A HYBRID MACHINE LEARNING MODEL FOR NETWORK INTRUSION DETECTION

***[1]Solanke O. O., [2]Adegboyega T. A., [1]Taiwo A. I., [1]Abdullah K-K. A., [1]Ayo F. E., [1]Odule T. J., [1]Hassan S. O., [1]Efuwape B. T.**

[1]Department of Computer science, Olabisi Onabanjo University, Ago-Iwoye, Ogun State
[2]Department of Computer science, Tai Solarin University of Education, Ijagun, Ogun State
*Correspondence: E-mail: solanke.olakunle@oouagoiwoye.edu.ng

ABSTRACT

Intrusion detection is a significant challenge in network security, as it involves detecting unseen attacks in a network or system. In this research, we aimed to build a hybrid machine learning model for intrusion detection using artificial intelligence (AI). To do this, we used the KDD CUP 99 dataset and applied two machine learning algorithms: AdaBoost and Stochastic Gradient Descent Classifier (SGDC). These algorithms were combined to form two hybrid models: SGDC_ADA and ADA_SGDC. The results of our study showed that the SGDC_ADA model had an accuracy of 0.97 and outperformed the ADA_SGDC model, which had an accuracy of 0.96. In addition, the SGDC_ADA model had an average precision of 0.97, average recall of 0.96, and average F1-score of 0.97, while the ADA_SGDC model had an average precision of 0.96, average recall of 0.95, and average F1-score of 0.96. Overall, our research suggests that the SGDC_ADA hybrid model is an effective method for intrusion detection, with high accuracy and low error rates. This model may be useful in improving network security and protecting against unseen attacks.

Keywords: *Network Intrusion, Machine Learning (ML), AdaBoost, SGDC, ADA_SGDC, SGDC_ADA*

INTRODUCTION

Cybercrime is one of the computer crimes that have caused a lot of damages to economic and technological prosperity of the world. The cybercrime constitutes threat to the confidentiality and viability of computer operations particularly the business angle of the operations. Cyber trust is greatly affected by the activities of hackers and cyber terrorists that have been using the developmental initiatives of computer technology to perpetuate their evil act and jeopardize the efforts of honest and sincere computer users.

Rupali et al. (2014) pointed out that as the cost of information processing and Internet accessibility decreases, more and more organizations become vulnerable to a variety of cyber threats. These threats are becoming increasingly complex and sophisticated due to the rapid expansion of the Internet, the complexity of communication protocols, and the anonymity of the Internet. Jones et al. (2000) conducted a survey that found that the rate of cyber-attacks is more than doubling every year in recent times.

The argument suggests that an organization can suffer significant losses if its systems and networks are attacked. Information security should focus on confidentiality, authentication, integrity, availability, and non-repudiation (McCumber, 1991). This means that secure information requires a process that provides protection from intrusions, automatic detection of intrusions, automatic reaction or alarm when the system is compromised, and repair or recovery of losses caused by the intrusion. Among these phases, the accurate detection of an intrusion is the most important (Menahem et al., 2009).

The Intrusion Detection System (IDS) is a security technology that can detect, prevent, and potentially react to computer attacks. It is a standard component of security infrastructures, monitoring target sources of activity such as audit and network traffic data in computer or network systems and using various techniques to provide security services. The IDS can also be seen as a cyber-defense mechanism, working to prevent attacks and detect security breaches at the application, network, host, and data levels (Singh et. Al., 2015).

This paper proposes a solution to improve the quality of anomaly-based IDS detection using hybrid machine learning technique and a dataset for deployment on an experimental network. The goal is to create a more effective network intrusion detection system.

The paper is organized into multiple sections. In Section II, we discuss related research. In Section III and IV, we describe the benchmark datasets, experimental criteria, methodology, and comparative analysis of multiple self-learning approaches on the benchmark datasets. Finally, in the last section, we outline future research work and provide a concluding synopsis.

RELATED WORKS

Intrusion can be defined as any set of actions that threatens the integrity, availability, or confidentiality of a network resource. So on the basis of this Intrusion detection can be defined as the act of detecting actions that attempt to compromise the confidentiality, integrity or availability of a resource (Deepthy et al, 2012)

Palenzuela et al. (2016) show how they designed their neural network by choosing a configuration among several options. They used a binary classifier: their goal is not to classify the attack (DoS, Probe, R2L, U2R), but to determine if a packet corresponds to a malicious activity or a benign one. Their preprocessing stage reduces the number of parameters in the KDD Cup 99 dataset from 41 to 38. They then use this dataset with seven different MLP configurations: from zero to three hidden layers, with different numbers of neurons. This comparison can be used to measure the impact of each topology on the results. Finally, they compare the accuracy of every configuration to conclude that the MLP with a single hidden layer of 10 neurons gives them the best results: a 39-10-2 structure (each number represents here the number of neurons per layer), yielding a 99.85% accuracy and 0.17% false positive rate. The authors used part of the training dataset here to evaluate their performance instead of using the test set provided for this purpose. To improve even further the accuracy of this topology, the authors made a strong trade-off by adding a bias of 0.9999 to the output for attacks. This trick increases the accuracy to 99.99%, but also the false positive rate to 9.83%.

Gupta et al. (2015) propose a new approach of combining SVM and Bee Colony to achieve high quality performance of IDS. Their algorithm is implemented and evaluated using a standard benchmark KDD99 dataset. Experimental results show that SVM with Bee colony achieves an average accuracy of 88.46%.

Guo et al (2016) proposed a hybrid approach toward achieving a high detection rate with a low false positive rate. The approach is a two-level hybrid solution consisting of two anomaly detection components and a misuse detection component. In stage 1, an anomaly detection method with low computing complexity is developed and employed to build the detection component. The k-nearest neighbor's algorithm becomes crucial in building the two detection components for stage 2. In this hybrid approach, all of the detection components are well-coordinated. The detection component of stage 1 becomes involved in the course of building the two detection components of stage 2 that reduce the false positives and false negatives generated by the detection component of stage 1. Experimental results on the KDD99 dataset and the Kyoto

University Benchmark dataset confirm that the proposed hybrid approach can effectively detect network anomalies with a low false positive rate.

In Hu et al (2016), an intrusion detection algorithm based on the AdaBoost algorithm was proposed. In the algorithm, decisions are used as weak classifiers. The decision rules are provided for both categorical and continuous features. By combining the weak classifiers for continuous features and the weak classifiers for categorical features into a strong classifier, the relations between these two different types of features are handled naturally, without any forced conversions between continuous features. Adaptable initial weights and a simple strategy for avoiding overfitting are adopted to improve the performance of the algorithm. Experimental results show that this algorithm has low computational complexity and error rates, as compared with algorithms of higher computational complexity, as tested on the benchmark sample data.

Mazraeh et al (2016) proposed method uses a training set of KDD-Cup99. The proposed method uses three main learning algorithms, SVM, Naive Bayes and the J48 decision tree is implemented and evaluated separately. These algorithms are also implemented and evaluated individually as well. The results show the superiority of the proposed method with 97% efficiency using J48 learning algorithm and AdaBoost classification by reducing the dimension IG method.

Theoretical framework

Stochastic Gradient Descent (SGD) Classifier is a simple yet efficient optimization algorithm used to find the values of parameters/coefficients of functions that minimize a cost function. In other words, it is used for discriminative learning of linear classifiers under convex loss functions such as Support Vector Machine (SVM) and Logistic regression. The structure is as shown in Figure 1.

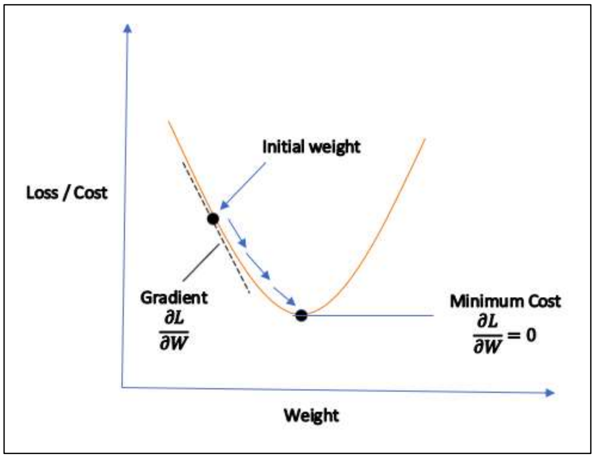

Loss / Cost
Initial weight
Gradient $\frac{\partial L}{\partial W}$
Minimum Cost $\frac{\partial L}{\partial W} = 0$
Weight

Figure 1: Stochastic Gradient Descent (SGD) Classifier Structure.

SGD has been successfully applied to large-scale and sparse machine learning problems often encountered in text classification and natural language processing. Given that the data is sparse, the classifiers in this module easily scale to problems with more than 105 training examples and more than 105 features. The advantages of Stochastic Gradient Descent include efficiency, ease of implementation (lots of opportunities for code tuning).

AdaBoost Classifier

AdaBoost, short for Adaptive Boosting, is a statistical classification meta-algorithm formulated by Yoav Freund and Robert Schapire. It can be used in conjunction with many other types of learning algorithms to improve performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. In some problems it can be less susceptible to the overfitting problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner.

METHODOLOGY

The methodology of this research work involves the use of two algorithms which are Stochastic Gradient Descendent Classifier (SGDC) and AdaBoost Classifier (Chun Guo et al., (2022), Wei Hu, & Weiming Hu. (2005)). This research method is entirely composed of Artificial Intelligence approach which accurately analyze cyber threat on a particular network and conduct a performance analysis using SGDC and Adaboost.

Flowchart of the model

As shown in Figure 2, the first step is to import all the necessary libraries, data scaling and encoding to improve the quality of the dataset. After wards, the improved dataset is passed into WEKA to perform feature selection using InfoGainAttributeEval. The selected features are then passed into Jupyter notebook for training and testing the developed hybrid models ADA_SGDC and SGDC_ADA. After the models are well trained and tested, a comparison analysis is done after which LIME is used to evaluate the best model.

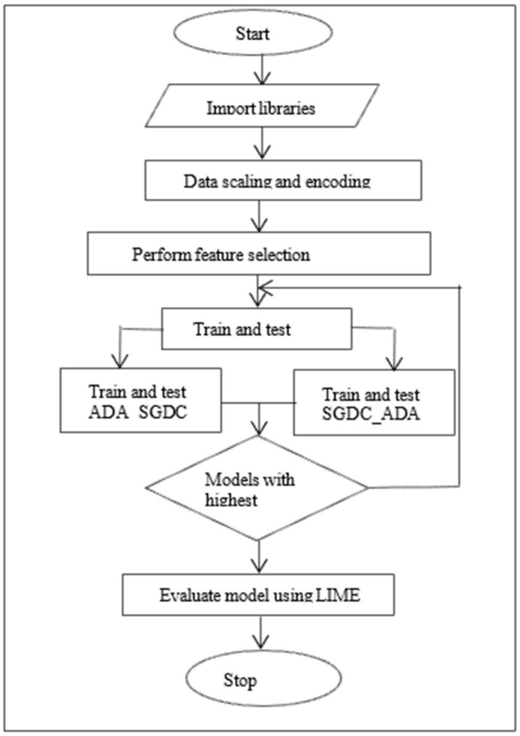


Figure 2: Flowchart used for developing the system.

Justification of KDD Datasets

The intrusion detection methods (supervised or unsupervised) determine the properties of datasets.

KDD Cup99

Knowledge Discovery and Dissemination (KDD) Cup99 is a benchmark dataset most widely used for anomaly based intrusion detection. The dataset created in KDD Cup challenge since 1999. It contains over 4 million network traffic records and 42 attributes or features about protocols (tcp, icmp, udp) connections. The dataset includes 5 million data records that encompass over 21 different types of attacks (e.g. DoS, Guess_passwd, buffer overflow) and comes along with an explicit test subset.

Machine-learning techniques generally rely on a frequency analysis to provide discriminative features; they thus perform poorly on complex data sets (e.g., NGIDS-DS) because of the similarities between attack and normal data, along with the loss of all positional data of the system call traces. Nonetheless, these algorithmic techniques are able to attain high classification accuracy on the KDD99 data set because it is limited to a single process that creates a larger and more detectable system call footprint (Siddique et al., 2019).

RESULTS AND DISCUSSION

Figure 3 shows the relationship between the features of the dataset individually and how they correlate with one another. This map shows the correlation between the features of the dataset. Correlation describes the strength of an association between two variables, is completely symmetrical and ranges from 0 to 1.

Model Performance measure

The three main metrics used to evaluate a classification model are accuracy, precision, and recall. Accuracy is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

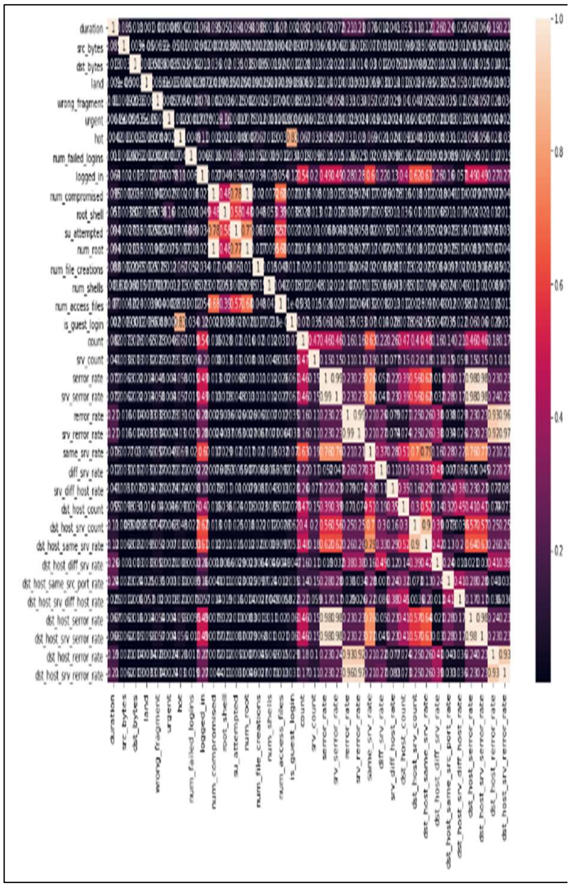
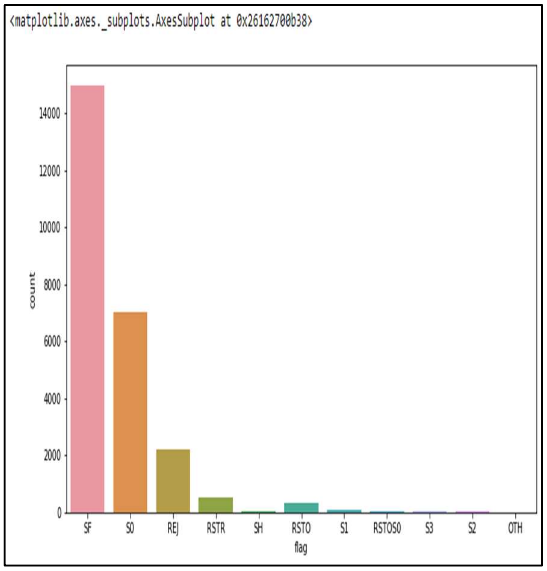
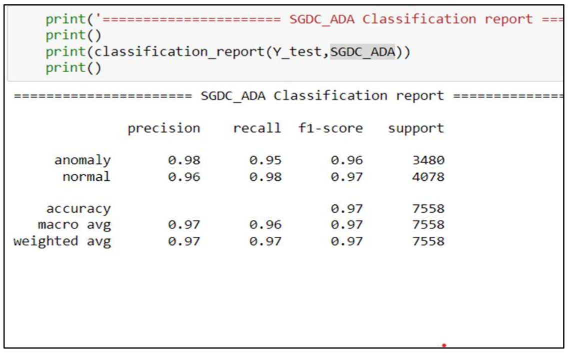


Figure 3: EDA (Heatmap showing the correlation between features of the dataset) (Jupyter)



```
<matplotlib.axes._subplots.AxesSubplot at 0x26162700b38>
```

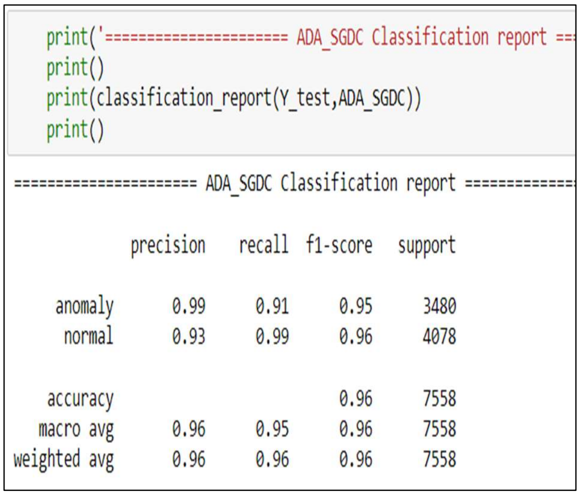
Figure 4: Bar-chart showing the various flag counts in the training dataset (Jupyter).



```
    print('===================== SGDC_ADA Classification report ==
    print()
    print(classification_report(Y_test,SGDC_ADA))
    print()
```

```
===================== SGDC_ADA Classification report =============

              precision    recall  f1-score   support

     anomaly       0.98      0.95      0.96      3480
      normal       0.96      0.98      0.97      4078

    accuracy                           0.97      7558
   macro avg       0.97      0.96      0.97      7558
weighted avg       0.97      0.97      0.97      7558
```

Figure 5: Showing the test classification report for SGDC_ADA model



```
    print('===================== ADA_SGDC Classification report ==
    print()
    print(classification_report(Y_test,ADA_SGDC))
    print()
```

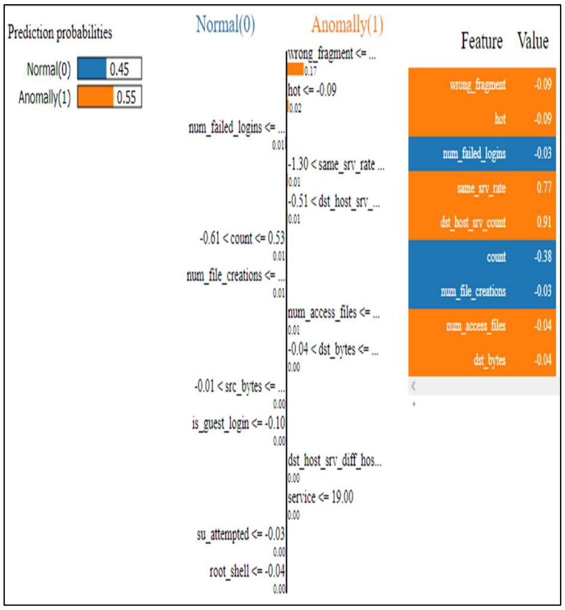
```
===================== ADA_SGDC Classification report =============

              precision    recall  f1-score   support

     anomaly       0.99      0.91      0.95      3480
      normal       0.93      0.99      0.96      4078

    accuracy                           0.96      7558
   macro avg       0.96      0.95      0.96      7558
weighted avg       0.96      0.96      0.96      7558
```

Figure 6: Showing the test classification report for ADA_SGDC model

From figure 5 and 6, it was observed that SGDC_ADA model which was the model gotten from passing SDGC before AdaBoost classifier (SGDC_ADA) using stacking technique outperformed passing AdaBoost before SGDC (ADA_SGDC) with a classification accuracy of 0.97 (97%) compared to SGDC_ADA 0.96 (96%). For evaluation, SGDC_ADA was selected and the result was explained using LIME in figure 7.

Figure 7 is a display gotten from the testing result when a user's network usage details were passed into the developed model which was later explained using LIME. It was observed from the probability displayed that the user was browsing anomaly which means the user might be trying to intrude the network.

From the various results SGDC_ADA had an accuracy of 0.97 while ADA_SGDC had an accuracy of 0.96. Considering the classification report after testing the data, SGDC_ADA had an average precision of 0.97, average recall of 0.96, and average F1-score of 0.97 while ADA_SGDC scored an average precision of 0.96, average recall of 0.95, and average F1-score of 0.96.



Feature	Value
wrong_fragment	-0.09
hot	-0.09
num_failed_logins	-0.03
same_srv_rate	0.77
dst_host_srv_count	0.91
count	-0.38
num_file_creations	-0.03
num_access_files	-0.04
dst_bytes	-0.04

Figure 7: Shows the evaluation of the research with SGDC_ADA using LIME as explainer.

CONCLUSION

Based on the results mentioned above, it can be concluded that the SGDC_ADA classifier outperformed the ADA_SGDC classifier with an accuracy of 0.97 (97%). The SGDC_ADA model had the lowest rates of true negatives (TN) and false positives (FP) and the highest rates of true positives (TP) and false negatives (FN). Therefore, the SGDC_ADA model was found to be more effective for intrusion detection using the KDD CUP 99 dataset compared to the ADA_SGDC model. Detection using the KDD CUP 99 dataset.

REFERENCES

Chun Guo, Yuan Ping, Nian Liu and Shou-Shan Luo, (2022). A two- level hybrid approach for intrusion detection, Neurocomputing, http://dx.doi.org/10.1016/j.neucom.2016.06.021

Deepthy K. Denatious & Anita John (2012). “Survey on Data Mining Techniques to Enhance Intrusion Detection”, International Conference on Computer Communication and Informatics (ICCCI 2012), Jan. 10 – 12, 2012, Coimbatore, INDIA.

Gupta M. and Shrivastava S. K. (2015). "Intrusion detection system based on SVM and bee colony", Int. J. Comput. Appl. 1 1 1, pp. 27-32.

Jones, A. K., & Sielken, R. S. (2000). Computer system intrusion detection: A survey. “Kdd-cup-99 task description.” [Online; accessed 2022-03-18].

Mazraeh S., Ghanavati M. and Neysi S. H. N. (2016). "Intrusion detection system with decision tree and combine method algorithm", Int. Acad. J. Sci. Eng, no. 3, pp. 21-31.

McCumber, J. (1991). Information systems security: A comprehensive model. In *Proceedings 14th National Computer Security Conference,* pp. 328-337.

Menahem, E., Rokach, L., & Elovici, Y. (2009). Troika–an improved stacking schema for classification tasks. *Information Sciences*, *179*(24), 4097-4122.

Rupali M. and Brajesh K. U (2014). “Comparison of NBTree and VFI Machine Learning Algorithms for Network Intrusion Detection using Feature Selection”, *International Journal of Computer Applciations* (0975-8887) Vol 108- No. 2, December, 2014.

Siddique K. (2019). KDD Cup 99 Data Sets: A Perspective on the Role of Data Sets in Network Intrusion Detection Research. ieeeexplore.ieee.org

Singh R, Kumar H. and Singla R. K. (2015). "An intrusion detection system using network traffic profiling and online sequential extreme learning machine", Expert Syst. Appl. 42, pp. 8609-8624.

Wei Hu, & Weiming Hu. (2005). Network-Based Intrusion Detection Using Adaboost Algorithm. The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI’05). doi:10.1109/wi.2005.107