International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 10 Issue 6 ǁ 2022 ǁ PP. 1630-1635

Intrusion Detection of Imbalanced Network Traffic Based On Ensemble Learning

SOFIYA MATHEW
TSERING NAMGAYAL
ZEHNEENA P N
(Department of Computer Science and Engineering Holy Grace Academy of Engineering Mala, Thrissur)

ABSTRACT
In unequal network traffic, vicious internet attacks often hide in large numbers normal data. It shows a high level of concealment and blurring on the Internet, making it difficult for the Network Intrusion Detection System (NIDS) to ensure accuracy and timing of detection.
In this project, the main objective of it is to develop a cyber security intrusion detection system using machine learning (deep learning) and with help of ensemble language to assemble different classifier methods to obtain a better predictive performance than could be get from implementing any one method alone. We use long short term memory classifier, Alexnet classifier and mini-VGGnet classifier for getting better predictive result
Here a machine learning technique called Ensemble Learning, which is a general meta approach that seeks better prediction by combining predictions from multiple models. Ensemble Learning voting classifier is created by using created classifiers and then predicting data using ensemble learning.

Date of Submission: 12-06-2022 Date of acceptance: 26-06-2022

I. INTRODUCTION

The number of dangers to users' information has increased as the Internet has expanded. The threats vary from minor annoyances to life-threatening situations. Furthermore, in recent years, the assault tools and methodologies have become significantly more complex. The Network Intrusion Detection System (NIDS), as the protection mechanism behind the firewall, must reliably identify malicious network attacks, provide real-time monitoring and dynamic protective measures, and make strategic decisions. The intrusion detection under huge pressure due to hiding of malicious or infected data hiding among large number of normal data and even machine learning algorithm comes under constrains and makes mistakes by allowing infected data through defence line to PC.

Since Lecun et al introduce the theory of Deep Learning as an important and vital subfield of machine learning, deep learning has shown excellent performance and played big role. Deep Learning is a subfield of artificial intelligence that is built on the discipline of machine learning. Deep learning will suffice since neural networks imitate human brain. Nothing is explicitly programmed in deep learning. Essentially, it is a machine learning class that does feature extraction and transformation using a significant number of nonlinear processing units. Each successive layer accepts the results from the previous layer as input.

1.1 Pre-processing

Real-world data frequently contains noise, missing values, and is in an unsatisfactory format that cannot be used directly in machine learning models. Data pre-processing is performed first in our intrusion detection structure in the proposed intrusion detection model. Data pre-processing is a major prerequisite for cleaning data and making it fit for a machine learning model, which enhances the model's effectiveness and precision. We remove null rows and columns and remove duplicative values.

Then we separate data into training set (subset to train the model) and testing set (a subset to test the train model). The majority of the data is used for training, while a smaller fraction is used for testing.

1.2 Data Balancing

Malicious cyber-attacks can lurk in vast quantities of normal data in unbalanced network traffic. In cyberspace, it exhibits high level of stealth and obfuscation, finding it challenging for network intrusion detection systems (NIDS) to guarantee detection accuracy and timeliness. first, divide the imbalanced training set into two groups: near-neighbour and far-neighbour. Because the samples in the near-neighbour set are so

similar, it's difficult for the classifier to identify the differences between the categories, we call them difficult samples and the samples in the far-neighbour set easy samples.

In the ENN algorithm, the K neighbours are used as the scaling factor for the entire algorithm. The number of challenging samples increases as scaling factor K increases, as does the compression rate of the majority of samples and the synthesis rate of the minority of class. A data point is categorized using the majority of the "K" nearest training data points in the K-Nearest Neighbours classifier. "K" is a positive number that must be determined ahead of time. When K is kept low, the model may include noisy data and play badly. The model does not reflect the general characteristics of the training data if K is too large. KNN is a variety of instance-based learning algorithm. Create a new training set from difficult and easy set.

1.3 Classifiers

Classifiers are machine learning algorithms that are trained using the extracted features outlined in feature extractors. They are an algorithm that sorts or classifying data into one or more "classes" automatically. Classes are described using terminology such as targets, labels, and categories.

We used three common supervised classification methods, that are long shot term memory classifier, Alexnet classifier and Mini-VGGnet classifier to predict whether or not a given data set contain attract or not.

1.3.1 Long Short Term Memory classifier

As you learn more about data-driven predictions, you'll come across the phrase LSTM. LSTM is the most effective method for overcoming forecasting difficulties. Hochreiter, a former Schmidhuber Ph.D. student, first closely researched these issues on Schmidhuber's RNN long time lag project in year of 1991.

A feedback network known as "Long Short-Term Memory" addresses the primary drawbacks of typical RNNs and It discussed the problem of RNN long-term dependency, in which the RNN seems unable to predict words stored in long-term memory but can produce more accurate predictions utilizing recent data.

The LSTM features a chain structure with four neural networks and various memory blocks known as cells. It has three gates: input gates, output gates, and forget gates, which control the flow of data into and out of the cell and allow the cell to remember values across arbitrary time intervals.

1.3.2 Alex net classifier

AlexNet is one of the classic basic networks of deep learning. AlexNet is a convolutional neural network (CNN) architecture created by Alex Krizhevsky in collaboration with Ilya Sutskever and Geoffrey Hinton, Krizhevsky's Ph.D. advisor.

On September 30, 2012, AlexNet actively participated in the ImageNet Large Scale Visual Recognition Competition, it achieved error of 15.3 percent, which was more than 10.8 percentage points lower than the runner- ups.

It contained eight layers; the first five were convolutional layers, and then followed by max-pooling layers, and the last three remaining layers were fully connected layers.

1.3.3 Mini-VGGnet classifier

Researchers from Oxford University's Visual Geometry Group and Google DeepMind collaborated in 2014 to produce a deep convolutional neural network with new features: VGGNet.

In the ILSVRC2014 classification project competetion came in 1st runner up of ILSVR2014 in the classification task while the winner is GoogLeNet.

VGGNet is distinctive in that it employs three kernels throughout its architecture. VGGNet's ability to generalise to classification issues outside of where it was trained is arguably due to the usage of these tiny kernels.

If you discover a network architecture that is solely made up of 33 filters, you may be sure it was influenced by VGGNet. This introduction to Convolutional Neural Networks is too advanced to go over the whole 16 and 19 layer variations of VGGNet.

SYSTEM DESIGN

To begin, divide the unbalanced training set into two groups: close neighbours and far neighbours. Because it's difficult for the classifier to distinguish between the categories because the samples in the near-neighbour set are so similar, we name them difficult samples and the samples in the far-neighbour set easy samples. The K neighbours are utilised as the scaling factor for the entire method in the ENN algorithm.

In the K-Nearby Neighbours classifier, a data point is classified using the majority of the "K" nearest training data points. "K" is a positive value that must be calculated beforehand. When K is set to a low value, the model is more likely to include noisy input and perform poorly. If K is greater than 1, the model does not reflect the general properties of the training data. Then we create new training data with easy set by setting 80% data used for training while reaming 20% data for testing.

CLASSIFIERS

With spited the dataset into training (80%) and test (20%) subsets we trained three classifiers namely LSTM, AlexNet and VGGNet.

Create LSTM Classifier model

- Training the model using training set
- Saving the LSTM model

Create AlexNet Classifier model

- Training the model using training set
- Saving the AlexNet model

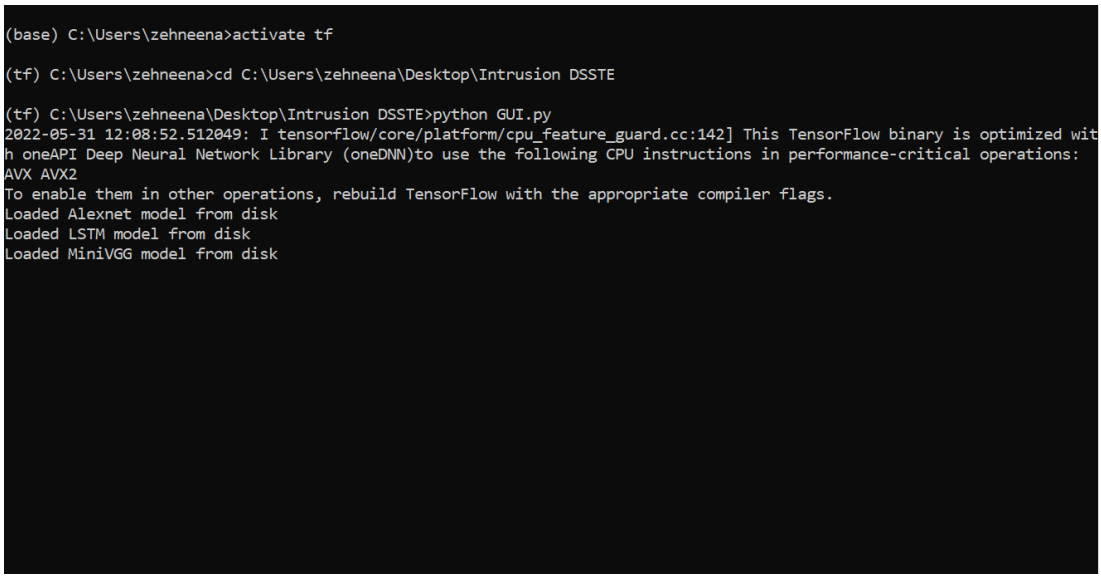
Create Mini-VGGNet Classifier model

- Training the model using training set
- Training the model using training set

ENSEMBLE MODEL

Loading Saved LSTM, AlexNet, Mini-VGGNet Model in system load the data and predict data using each loaded model to obtain the result. Prediction result of three model should be stored into list.

Create Ensemble learning Voting classifier by using created classifiers and saving the Ensemble model. Find the majority prediction result using assembly language approaches by combine many three algorithms to achieve higher predictive performance than any of the individual model could.



Hardware Specification

- Processor: i5 or i7 (i5 is better)
- RAM: 12GB (Minimum)
- Hard Disk: 500GB or above
- Mouse
- Keyboard

Software Specification

- Tool: Python IDLE
- Python: version3
- Operating System: Windows 7 or later
- Front End: Python

GUI (PYTHON TKINTER)

A Page where users can give input. The result predicted will be shown on the very same page. The Python interface to the Tk graphic user interface toolkit that accompanies with Python is called Tkinter. It is also a Python's standard graphic user interface library. When Python is used in combination with Tkinter,

developing graphical user interfaces is quick and also very simple. The Tk graphic user interface toolkit has a comprehensive object-oriented interface called Tkinter.

II. RESULT

This section, we summarize our test results. We used Accuracy, Prediction, Recall and F1-score as performance measures which are derived from the values in the confusion matrix. Confusion matrix is defined in Table (a).

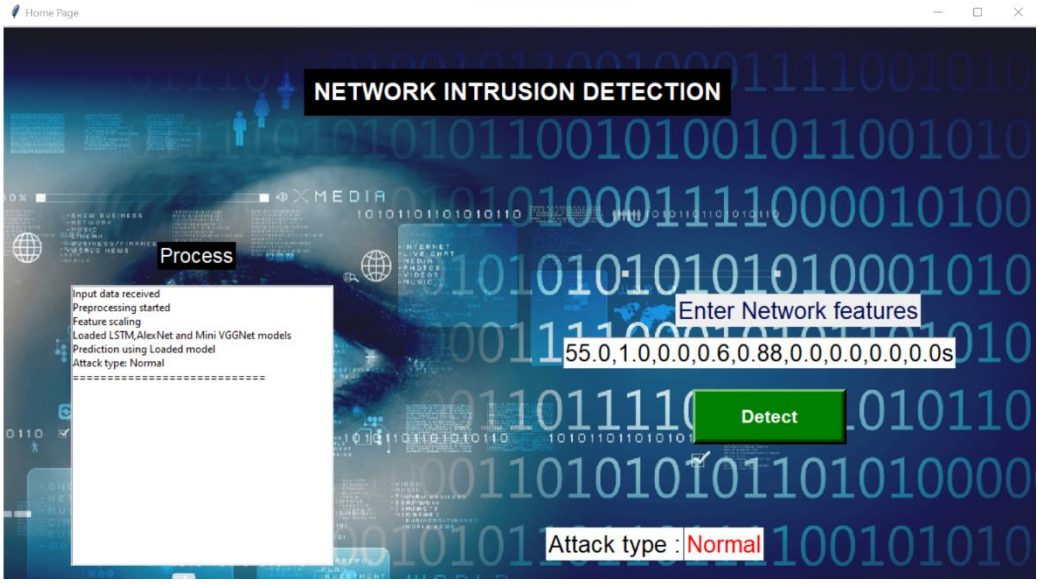
Positive (1)	Negative (0)	
TP	FP	Positive (1)
FN	TN	Negative (0)

These evaluation criteria indicate the flow recognition accuracy rate and false alert rate of the intrusion detection system.

Accuracy = TP+TN / TP+TN+FP+FN
Precision= TP / TP+FP
Recall= TP+TN / TP+TN+FP+FN
F1_Score=2*precision*recall / precision + recall
Where
TP= true positive
FP= false positive
FN=false negative
TN= true negative

CLASSIFIERS

Three classifiers (LSTM, mini VGGNet and AlexNet) to classification where LSTM achieved the highest accuracy rate of 78.72% and the highest recall rate of 75.82% and miniVGGNet achieves the highest accuracy of 96.99% and the highest recall of 97.04%. finally, AleNet achieved the highest accuracy rate of 82.84% and the highest recall rate of 81.66%. Finding the majority prediction result using assembly language approaches by combine many three algorithms to achieve higher predictive performance than any of the individual model could.



III. CONCLUSION

With the rapid development of internet and many activities done online with sharing and storing lots of private vital information online lead number of cases of intrusion detection. We user as well need to improve our security of data and prevent the attackers from accessing user data. We need to develop vastly and properly in intrusion detection field

In this project we studied intrusion detection system build with three classifiers namely LSTM, mini VGGNet and AlexNet classifier and find their flow recognition accuracy rate and false alert.

In this experiment we used ensemble model and found that the majority prediction result using assembly language approaches by combine three classifiers to achieve higher predictive performance than individual based intrusion detection system performance.

As for future works, more features like video, audio and larger file can be used as input and we can build intrusion prevention and elimination features after detecting them.

Features like shield from intrusion can be developed and it can be inbuild in g-mails, WhatsApp and Facebook.

REFERENCE

[1]. M. Zamani and M. Movahedi, ‘‘Machine learning techniques for intru- sion detection,’’ 2013, arXiv:1312.2177. [Online]. Available: http://arxiv. org/abs/1312.2177

[2]. M. Panda and M. R. Patra, ‘‘Network intrusion detection using Naive Bayes,’’ Int. J. Comput. Sci. Netw. Secur., vol. 7, no. 12, pp. 258–263, 2007.

[3]. N. Japkowicz, ‘‘The class imbalance problem: Significance and strategies,’’ in Proc. Int. Conf. Artif. Intell., vol. 56, 2000, pp. 111–117.

[4]. T. Young, D. Hazarika, S. Poria, and E. Cambria, ‘‘Recent trends in deep learning based natural language processing [review article],’’ IEEE Comput. Intell. Mag., vol. 13, no. 3, pp. 55–75, Aug. 2018.

[5]. N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, ‘‘A deep learning approach to network intrusion detection,’’ IEEE Trans. Emerg. Topics Comput. Intell., vol. 2, no. 1, pp. 41–50, Feb. 2018.

[6]. W. Zhong, N. Yu and C. Ai, "Applying big data based deep learning system to intrusion detection," in Big Data Mining and Analytics, vol. 3, no. 3, pp. 181-195, Sept. 2020, doi:10.26599/BDMA.2020.9020003.

[7]. B.A.Tama,M.Comuzzi,andK.-H.Rhee,‘‘TSE-IDS:Atwo-stageclas- sifier ensemble for intelligent anomaly-based intrusion detection system,’’ IEEE Access, vol. 7, pp. 94497–94507, 2019.

[8]. P. Jeatrakul, K. W. Wong, and C. C. Fung, ‘‘Classification of imbalanced data by combining the complementary neural network and smote algorithm,’’ in Proc. Int. Conf. Neural Inf. Process. Springer, 2010, pp. 152–159.

[9]. B. Yan and G. Han, ‘‘LA-GRU: Building combined intrusion detection model based on imbalanced learning and gated recurrent unit neural net- work,’’ Secur. Commun. Netw., vol. 2018, pp. 1–13, Aug. 2018.

[10]. D. E Denning, ‘‘An intrusion-detection model,’’ IEEE Trans. Softw. Eng., vol. SE-13, no. 2, pp. 222–232, Feb. 1987.

[11]. N.B. Amor, S. Benferhat, and Z. Elouedi, ‘‘Naïve Bayes vs decision trees in intrusion detection systems,’’ in Proc. ACM Symp. Appl. Comput. (SAC), 2004, pp. 420–424.

[12]. S. Hochreiter and J. Schmidhuber, ‘‘Long short-term memory,’’ Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[13]. T. Chenand C. Guestrin, ‘‘XGBoost: A scalable tree boosting system,’’ in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016, pp. 785–794.

[14]. Generative Oversampling for Mining Imbalanced Datasets Alexander Liu, Joydeep Ghosh Member, IEEE, and Cheryl Martin Member, IEEE
A Detailed Analysis of the KDD CUP 99 Data Set
Tavallaee, Mahbod; Bagheri, Ebrahim; Lu, Wei; Ghorbani, Ali-A
A Detailed Analysis of the KDD CUP 99 Data Set
Tavallaee, Mahbod; Bagheri, Ebrahim; Lu, Wei; Ghorbani, Ali-A
A Detailed Analysis of the KDD CUP 99 Data Set
Tavallaee, Mahbod; Bagheri, Ebrahim; Lu, Wei; Ghorbani, Ali-A

[15]. M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009, pp. 1-6, doi: 10.1109/CISDA.2009.5356528.