Network Intrusion Detection

Nada Aboelfetouh [a], Esraa Badawi [b], Salma Sultan[c], Yomna Ahmed [d]

[a] Electrical Engineering Department, Cairo, Egypt, naboe016@uottawa.ca

[b] Electrical Engineering Department, Cairo, Egypt, ebada030@uOttawa.ca

[c] Electrical Engineering Department, Cairo, Egypt, ssult103@uottawa.ca

[d] Electrical Engineering Department, Cairo, Egypt, yahme022@uottawa.ca

Abstract

Network intrusion is an illegal activity in which some intrusion can come from outside or inside. Their effect on the network was defacing the website with various kinds of messages or crud images. Others can be sought to extract information. The protection of a network from unauthorized access is made through a build system to detect these malicious and network security.

The network-based intrusion detection system is a passive device that does not interfere with traffic monitoring. It includes NIDS sniffs that send alerts to a NIDS management server.

The goal of the project is to develop a model to detect network intrusion by using machine learning algorithms. Thereby we develop different models and evaluate them by accuracy and recall to not miss any malicious, so we focus on high recall of class 1.

After evaluation, the best five models are SVM, Random Forest, Logistic Regression, Decision Tree, and MLP.

The following step is to build the Stacking algorithm and voting algorithm with hyperparameter and without the hyperparameter.

The champion model was a stacking algorithm (Random Forest, Decision Tree, SVM) with default parameters that Accuracy equals 80% and recall of 84%.

Keywords: Network intrusion Detection, Machine Learning algorithm, Ensemble Models, Stacking Algorithm, Voting Algorithm.

1. Introduction

In recent years, the networking revolution has fully matured. More than ever, we can see how the Internet has changed the way that computing is done. Sadly, just as there are countless options and opportunities, there are also countless threats and opportunities for hostile invasions.

A system's security controls must be configured to prevent unauthorized access to its data and resources. To completely prevent security breaches at this time, nevertheless, would be impracticable. We can work to spot these intrusion efforts, though, so that measures can be taken to lessen the damage that results.

Intrusion detection is the name of this field of research. A network intrusion detection system, which is a system that monitors risky activity in network data and sends out notifications when it is found, will be used to detect and respond to unknown threats.

The identification method used in our project involves applying the dataset to machine learning models such as Random Forests, Adaboost and Support Vector Machines, GaussianNB, Logistic Regression, Decision Tree, and MLP.

A collection of real-world networks with "malicious" and "benign" labels are included in the applied dataset, which aids in the development of a reliable model that makes accurate predictions when compared to the real world.

2. Related Work

Intrusion detection systems (IDS) is an essential part of network security because it vulnerable to attacks these days. This paper[1] aims to consider the computation time in their methodology because the accuracy of the model is already high beside increasing of the network IDs system should get along with the high inflow on network connection. Therefore, we need a classifier model with high accuracy.

As a crucial component of system defense, IDSs use network traffic data they collect from a specified place on the network to protect the network. Network intrusions are increasingly being detected using machine learning techniques, which further empowers the network administrator to take precautionary action. To effectively detect network intrusions, they suggest using ten machine learning algorithms, For KDD Cup, they use the widely used NSL-KDD dataset. They evaluated the suggested machine learning techniques for network intrusion detection. The detection rate, false positive rate, and average cost for classification misclassification are among the final experimental results with five classes that are displayed. These are used to aid in the knowledge acquisition of researchers in the field of network intrusion detection.

There are three dangers that can put systems at risk of hacking: incursions from outside the system, such as unauthorized users, Internal hacks include authorized system users who utilize the system improperly, followed by errant users who attempt to abuse their access credentials .

The two main categories of intrusion detection techniques are abuse detection and anomaly detection. All known sorts of assaults (intrusions) can be found using a misuse detection system by examining the established intrusion patterns in system audit flow[2].

3. Research Objective

The main goal of our project to apply different classifier model for instance isolation forest, Random Forest and Adaboost, MlP, logistic Regression, decision tree , SVM and gaussian naïve bayes to detect malicious also we apply stacking algorithm and voting algorithm in the best model before and after hyperparameter tuning.

4. Dataset

Our dataset is the KISTI+IDS2021-CDMC data, which includes 153,829 IDS alerts labeled as either malicious (label: 1) or benign (label: 2). The data is divided into two parts: training data with labels and testing data without labels. Data includes a word vector column where every row consists of unbalanced number of nested list and each list is 1*100 dimensions. For instance, the first row contains 19 nested lists, but second row has 25. That needs to be converted from many nested lists to a single (1 x 100) feature by selecting the maximum value from the nested lists inside each row, representing the most important word. After transforming the data, we found that the target was balanced.

```
1    61520
0    61520
Name: label, dtype: int64
```

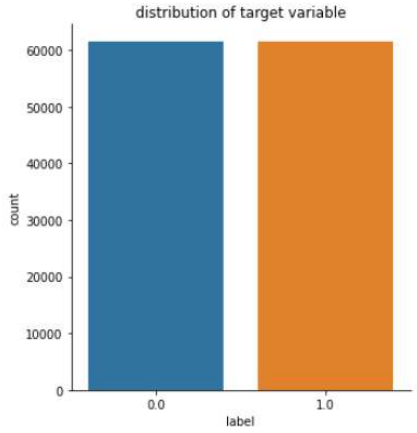

Figure 1 (Distribution of target Variable)

```
0         [[-0.507, -0.49518, 0.46885, 0.54524, -0.11552...
1         [[0.19911, -0.46156, 0.19674, -1.3298, 0.51805...
2         [[-0.7403, -0.78746, 0.47018, 0.43474, 0.05842...
3         [[-0.44257, -0.54624, 0.25403, 0.80731, 1.026,...
4         [[0.19911, -0.46156, 0.19674, -1.3298, 0.51805...
                                ...
123035    [[-0.081545, 0.25175, 0.027983, 0.064531, 0.25...
123036    [[-0.35721, -0.54399, 0.26479, -0.15312, -0.00...
123037    [[-1.5145, 0.9682, 0.34735, 1.0024, 0.042789, ...
123038    [[0.19911, -0.46156, 0.19674, -1.3298, 0.51805...
123039    [[0.19911, -0.46156, 0.19674, -1.3298, 0.51805...
Name: word_vector, Length: 123040, dtype: object
```

	0	1	2	3	4	5	6
0	0.744250	0.314920	1.15810	1.82440	1.57700	0.45372	0.680690
1	0.199110	-0.461560	0.19674	-1.32980	0.51805	1.27910	0.408520
2	1.264000	0.093259	1.15810	0.46354	1.54510	0.91477	0.556690
3	-0.000677	0.300840	0.47018	1.68840	1.13670	1.13270	0.808910
4	0.199110	-0.461560	0.19674	-1.32980	0.51805	1.27910	0.408520
...	...	...	...	...	...	...	...
123035	0.744250	0.595200	1.15810	1.05870	0.59834	0.87438	0.808910
123036	-0.000677	-0.543990	0.87098	1.68840	0.14309	0.43380	-0.040783
123037	1.264000	0.968200	0.93292	1.68840	1.54510	0.46053	0.162770
123038	0.199110	-0.461560	0.19674	-1.32980	0.51805	1.27910	0.408520
123039	0.199110	-0.461560	0.19674	-1.32980	0.51805	1.27910	0.408520

123040 rows × 100 columns

Figure 2 (Data before Transformation) **Figure 3 (Data After Transformation)**

As a final step in preprocessing, we normalized the data using the Minmax Scaler from Sklearn. The figure below shows the range of each feature before and after normalization. Before normalization, the features had different ranges and distribution is restricted in a small range, but after normalization, they are all ranged from zero to one and the distribution of features got rid of skewness. Each color in the graph after normalization represents the different values of features.

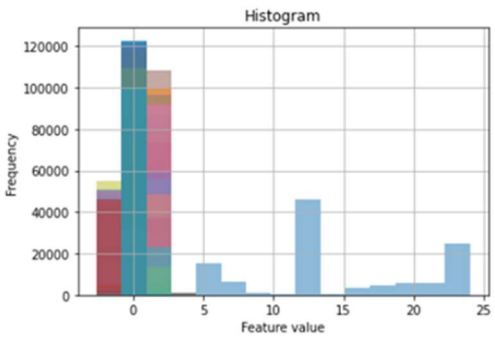

Figure 4 (Data before normalization)

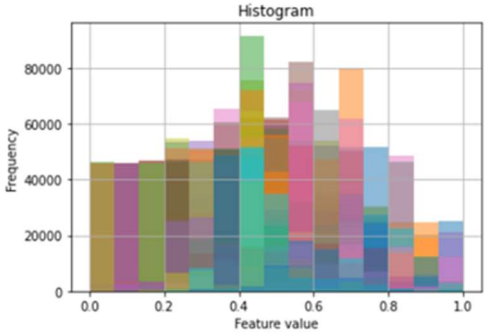

Figure 5 (Data after normalization)

5. Methodology

5.1. Cross Validation

It has a parameter called K that the data will split depending on the number of K, it helps in see how the model will perform in general.

In our case we choose the K = 5 because usually the k =5 or 10 but there is no rule for it.

The model that we apply divides into:

5.2. Classification model

- The Support vector machines are supervised learning models that performs classification by finding the hyperplane that maximizes the margin between two classes. in our case, the two classes are 1(malicious) and 0(benign) after mapping them.

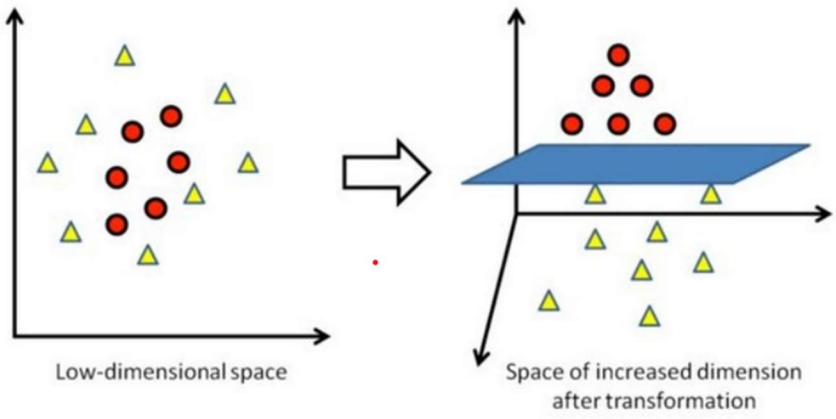

Figure 6(The data before margin and after)

SVM has been successfully applied in many application areas also an intrusion Detection System has an efficient way to detect intrusions.

It requires training data and labels Therefore we chose SVM in our case.

- Gaussian Naive Bayes Supervised Classifier is the model using Bayes' theorem. the presence of one value of a feature doesn't affect the presence of another. its performance increase with the growth of the training set
- Isolation Forest design specifically for anomaly detection by

  using binary tree. The data randomly sub-sampled in tree structure. The sample in short branches indicates anomalies[8]
- Decision Tree is the most popular tool for classification following the tree structure that represents decision and decision making. How it works the data is split according to a certain parameter. The leaves are the final outcome, and the decision node is where data is split[3].

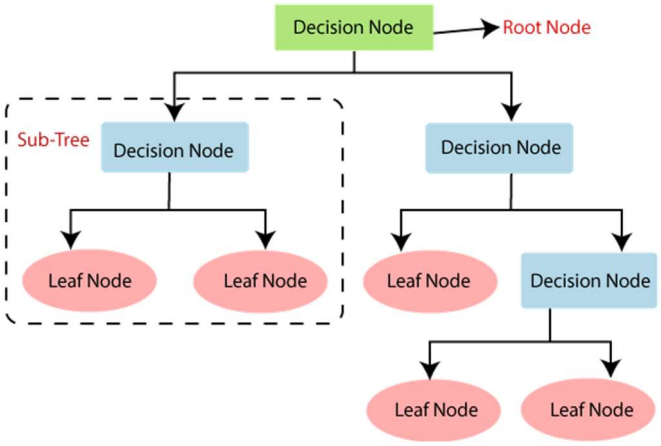

Figure 7 (Decision Tree)

- Logistic regression under the supervised technique needs the outcome to be a categorical or discrete value but it gives probabilistic values which lie between 0 and 1. It has many types like Binary or Binomial, Multinominal, and Ordinal. In our case, we use the binary because it is a dependent variable having only two possible types, either 1 or 0[4].

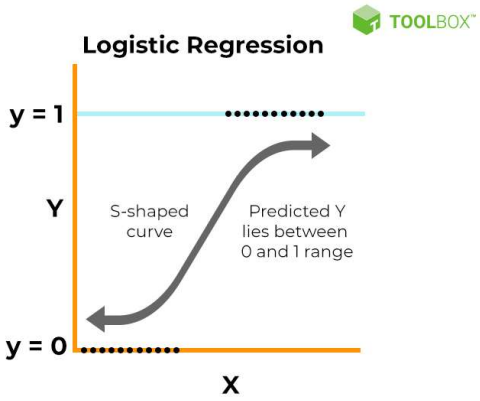

Figure 8 (Logistic Regression)

- Multi-layer Perceptron classifier is different from SVM and other classification models because it relies on an underlying Neural Network to perform the task of classification. It learns the characteristics of the data. It consists of the First layer where input data will feed, the last layer which takes from it the output and the hidden layer we can increase as much as we want. Beside classifier models we want to see how the effect of the simply neural network model[5].

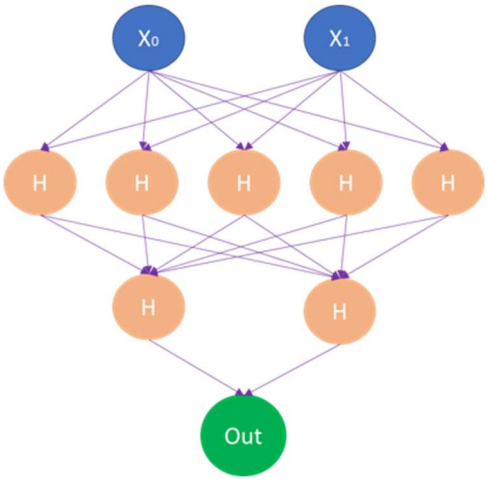

Figure 9 (MlP Layers)

5.3. Ensemble model

Because in many cases one model isn't enough to make the perfect prediction, an ensemble model combines multiples models to reduce the model error and maintain the generalization of models[6].

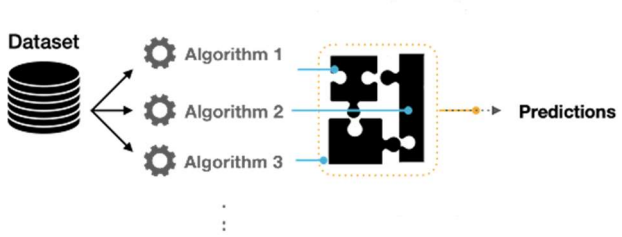

Figure 10: Diversify the model predictions using multiple algorithms

5.3.1. Bagging

Is the one of ensemble technique the basic idea that each model learns the error from the previous model using a different subset of the training data set. The Random Forest algorithm is an example of bagging technique.

- Random Forest likes its name implies. Each individual tree in it gives a class prediction and the class with the most votes become the model prediction. The feature is implemented in random mode.

5.3.2. Boosting

It works by creating a strong learner by adding iteratively weak learner with weights that reflect its accuracy. After that combine all learners using a weighted – averaging strategy.

- AdaBoost Algorithm the goal of it to reach a form of additive model composed of k weak models.

5.3.3. Stacking Algorithm

The third technique of ensemble algorithms, the main point of stacking is finding the space of different model of the same problem. It uses the prediction of each weak learner to build intermediate prediction after that adding new model that learns from the intermediate prediction the same target.

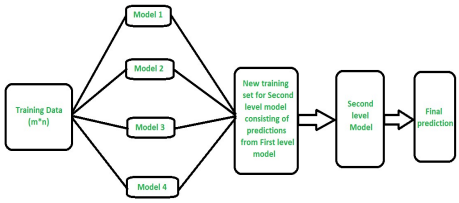

Figure 11 : How Stacking Algorithm Work

We use stacking algorithms on Random Forest, Decision Tree and SVM before and after Hyperparameter tuning.

5.3.4. Voting Algorithm

It aggregates the result of each classifier passed into a voting classifier and predicts the output class depend on the highest majority of voting. We applied it to Random Forest, Logistic Regression, Decision Tree, MLP and SVM.

5.4. Hyperparameter tuning

Is searching for the ideal model architecture to control the behavior of machine learning model to minimize the[7] loss Function.We apply Hyperparameter tuning in the best five model.

Models	Best Hyperparameter
SVM	kernel: rbf gamma: 1
Random Forest	n_estimators: 10 max_depth: 30
Decision Tree	min_samples_split: 5 min_samples_leaf: 3 max_depth: 8 criterion: Gini
MLP	solver: Adam max_iter: 150 learning_rate: constant hidden_layer_sizes: (150, 100, 50) alpha: 0.0001 activation: relu
Logistic Regression	penalty: l2 C: 100

Table 1: The Hyperparameter of Models.

5.5. Performance evaluations

We evaluate our model by two metrics

5.5.1. Accuracy

Is the simplest metric to use equal

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ number\ of\ predictions}$$

Figure 12: the form of accuracy

5.5.2. Recall

It attempts to determine the percentage of actual positives that were mistakenly detected.

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

Figure 13: the form of Recall

5.5.3. F1 Score

It measures accuracy by combining the precision and recall scores of a model[9]

$$\text{F1 Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Figure 14: The form of F1 score

6. Result and Implications

6.1. Result

After Fit the pre- processing data to our model here the result

Model	Accuracy	Recall (Class 1)
Isolation Forest	0.56	0.80

SVM	0.79	0.83
Random Forest	0.80	0.84
Logistic Regression	0.77	0.81
Decision Tree	0.80	0.84
Adaboost	0.77	0.74
MLP	0.79	0.83
GaussianNB	0.71	0.86

Table 2: The Result of models

The results that we present show that the best Five model was SVM, Random Forest , Logistic Regression ,Decision Tree and MLP

We choose the highest three algorithms to Stacking algorithm are SVM, Random Forest and Decision Tree and the best five model to Hard voting algorithms are SVM, Random Forest and Decision Tree, MLP and logistic Regression without any hyperparameter tuning.

Models	Acc	Recall	F1-score
Stacking Algorithm	0.80	0.84	0.81
Voting Algorithm	0.80	0.84	0.80

Table 3: The result of Stacking Algorithm and Voting Algorithm

After Applying hyperparameter tuning to the five models repeat the step of the stacking algorithm and voting to see the improvement but depend on the result there are no improvement.

Models	Acc	Recall	F1-score
Stacking Algorithm	0.80	0.69	0.78
Voting Algorithm	0.80	0.84	0.80

Table 4: The result of Stacking Algorithm and Voting Algorithm after hypermeter tuning

The champion model was stacking algorithm with default parameters

6.2. Implication

However, network intrusion is the main topic in Cybersecurity. Network intrusion detection is still a challenging task. The most significant thing in our project was the transformation of word vector column also the variety of model that we implemented.

7. Conclusion and Future work

In conclusion, we applied eight traditional machine learning models, which are used as baseline models, and two ensemble learning techniques to the KISTI+IDS2021-CDMC data for anomaly detection. Traditional machine learning models are generally effective for large datasets, and in our evaluation, we found that all the models performed well, particularly the stacking algorithm model in terms of the Recall. We also evaluated the models using three metrics: recall and accuracy. The stacking algorithm model emerged as the top performer and has been saved for future use with different datasets.

Future work:

In future work, it would be interesting to explore the use of deep learning models for anomaly detection on this dataset. Additionally, we plan to test the performance of the saved stacking algorithm model on other datasets to further assess its robustness and generalizability

8. REFERENCES

[1] Mishra, A., Cheng, A. M. K., & Zhang, Y. (2020, October 1). Intrusion Detection Using Principal Component Analysis and Support Vector Machines. IEEE Xplore. https://doi.org/10.1109/ICCA51439.2020.9264568

[2] Panda, M., Abraham, A., Das, S., & Patra, M. R. (2011). Network intrusion detection system: A machine learning approach. Intelligent Decision Technologies, 5(4), 347–356. https://doi.org/10.3233/idt-2011-0117

[3] Gupta, P. (2017, May 17). Decision Trees in Machine Learning. Towards Data Science; Towards Data Science. https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052

[4] Logistic Regression in Machine Learning - Javatpoint. (n.d.). Www.javatpoint.com. https://www.javatpoint.com/logistic-regression-in-machine-learning

[5] Nitin Kumar Kain. (2018, November 21). Understanding of Multilayer perceptron (MLP). Medium; Medium. https://medium.com/@AI_with_Kain/understanding-of-multilayer-perceptron-mlp-8f179c4a135f

[6] *E*nsemble Models: What Are They and When Should You Use Them? | Built In. (n.d.). Builtin.com. https://builtin.com/machine-learning/ensemble-model

[7] Anyscale - What is hyperparameter tuning? (n.d.). Anyscale. https://www.anyscale.com/blog/what-is-hyperparameter-tuning

[8] Isolation Forest | Anomaly Detection with Isolation Forest. (2021, July 26). Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/07/anomaly-detection-using-isolation-forest-a-complete-guide/

[9] F1 Score in Machine Learning: Intro & Calculation. (n.d.). Www.v7labs.com. Retrieved December 21, 2022, from https://www.v7labs.com/blog/f1-score-guide