Intrusion Detection on UNSBW-NB15 by using Machine Learning Techniques

1st Eslam Mahmoud
University of Ottawa
Cairo, Egypt
emahm025@uOttawa.ca

2nd Loai Nazeer
University of Ottawa
Cairo, Egypt
lnaze101@uOttawa.ca

3rd Mohamed Elgaidy
University of Ottawa
Cairo, Egypt
melga045@uOttawa.ca

4th Mohamed El-Gharieb
University of Ottawa
Cairo, Egypt
melgh048@uOttawa.ca

***Abstract*—Since the world prepares to enter fifth-generation communication technology and embrace concepts like virtualization and cloudification, the most important factor to consider is "security," as more and more data is connected to the internet. This article depicts a paradigm for calculating numerous data parameters in a network, such as accuracy, precision, and confusion matrix, among others. The overall goal is to have a better understanding of data integrity and improve data prediction accuracy. By doing so, the quantity of malicious material floating around in a network may be reduced, making it a safer place to share data.**

***Index Terms*—intrusion detection system (IDS) , classifiers, UNSBW-NB15,network security , machine learning , dataset preprocessing**

I. Introduction

The need to secure networks from attackers is more than ever before. Tools and devices have surfaced to secure networks and computer systems, ranging from antiviruses, firewalls, vulnerability testing, and elimination of the programming errors that provide a backdoor entry point to the system. All these devices for network security can be categorized as reactive or proactive, where proactive methods are based on the assumption that security can be guaranteed, and hence it is possible absolutely. Prevention is no more effective. No matter how securely a network is laid down, eventually, hackers will find a way to break into the system and hence threaten the confidentiality, availability, and integrity of the system.

This leads us to the reactive system that makes use of network and system logs to unearth the footprints of disastrous network connections, IDS starts with the collection of data from the networks, followed by data manipulation and finally a test for abnormality, that classifies a network connection as normal or abnormal. In the earlier phases, the research focal point lied with rule-based expert methods and statistical procedures. However, these rule-based systems tend to be less effective on larger datasets. A considerable measure of machine learning (ML) techniques has been introduced to tackle the problem of ID. IDS comes in handy not only in the successful detection of intrusions but is also effective in monitoring attempts to break security, which is indispensable for timely countermeasures against the ongoing intrusion activity.

NIDS collects and analyses the traffic communicated over the network. So, it can carry out intrusion detection with security measures reflecting entire network information, hence preventing the attacks from reaching the hosts. anomaly-based approaches are trained on the unlabeled data, and in the training phase they build up the model for the normal connections. Over the span of operation, the system captures the data and compares it with the model for the normal data. If it differs by more than some threshold then it is classified as an attack, otherwise a legitimate connection. The power of anomaly-based systems comes with the ability to detect novel attacks, but this comes with higher false alarm rates. For the last few decades researchers have approached the ID problem in various models and the most popular of them all is the application of data mining methods. Data mining explores and analyzes gigantic datasets to unearth the useful and understandable patterns from the data. A prerequisite for data mining is the availability of lots of data, the presence of patterns in the data, and the presence of no simple model to understand the data.[1]

In the preprocessing part, we noticed the UNSW-NB15 15 dataset. Some problems that will affect our Progress in our model as we plan to handle redundancy by under-sampling by using Condensed Nearest Neighbours Rule CNN. Also, the missing data as reference [2] we will handle by using multiple methods as mean and linear regression. For category data, we decided to use Ordinal Encoding in features that had reflected sequence. And One Hot Encoding in others features.

Moving to feature normalization we found consensus from most of the previous work using Min-Max scaling related to types of the features. Finally, in feature selection, we will remove irrelevant or less relevant attributes or noisy data and use RMS Methodology to find correlation and relations between features and finally handle the most important feature using XGBoost algorithm feature importance function as reference [7].

From other works we decided to use a collection of machine learning models with ensemble learning techniques, so we will apply Random Forest, Decision Tree, XGBoost, and Support Vector Machine algorithms with bagging and stacking as ensemble learning techniques.

II. Related Work

A. Dataset

The complete dataset was described as having accurately identified classes that were open to everyone, including real-world network traffic rather than synthetic traffic, containing all types of attacks, and being constantly updated. It should also include packet header information and the data payload, which should be recorded over an extended period. The UNSW-NB15 was one of their general suggestions for IDS testing based on the number of assaults reported in the accessible datasets.[2]

The rationale for improved results on NSL-KDD vs UNSBW-NB15 is that this dataset does not contain any duplicated connections, and the data is divided uniformly between various kinds of attacks as well as regular connections.[3] For efficacy measurement, the features in the KDD99 and UNSW-NB15 datasets were studied. For their trials, they utilized an association rule mining method as well as a few other known classifiers. According to the study, the UNSBW-NB15 has better detection accuracy and a lower number of false alarms than the KDD99.[4]

Used a deep neural network (DNN) on the internet of things to evaluate the KDD99, NSL-KDD, and UNSW-NB15 datasets. Using a same evaluation metric and F1 measure, the study showed that DNN was able to gain an accuracy of more than 90% for all datasets. DNN also got the best performance on UNSW-NB15.[5]

In [6] The NSL-KDD and UNSW-NB15 features were assessed using four filter-based feature-selection measures: correlation measure (CFS), consistency measure (CBF), information gain (IG), and distance measure (ReliefF). The features chosen from those four approaches were then assessed using four classifiers to indicate training and testing performance, namely k-NN, random forests (RF), support vector machine (SVM), and deep belief network (DBN). The study published the selected features for each feature selection approach, as well as the classification results, intending to assist cybersecurity researchers in creating successful IDS.

B. Data Preprocessing

In this section, we discussed the preprocessing technique to solve the problem in UNSW-NB15 to verify the existence of redundant input features in UNSW-NB15 data records, convert the existing nominal input features to the numerical format, map them into the same scale, and extract the most informative input features to prevent the features from misleading clustering results as well as reducing the computational cost in the visualization [7][10][12] which will affect the result and accuracy of our Models, the previous work on preprocessing divided into the following points:

1) Redundancy in dataset: Redundancy in the dataset: Redundant in the UNSW-NB15 dataset is imbalance Class imbalance is composed of between classes and within-class imbalance.[7] Between-class imbalance corresponds to the case where one class or multiple classes in a dataset are underrepresented in comparison with other classes.[7]

- As related to [7] Two input features, namely record the start time and record last time, are redundant due to the presence of total duration (dur) that is obtained from the difference of 5 values for these two features.
- In reference [8] use of used under damping imbalanced dataset to overcome this problem. They reduced the number of normal packets by 50% but kept the original number of packets of other classes, so the remaining data are now 6% of the actual data.[8]

2) Handling the missing values: Missing data can present significant bias, making the management and analysis of the data more difficult in addition to dropping the accuracy.[8]

- Retiled to the author in [8] the missing values occur predominantly in three features, i.e., CT FW HTTP MTHD, is FTP login and CT FTP cmd. [8] they choose to apply imputation better than the drop and effect accuracy model. they applied three types of means, Multiple and Linear Regression imputation techniques to overcome the problem of missing values.
- But related to [14] they choose Data encoding was used to convert object features into vectors and simultaneously created a distinct feature-category for missing data.

3) Encoding: Data encoding is involved converting each value in a column to a number. Consider a dataset this change based on several points as if our feature is Categorical variables or Continuous variables. . . etc.

- In reference [15] they went the categorical data is to convert into numerical form with the help of label encoder. Then one hot encoder is used to break the relation between the values obtained through the label encoder.
- Retiled to other [16] convert the categorical features into numerical, the normalization of the dataset with min-max normalization technique, then we label the multiclass classification to be started with 0 to 9 categories.
- Related to [17] We convert the nominal features to numerical as most of the machine learning models and scalers can readily work with the numerical values. To convert the nominal to numerical, we use a label encoder implemented as in scikitlearn in Python.

4) Feature Normalization: Feature Normalization is a method used to normalize the range of independent variables or features of data. In data processing, it is also known as data normalization

- As [15] they choose to use the Min–Max scaling Feature Normalization
- In [15] they applied to test six different data transformation algorithms called data normalizer, min-max scaler, robust scaler, standard scaler, quantile transformer, and power transformer were implemented and evaluated for suitability on the numerical input features and finally get to use Min–Max scaling related to best result.
- Related to [16] As they work on neural network application in our dataset, they also choose Min-max normalization helps to build neural networks more consistently.

5) Feature selection: To use machine learning algorithms efficiently, it is becoming increasingly important to perform feature selection [9] is a process of selecting more relevant attributes, and removing irrelevant or less relevant attributes or noisy data or features that do not add additional value to a machine learn [10] A lot of work has been done on feature selection using traditional techniques like Information Gain, the Gini Index, uncertainty, and correlation coefficients [11] There are total of 49 features including packet-based and flow-based features were extracted from the raw network packets by Argus and Bro-IDS tools.[1] Packet-based features are extracted from the packet header and its payload (also called packet data). In contrast, flow-based features are generated using the sequencing of packets, from a source to a destination, travelling in the network.[7]

- Basic distribution given for UNSW-NB15 [12] Six categories have been defined, i.e., basic, time, content, flow, additional generated and labels. The features generated additionally are calculated from fow, basic, content and time features.[12]
- Authors in [13] employ Association Rule Mining (ARM) methodology to define a relation between two or more features for selecting features with the highest rank. Te features are selected by comparison of UNSW-NB15 and KDD 99.
- As referring to [8] that I see, it is most powerful to get throw many steps in feature selection
  First step, they used feature importance technique using RF algorithm, in this step Five features were removed during data type resolution.
  Second step, From the graph in Fig.1, it is observed that there are few features which are contributing most toward classification This can cause overfitting [8] so they removed feature that have importance above 0.05. after the previous steps, feature reached to 37 features is shown in Fig.1.
  The third step, is feature clustering, the remaining features are then clustered according to protocols. Clustering can be done according to packet and fow-based features [13] they made six clusters, i.e., fow, DNS, HTTP, File Transfer Protocol (FTP), MQTT and TCP. MQTT has been developed as a basic message protocol for equipment with restricted bandwidth. It is, therefore, ideal for IoT applications. the features in this cluster are also shown in Fig.2. Feature importance was calculated and top five and top six features from fow / MQTT and TCP were kept, respectively. The feature importance graphs of fow/MQTT, TCP.

C. Methodology

In [17] Some of the classification algorithm that most commonly used to classify the dataset are SVM, J48, Random Forest, CART and Naïve Bayes. The experiment was carried out using five popular and efficient machine learning algorithms (RF, J48, SVM, CART, and Naïve Bayes). The correlation feature selection approach was used to reduce

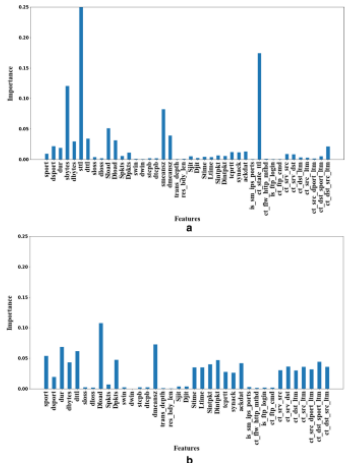

Fig. 1

Number	Full features	Flow/MQTT features	TCP features	Top features (flow/MQTT and TCP)
1	sport	dur	sport	dur
2	dsport	Stime	dsport	dmeansz
3	dur	Ltime	dbytes	Dload
4	dbytes	Dload	sloss	Ltime
5	dttl	dmeansz	dloss	Stime
6	sloss	is_sm_ips_ports	Spkts	Dpkts
7	dloss	ct_srv_src	Dpkts	dbytes
8	Dload	ct_srv_dst	swin	Sintpkt
9	Spkts	ct_dst_ltm	dwin	Dintpkt
10	Dpkts	ct_src_ltm	stcpb	sport
11	swin	ct_src_dport_ltm	dtcpb	dsport
12	dwin	ct_dst_sport_ltm	Sjit	
13	stcpb	ct_dst_src_ltm	Djit	
14	dtcpb		Sintpkt	
15	dmeansz		Dintpkt	
16	trans_depth		tcprtt	
17	res_bdy_len		synack	
18	Sjit		ackdat	
19	Djit			
20	Stime			
21	Ltime			
22	Sintpkt			
23	Dintpkt			
24	tcprtt			
25	synack			
26	ackdat			
27	is_sm_ips_ports			
28	ct_flw_http_mthd			
29	is_ftp_login			
30	ct_ftp_cmd			
31	ct_srv_src			
32	ct_srv_dst			
33	ct_dst_ltm			
34	ct_src_ltm			
35	ct_src_dport_ltm			
36	ct_dst_sport_ltm			
37	ct_dst_src_ltm			

Fig. 2

feature complexity, yielding just 13 features in the NSL-KDD dataset. This study evaluates the NSL-KDD dataset's performance in detecting real-world anomalies in network activity. Five classic machine learning models, RF, J48, SVM, CART, and Naïve Bayes, were trained on all 41 characteristics against the five standard types of attacks, DOS, probe, U2R, and R2L, and achieved average accuracies of 97.7 percent, 83 percent, 94 percent, 85 percent, and 70 percent, respectively. The same classifiers are trained again with the reduced 13 features, yielding average accuracies of 98 percent, 85 percent, 95 percent, 86 percent, and 73 percent for each classifier.

In [18] They built a basic deep neural network that had an input layer, three hidden layers, and an output layer. Six is the input dimension, while two is the output dimension. The neurons in the hidden layers are twelve, six, and three, respectively. They used different values for learning rate they achieved a good result with 0.001 which gives 75.75 percent in accuracy, 83 percent in precision, 76 percent in Recall and 75 percent in F1-Score.

In [19] they use a XGBoost algorithm with NSL-KDD data set. They split the data set to train and test with test size equal to 0.33. they also tuned hyperparameters of the algorithm to achieve best results. the XGBoot with the test dataset give them an accuracy of 98.7 percent,98.4 percent in precision, 99.1 percent in recall and 98.76 percent in F1-score. They make a comparison between XGBoost and other algorithms, and they found XGBoost is the best one.

In [20] they use a SVM algorithm integrated with the principal component analysis (PCA) algorithm. In this technique, the NSL-KDD dataset is used to train and improve the model for recognizing anomalous patterns. To handle the diversity of data scale ranges with the fewest misclassification errors, a Min-Max normalization technique was presented. The PCA algorithm is selected as a statistical technique to make dimensionality reduction for the NSL-KDD data set. They achieve accuracy of 95 percent using 31 features in the data set.

In [21] they employed supervised learning, binary and multi-class classification on the data, and techniques such as RF, SVM, and ANN. Multiple DT are employed in RF to produce data predictions. Each tree predicts a class, with the most trees predicting the same class as our model prediction. A split hyper plane characterises SVM. It's a classifier that can tell the difference between two things. It allows you to quickly categorise fresh data instances depending on their placement in relation to the divide (or split). The biological rationale for ANN stems from the nervous system's structure. They achieve accuracy for binary classification 98.6 percent with RF,97.6 percent with SVM and 94.7 Percent with ANN.

In [22] Their research was divided into two primary phases, with the following machine learning approaches being considered: ANN, LR, kNN, SVM, and DT. For the binary setup and the multi-class configuration, they used the UNSW-complete NB15's feature space (42 features) in the initial stage of the studies. They introduced the feature selection approach using the XGBoost algorithm in the second phase. Following that, a reduced feature vector with 19 attributes was created. They carried out the experiments using the best feature vector, which included a binary classification phase as well as a multi-class classification technique.

III. Our Work

A. Data set

UNSW-NB15 is a network-based anomaly detection dataset that was created in 2015. It has 45 attributes, including ID and label. It has over 1 million records and is highly recommended for anomaly detection problems. Attributes define the manner of a network and describe how 10 behaviors, including the normal one, and 9 attack and analysis categories, are happening during a network session.

Some attributes have generous observations, but others have just zero or one value, such as ct_flw_http, ct_ftp_cmd, and ct_dst_sport_ltm. In our approach, we have dropped these attributes as they have a slight impact on the whole dataset. In order to reduce dimensionality, we have dropped 9 features, including ID and attack_cat.

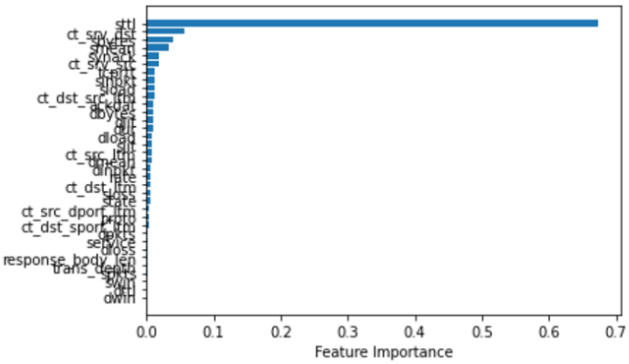

Fig. 3: Feature extraction

We used over 175 thousand records with an 80/20 train-validation ratio and managed to achieve dimensionality reduction by using only 4 features. UNSW-NB15 has five categorical attributes such as attack_cat, state, and proto. and 40 numeric attributes. The categorical attributes were encoded before any preprocessing of the dataset.

B. Feature extraction

In order to achieve higher and higher results, feature extraction and the analysis of the dataset have become a must in recent days. It is crucial to be aware of every aspect of your data. Is it balanced? Are your records in tune? And even the interrelationships between each feature.

In our proposed algorithm, we have had a sight to be aware of the possibility of getting an overfitted model if we have not extracted our most valuable features well. and even more to underfit if we could not manage these features after extraction. Thus, we decided to use some external resources to see and observe our dataset's UNSW-NB15 features. and have found some interesting results, as in Fig.3.

Four Features only were enough to those to achieve their goals to detect abnormality with better accuracies using a combination fusion of Decision Tree and Random Forest Classifiers. While others were interested in seeing the differences between the old KDD and the branded version of its UNSW-NB15.

Our proposed algorithm is a hybrid of Random Forest Classifier and Random Forest Regressor to rank the features according to their impact on other features in the dataset and to extract the most relevant four features from the whole dataset in order to achieve higher results, especially to give the machine learning guys the variety they need to accomplish their missions too.

C. Data Preprocessing

In data preprocessing stage, we go throw several steps.

First one, our dataset UNSW-NB15 have imbalance classes as shown in Fig.4. We traded many methods to handling and deal with this problem as Condensed Nearest Neighbour CNN,

Method	Test	Val	train
RandomOverSampler	82	68	73
RandomUnderSampler	75	74	75
CNN	97	89	89
SMOTE	96	85	85
Tomek Links	91	84	93
Cluster Centroids	59	58	59

TABLE I: Data Preprocessing

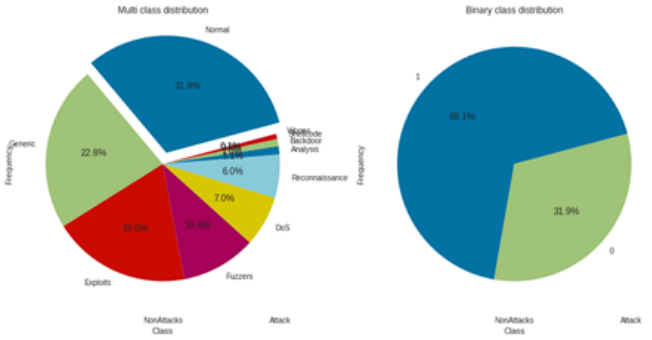

Fig. 4: Data Distribution

Cluster Centroids, Tomek Links, Random Under Sampler to under sampling majority Class and used Random Over Sampler, SMOTE to oversampling, minority class, but finally all these trays result shown in Table 1. But no one of this method can improve basic random forest model performance. We moved to use another way to try to balance data that fit in model to pass data into two models, first one binary model to check if test record is attacker or benign. Get all recorded that predicted as attack to multi class classifier model without normal data. So, by this way, we minimize the difference between class by remove Normal category.

Second stage, handing missing data we noticed there are some missing data as Null and NAN in categorical and numerical feature we deal with numerical column by replace missing data by mean value of the whole column, and deal with categorical feature by drop them, we believe dropping it is better than replace by one of values is same column that by change the distribution of the original data. Another handling in feature "service" there are some recorded had '-' in this value that dispersion multi classification model. So, we dropped this record.

Third stage, encoding we used two type of encoding method, one hot encoding (dummy) that applied on all categorical feature that "proto", "service", and "state". And apply label encoding in "label" and "attack_cat" feature.

Fourth stage, normalization, that used to normalize the range of independent variables or features of data. We applied three method min-max scaler, robust scaler, and standard scaler. And we choose min-max scaler normalization that get the best performance method in our case.

D. Methodology

For our methodology, we chose XGBoost, Decision Tree, Random Forest, and an ensemble technique called "Stacking",

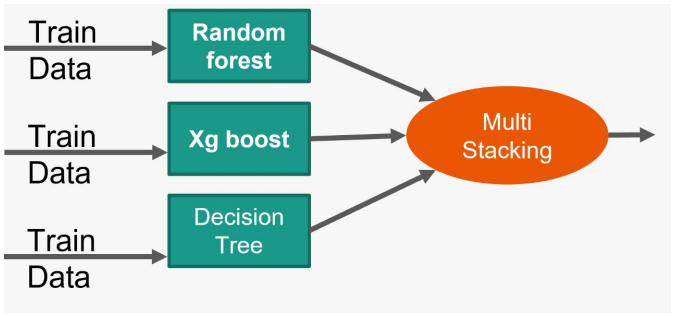

Fig. 5: Methodology

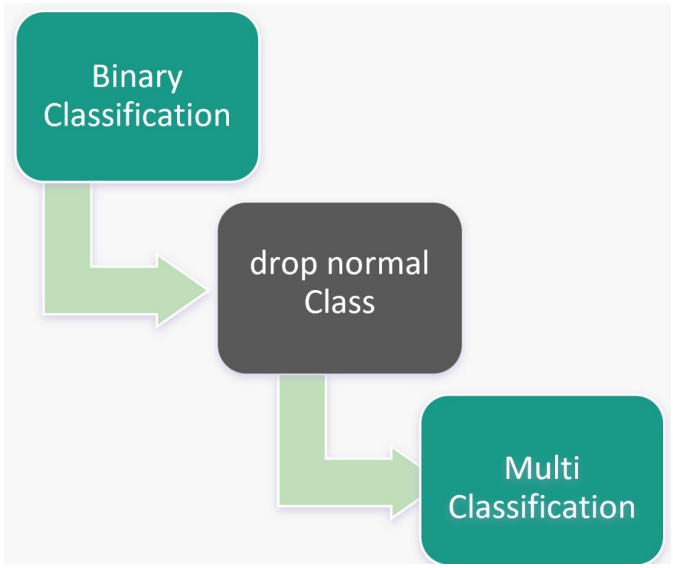

Fig. 6: Methodology

we used stacking with all models.

From previous research papers, we found that the best results obtained with the UNSBW-NB15 dataset used tree-based models such as Decision Tree, Random Forest, and XGBoost, among others. We wanted to try a new technique with these models that was not used in the previous research, so we used an ensemble technique called the "Stacking" technique.

we found that the data was imbalanced, and we tried many of sampling techniques but all of them will damage the data, not solve the problem because the count of class was so different. so, we thought to make two models, one for the binary class (Normal, Abnormal) and one for multi classes of attacks as in Fig.5.

Finally, we merge the two results as in Fig.6.

- Models descriptions:
  A decision tree is a machine learning model algorithm that partitions the data into subsets. The partitioning process starts with a binary split and continues until no further splits can be made. Variable lengths are formed. It is also used for regression problems and classification problems.

Random forest is a machine learning model algorithm that is also used in classification and regression problems. It builds decision trees on different samples.

The XGBoost algorithm is a gradient boosting algorithm, a common technique in ensemble learning. It has some features like execution speed and model performance.

Stacking is an ensemble machine learning algorithm. It uses a meta-learning algorithm to learn how to best combine the predictions from two or more base machine learning algorithms. We used a stacking algorithm to combine the Decision tree, Random Forest and XGBoost.

- Hyperparameters Tuning:
  We used the "Grid Search" method with all models for hyperparameter tuning, for Random Forest, we selected max_depth, min_samples_leaf and min_samples_split parameters, and we get that the best values with the binary problem were (max_depth=20,min_samples_leaf=4,min_samples_split=5) and for the multi-class problem was (max_depth=12,min_samples_leaf=2,min_samples_split=4). For Decision Tree we select max_depth, min_samples_leaf and min_samples_split to be tuned also by using the Grid Search technique, and we get that the best values were (max_depth=12,min_samples_leaf=3,min_samples_split=7) for the binary problem, and for multi-class problem (max_depth=12,min_samples_leaf=7,min_samples_split=2). For XGBoost we used the default hyperparameters with the binary problem and for the multi class problem we used these hyperparameters by trial and error until reach the best result, so these parameters were (n_estimators=50,max_depth=20).

IV. Results

Finlay, we start training the models with the train dataset and the validation dataset for the binary and multi classes problems to found best one of the four models to be used on testing, as you can see the results for all stages in Table.2.

We used four evaluation matrices which are (Accuracy, Precision, Recall and F1-Score)

Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$

Precision = $\frac{TP}{TP+FP}$

Recall = $\frac{TP}{TP+FN}$

F1 = $\frac{2*Precision*Recall}{Precision+Recall} = \frac{2*TP}{2*TP+FP+F}$

Also, we used the ROC curve for evaluation, as you can see in Fig.7 and Fig.8.

V. Conclusion

In the end, we can see that the dataset of UNSBW-NB15 needs more data preprocessing to give better results for handling the categorical features and making better feature selection.

When compared to related works, our model's results are not the best, but perhaps using different algorithms, such as SVM or KNN, will give us better results, because all algorithms are tree-based models, so we can see that stacking did not improve the results because nearly all the models give the same prediction. As a result of our research, we can conclude that the XGBoost is the best of the experiments.

Random Forest				
Stage	Accuracy	Precision	Recall	F1-Score
Validation	0.98	0.99	0.99	0.99
Test	0.89	0.91	0.78	0.84
Decision Tree				
Stage	Accuracy	Precision	Recall	F1-Score
Validation	0.98	0.98	0.98	0.98
Test	0.89	0.82	0.85	0.83
XGBoost				
Stage	Accuracy	Precision	Recall	F1-Score
Validation	0.98	0.99	0.99	0.99
Test	0.925	0.95	0.86	0.90
Stacking				
Stage	Accuracy	Precision	Recall	F1-Score
Validation	0.98	0.99	0.98	0.98
Test	0.91	0.93	0.83	0.87

TABLE II: Results of Models for Binary Problem

Random Forest				
Stage	Accuracy	Precision	Recall	F1-Score
Validation	0.94	0.94	0.94	0.93
Test	0.90	0.92	0.91	0.91
Decision Tree				
Stage	Accuracy	Precision	Recall	F1-Score
Validation	0.94	0.93	0.94	0.93
Test	0.88	0.89	0.87	0.87
XGBoost				
Stage	Accuracy	Precision	Recall	F1-Score
Validation	0.95	0.94	0.95	0.94
Test	0.95	0.95	0.95	0.94
Stacking				
Stage	Accuracy	Precision	Recall	F1-Score
Validation	0.95	0.94	0.95	0.94
Test	0.92	0.93	0.91	0.92

TABLE III: Results of Models for Multi Problem

VI. Future work

During the work on this project, we encountered some tips compared to the results of the related work that we mentioned. Data preprocessing using Recursive Feature Elimination in data cleaning and trying to replace a missing categorical feature with any technique will better than drop it. We can

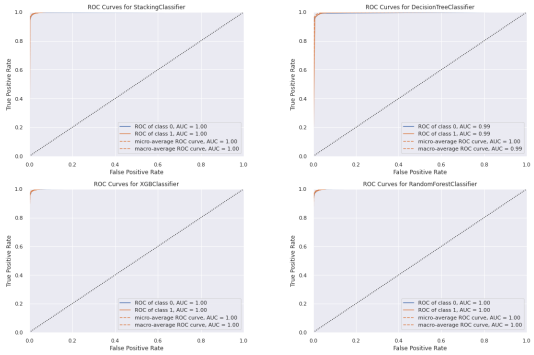

Fig. 7: ROC Curves for Binary Problem in Validation

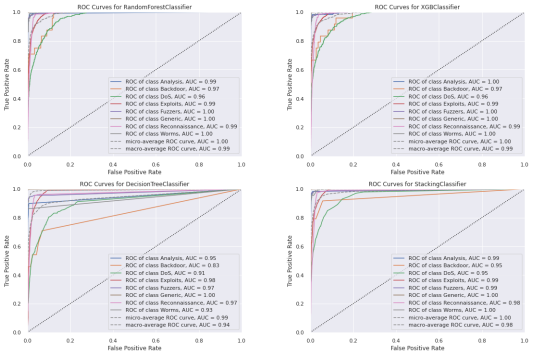

Fig. 8: ROC Curves for Multi Classes Problem in Validation

apply filters to remove outliers in data we think will help the model.

And finally, we used four algorithms and applied Stacking between them. If you use another strategy for combination, such as bagging, boosting, or moving to a deep learning level, you will probably get better performance.

REFERENCES

[1] Hamid, Yasir & Ranganathan, Balasaraswathi & Journaux, Ludovic & Muthukumarasamy, Sugumaran. (2018). Benchmark Datasets for Network Intrusion Detection: A Review. International Journal of Network Security. 10.6633/IJNS.2018xx.20(x).xx).

[2] Markus Ring, Sarah Wunderlich, Deniz Scheuring, Dieter Landes, Andreas Hotho, A survey of network-based intrusion detection data sets, Computers & Security, Volume 86, 2019,Pages 147-167,ISSN 0167-4048,https://doi.org/10.1016/j.cose.2019.06.005.

[3] Ferriyan, Andrey & Thamrin, Achmad & Takeda, Keiji & Murai, Jun. (2021). Generating Network Intrusion Detection Dataset Based on Real and Encrypted Synthetic Attack Traffic. Applied Sciences. 11. 10.3390/app11177868.

[4] Moustafa, Nour & Slay, Jill. (2015). The significant features of the UNSW-NB15 and the KDD99 Data sets for Network Intrusion Detection Systems. 10.13140/RG.2.1.2264.4883.

[5] Sarika Choudhary, Nishtha Kesswani, Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT, Procedia Computer Science, Volume 167,2020,Pages 1561-1573,ISSN 1877-0509,https://doi.org/10.1016/j.procs.2020.03.367.

[6] Binbusayyis, A.; Vaiyapuri, T. Comprehensive Analysis and Recommendation of Feature Evaluation Measures for Intrusion Detection. Heliyon 2020, 6, e04262.

[7] Z. Zoghi, G. Serpen "UNSW-NB15 Computer Security Dataset: Analysis through Visualization", "University of Toledo, Ohio, USA", Jan 2021

[8] M. Ahmad, Q. Riaz, M. Zeeshan, H. Tahir, S. Haider and M. Khan "Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW NB15 data set", J. Wireless Com Network, oct 2021

[9] S. Bagui, E. Kalaimannan, S. Bagui, D. Nandi and A. Pinto, "Using machine learning

[10] P. Yang, Q. Zhu, "Finding key attribute subset in dataset for outlier detection" Known Based Syst. Feb 2011

[11] G. Forman, "An extensive empirical study of feature selection metrics for text classification", J. Mach Learn Res. , Mar 2003.

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

[13] N. Moustafa, J. Slay, "The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems", in 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS) pp. 25–31, Nov 2015

[14] S. Al-Emadi, A. Al-Mohannadi and F. Al-Senaid, "Using Deep Learning Techniques for Network Intrusion Detection," 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), May 2020.

[15] G. Kocher and G. Kumar "PERFORMANCE ANALYSIS OF MACHINE LEARNING CLASSIFIERS FOR INTRUSION DETECTION USING UNSW-NB15 DATASET", Conference: 2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC), Dec 2018.

[16] A. ALEESA, M. YOUNIS., A. MOHAMMED and M. SAHAR, "DEEP-INTRUSION DETECTION SYSTEM WITH ENHANCED UNSW-NB15 DATASET BASED ON DEEP LEARNING TECHNIQUES" Journal of Engineering Science and Technology, Feb 2021

[17] S. Revathi and A. Malathi, "A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection." [Online]. Available: www.ijert.org

[18] Tuan A Tang, Lotfi Mhamdi, des McLernon, Syed Ali Raza Zaidi, and Mounir Ghogho, Deep Learning Approach for Network Intrusion Detection in Software Defined Networking.

[19] S. S. Dhaliwal, A. al Nahid, and R. Abbas, "Effective intrusion detection system using XGBoost," Information (Switzerland), vol. 9, no. 7, Jun. 2018, doi: 10.3390/info9070149.

[20] S. T. Ikram and A. K. Cherukuri, "Improving accuracy of intrusion detection model using PCA and optimized SVM," Journal of Computing and Information Technology, vol. 24, no. 2, pp. 133–148, 2016, doi: 10.20532/cit.2016.1002701.

[21] Ahmad, M., Riaz, Q., Zeeshan, M. et al. Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set. J Wireless Com Network 2021, 10 (2021).

[22] Kasongo, S.M., Sun, Y. Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset. J Big Data 7, 105 (2020).