Zhang, Z. Lu, B. Liu, and B. Jiang, "CETAnalytics: Comprehensive effective traffic information analytics for encrypted traffic classification," *Computer Networks*, vol. 176, Jul. 2020, Art. no. 107258, https://doi.org/10.1016/j.comnet.2020.107258.

[18] W. Ruan, Y. Liu, and R. Zhao, "Pattern Discovery in DNS Query Traffic," *Procedia Computer Science*, vol. 17, pp. 80–87, Jan. 2013, https://doi.org/10.1016/j.procs.2013.05.012.

[19] H. He, X. Luo, F. Ma, C. Che, and J. Wang, "Network traffic classification based on ensemble learning and co-training," *Science in China Series F: Information Sciences*, vol. 52, no. 2, pp. 338–346, Feb. 2009, https://doi.org/10.1007/s11432-009-0050-8.

[20] B. Yamansavascilar, M. A. Guvensan, A. G. Yavuz, and M. E. Karsligil, "Application identification via network traffic classification," in *International Conference on Computing, Networking and Communications*, Silicon Valley, CA, USA, Jan. 2017, pp. 843–848, https://doi.org/10.1109/ICCNC.2017.7876241.

[21] B. Zhang, Z. Liu, Y. Jia, J. Ren, and X. Zhao, "Network Intrusion Detection Method Based on PCA and Bayes Algorithm," *Security and Communication Networks*, vol. 2018, Nov. 2018, Art. no. e1914980, https://doi.org/10.1155/2018/1914980.

[22] G. Harinahalli Lokesh and G. BoreGowda, "Phishing website detection based on effective machine learning approach," *Journal of Cyber Security Technology*, vol. 5, no. 1, pp. 1–14, Jan. 2021, https://doi.org/10.1080/23742917.2020.1813396.

[23] A. Hussein, J. Agbinya, and I. Satti, "A Survey on Data mining Techniques for Water Flow Forecasting," *Australian Journal of Basic and Applied Sciences*, vol. 14, no. 3, pp. 13–27, 2019, https://doi.org/10.22587/ajbas.2020.14.3.2.

[24] S. E. Gomez, L. Hernandez-Callejo, B. C. Martinez, and A. J. Sanchez-Esguevillas, "Exploratory study on Class Imbalance and solutions for Network Traffic Classification," *Neurocomputing*, vol. 343, pp. 100–119, May 2019, https://doi.org/10.1016/j.neucom.2018.07.091.

[25] M. Soysal and E. G. Schmidt, "Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison," *Performance Evaluation*, vol. 67, no. 6, pp. 451–467, Jun. 2010, https://doi.org/10.1016/j.peva.2010.01.001.

[26] F. Dehghani, N. Movahhedinia, M. R. Khayyambashi, and S. Kianian, "Real-Time Traffic Classification Based on Statistical and Payload Content Features," in *2nd International Workshop on Intelligent Systems and Applications*, Wuhan, China, Dec. 2010, pp. 1–4, https://doi.org/10.1109/IWISA.2010.5473467.

[27] R. M. AlZoman and M. J. F. Alenazi, "A Comparative Study of Traffic Classification Techniques for Smart City Networks," *Sensors*, vol. 21, no. 14, Jan. 2021, Art. no. 4677, https://doi.org/10.3390/s21144677.

[28] D. Chopra, N. Joshi, and I. Mathur, "Improving Translation Quality By Using Ensemble Approach," *Engineering, Technology & Applied Science Research*, vol. 8, no. 6, pp. 3512–3514, Dec. 2018, https://doi.org/10.48084/etasr.2269.

[29] A. J. Wyner, M. Olson, J. Bleich, and D. Mease, "Explaining the Success of AdaBoost and Random Forests as Interpolating Classifiers," *Journal of Machine Learning Research*, vol. 18, no. 48, pp. 1–33, 2017.

[30] G. S. Oreku, F. J. Mtenzi, and C. A. Shoniregun, "Traffic classification and packet detections to facilitate networks security," *International Journal of Internet Technology and Secured Transactions*, vol. 3, no. 3, pp. 240–252, Jan. 2011, https://doi.org/10.1504/IJITST.2011.041294.

[31] R. Mills, "LUFlow Network Intrusion Detection Data Set." https://www.kaggle.com/mryanm/luflow-network-intrusion-detection-data-set (accessed Apr. 20, 2022).

[32] B. Hudson, "Understanding Encrypted Traffic Using 'Joy' for Monitoring and Forensics," presented at the Cisco Live!, Orlando, FL, USA, Jun. 10, 2018.

[33] R Core Team, *R: a language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing, 2018.

[34] J. J. Allaire, "RStudio: Integrated Development Environment for R," presented at the The R User Conference 2011, Coventry, UK, Aug. 2011.

[35] J. Hauke and T. Kossowski, "Comparison of Values of Pearson's and Spearman's Correlation Coefficients on the Same Sets of Data," *Quaestiones Geographicae*, vol. 30, no. 2, pp. 87–93, Jun. 2011, https://doi.org/10.2478/v10117-011-0021-1.

[36] J. I. Daoud, "Multicollinearity and Regression Analysis," *Journal of Physics: Conference Series*, vol. 949, Sep. 2017, Art. no. 012009, https://doi.org/10.1088/1742-6596/949/1/012009.

[37] J. R. Quinlan, "Book Review : C4 . 5 : Programs for Machine Learning," *Machine Learning*, vol. 16, pp. 235–240, 1994.

[38] M. Kuhn, "The caret Package," *Journal of Statistical Software*, vol. 28, Jan. 2012.

[39] S. Garg, "An evaluation of investor acceptability for physical gold using classification (Decision Tree)," *Materials Today: Proceedings*, vol. 37, pp. 950–954, Jan. 2021, https://doi.org/10.1016/j.matpr.2020.06.177.

[40] D. Hamilton, R. Pacheco, B. Myers, and B. Peltzer, "kNN vs. SVM: A comparison of algorithms," in *Fire Continuum-Preparing for the future of wildland fire*, Missoula, USA, Dec. 2018, vol. 78, pp. 95–109.

[41] W. Feng, J. Sun, L. Zhang, C. Cao, and Q. Yang, "A support vector machine based naive Bayes algorithm for spam filtering," in *35th International Performance Computing and Communications Conference*, Las Vegas, NV, USA, Dec. 2016, pp. 1–8, https://doi.org/10.1109/PCCC.2016.7820655.

[42] L. Jiang, Z. Cai, and D. Wang, "Improving Naive Bayes for Classification," *International Journal of Computers and Applications*, vol. 32, no. 3, pp. 328–332, Jan. 2010, https://doi.org/10.2316/Journal.202.2010.3.202-2747.

[43] A. Callado *et al.*, "A Survey on Internet Traffic Identification," *IEEE Communications Surveys Tutorials*, vol. 11, no. 3, pp. 37–52, 2009, https://doi.org/10.1109/SURV.2009.090304.

[44] S. Chen, G. I. Webb, L. Liu, and X. Ma, "A novel selective naïve Bayes algorithm," *Knowledge-Based Systems*, vol. 192, Mar. 2020, Art. no. 105361, https://doi.org/10.1016/j.knosys.2019.105361.

[45] O. Aouedi, K. Piamrat, and B. Parrein, "Performance evaluation of feature selection and tree-based algorithms for traffic classification," in *International Conference on Communications Workshops*, Montreal, QC, Canada, Jun. 2021, https://doi.org/10.1109/ICCWorkshops50388.2021.9473580.

[46] J. J. Estevez-Pereira, D. Fernandez, and F. J. Novoa, "Network Anomaly Detection Using Machine Learning Techniques," *Proceedings*, vol. 54, no. 1, 2020, Art. no. 8, https://doi.org/10.3390/proceedings2020054008.

[47] V. Dutta, M. Choras, M. Pawlicki, and R. Kozik, "A Deep Learning Ensemble for Network Anomaly and Cyber-Attack Detection," *Sensors*, vol. 20, no. 16, Jan. 2020, Art. no. 4583, https://doi.org/10.3390/s20164583.

[48] M. Singh, G. Srivastava, and P. Kumar, "Internet Traffic Classification Using Machine Learning," *International Journal of Database Theory and Application*, vol. 9, pp. 45–54, Dec. 2016, https://doi.org/10.14257/ijdta.2016.9.12.05.

[49] T. C. Obasi, "Encrypted Network Traffic Classification using Ensemble Learning Techniques," M.S. thesis, Carleton University, Ottawa, ON, Canada, 2020.

[50] D. K. Singh and M. Shrivastava, "Evolutionary Algorithm-based Feature Selection for an Intrusion Detection System," *Engineering, Technology & Applied Science Research*, vol. 11, no. 3, pp. 7130–7134, Jun. 2021, https://doi.org/10.48084/etasr.4149.