Intrusion Detection System Using Machine Learning with C/C++, Rust, python

Jatin Aggarwal, TU-Ilmenau, IA, RCSE

Problem statement-The task is to find the best possible tools and language for a network anomaly attack, a predictive model capable attacks.

***Abstract*—In today's world, the increasing complication and severity of security attacks on computer networks, to protect the organizations' data and reputation, security researchers to incorporate different machine learning methods. Recently are vastly anomaly attacks and intrusion detection systems increase their performance in securing the computer networks and hosts by the deep learning and machine learning techniques. This article focuses on the deep learning-based intrusion detection schemes and puts forward languages and tools best in use today. First, we introduce the article and the primary background concepts about anomaly attacks, IDS architecture and the type of deep learning methods available. Then describes how data set collected through and different platforms and features selection. Finally, a complete analysis of the investigated IDS frameworks language and tools best used in organizations and concluding remarks and future.**

Keywords- machine learning, deep learning, wireshark, rust, C/C++, libpcap, Intrution detection system IDS, LSTM, ANN

I. Introduction

As we know, for the various type of security attacks, the IDS probably are one of the main cyber security components. With the combination of firewalls it can effectively handle the anomaly attacks. Machine learning help us to realized the different anomaly classification. By relying on the users' normal behavior profiles, this approaches can detect new attacks [1]. However, in dynamic establishments in which end users' roles change periodically, user profiles should be updated functionally. Many of recent investigators are conducted in both anomaly detection and misuse detection contexts using various deep learning techniques [2]. Conventional machine learning techniques suffer from the lack of efficiency of tools which makes it difficult for deployment on large platforms[3].The third part discussed the framework/model for running or adopting IDSs. We discussing the challenges, motivations, recommendations and substantial analysis of the related work. The result that mapped new directions and a discussion on the efficiency of hybrid techniques were presented to identify the gap in future directions that focused on DL techniques, which achieved better accuracy rate than the other techniques.

II. Background Study

[4]Yu et al[5] provided a session-based network IDS using a deep learning-based scheme and achieved performance in recognizing botnet traffics. IDS performance is not discussed.The IDS in [6] applied real-time data as input to a neural network. They used a deep multi-layer perceptron and also an RNN model, which benefits from an LSTM hidden layer for learning the temporal context of several attacks such as command injection and DDoS. Yang et al.[7], proposed ICVAE-DNN, an IDS model that combines an improved conditional variational auto- encoder with a DNN. In this scheme, the auto-encoder is used to learn and explore sparse representations between network data features and classes. Though there are many surveys and publications those talk about machine learning for IDS but there are very limited publications for the platforms that suitable for the same. The above studies shows that, the performance of RNN, ANN and DNN for IDS useful in many cases.

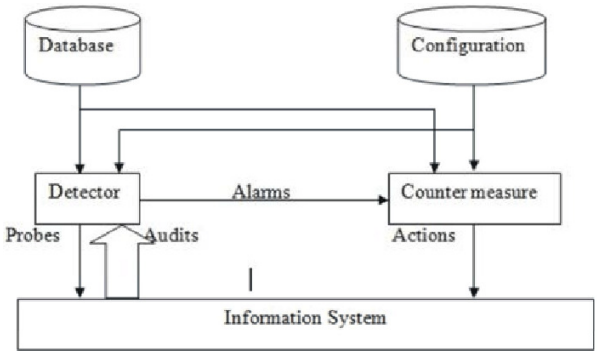

Fig. 1. IDS Architecture [20]

A. Machine learning

The machine learning performance is compared to a simple baseline model. In this study, we considered the linear and regularized linear models, k-Nearest Neighbors, Random Forest, Support Vector Machine, and Multi-Layer Perceptron Neural Networks. For classification, consider the Naive Bayes model. For models implementation, we can use the sklearn library.

B. Deep learning

For various signal analysis research projects deep learning is one of the hot topics. In deep learning, YOLO is the most widely used tool. It has a darknet framework based on an Artificial neural network (ANN). BBOXtool is also a very useful tool for classification and labeling. Multilayer perceptron (MLP)[8] is a class of feed-forward ANN. The architecture consists of three different layers. The Innermost layer (input layer) received the input features x, an arbitrary number of hidden layers where the computation is done, and the Outermost layer (output layer) which outputs the prediction. Every layer is formed by one or multiple operations, e.g.

neurons that compute a linear combination of the outputs from the previous layer. The non-linearity is incorporated into the model by passing the output through an activation function. In this study, we see the RELU activation function at the end of each hidden layer.

III. Data set

A. Wireshark

Wireshark is the commonly known foremost and widely-used network protocol analyzer. Wireshark let you see what's happen on your network at a very deep level and is the de facto standard across many commercial and non-profit organisations, government companies, and institutions. Wireshark development flourish, thanks to the volunteer contributions of networking people around the world. Wireshark, formerly known as the Ethereal network analytical tool, captures and displays packets in real time in experts readable format. This tool comprises filters, colour encoding and other characteristics that allow experts to check individual packets and deep down into network traffic. Libpcap or Winpcap libraries to capture network traffic in C. Winpcap libraries are not intended to work with wireless network cards and do not support traffic capture on Windows using Wireshark. Therefore, the Wireshark monitor mode is not recommended by default for Windows.Data can capture from this tool. Which is in .Pcap file format after we can convert that into .CSV file format with respect to the language we are using. The bigger picture of this tool can be read through references.

In data collection phase, which involves set-up and data capture. The test-bed set-up for the intrusion data set requires, a router as the major components for the start. The victim uses Wireshark to monitor and capture packets. The collection of Winpcap is mostly restricted, whereas that of Wireshark is not. However, [9]wireshark supports Airpcap, a unique and expensive set of network Wi-Fi adapters that drive support monitor network traffic. That is, after the packets are captured in Wi-Fi network traffic, the software can export them into a pcap file type.

IV. Languages

A. Python

As we know, nowadays the one of the most popular language for machine learning in universities is python. Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including meta programming and meta objects). Many other paradigms are supported via extensions, including design by contract and logic programming.[8] Which is easier to use due to its user friendly IDS and platforms. But in this case effectiveness of the system is important rather a user friendly language. The popular libraries in the python are - pandans, Sk-learrn, tensorflow, matplotlib, openCV, etc. Google colab and jupiter notebook are the some IDE used for python.

B. C/C++

Run inference on your machine learning models with framework you train it with. The best part of C is that it is efficient. Using C in Linux is the one of the best choice if we have to deal with machine learning models. Thre are some libraries for Linux operating system libpcap, shark library, tensorflow, thundersvm, darknet, etc. One of the finding is that we cannot use the pcap files in C with windows operating system efficiently. This is because the supporting libraries are not updated with the latest version of windows. For example, winpcap is a library to read pcap file that is not supporting and updating in latest version of window.

C. Rust

Due to its speed, rust is a great language for scientific computation, expressiveness and memory safety but the lack of good machine learning libraries makes it harder to use this language for statistical modeling. Smart core was created to help establish Rust as a leading language for data science and machine learning.[10] Also in our experience we found out that the community for rust users are limited. People with higher experience can use rust for machine learning.community, and the right tools don't exist, yet. As said[11] Machine Learning Framework is essentially a layer on top of very performing data management, computation and mathematical libraries. It is unfortunate that there are no solid building blocks for those either, which actually leads these fields and other fields of scientific computation to the end. Hope, that the Rust community attracts more people(right now is less) from different scientific fields to create the tools all parties of scientific computation will need. The current Machine Learning field, there are around 15-20 Rust Machine Learning libraries on Github. None of them, would be safe to consider them for serious applications. Experimentation would be try to get the current state of Rust's community.

V. Conclusion

In this case study mainly focused on types of languages and platforms used for anomaly attack detection system and different frameworks. The efficiency of the tools and communities can make a difference creating the good system. This is why it is clearly picture out that, the C language could give us a best results. as we have good community and mature organisations to protect the solid building block for performance and management of C libraries. The experimentation and development is ongoing without any blockage of concerns. The pipeline with c and machine learning is also better in many cases. compare to Rust, C have better community. We also observe that there are vast market for embedded system where the C is more efficient to connect the pipelines of machine learning models. Python on the other hand, is definitely easier to use and it community support it dynamically high. But it has no speed compare to C. Also python need more time and difficulty when we need to connect the model with embedded system. Many systems did not work or comparative to python models. for that we need to develop bulky header files and

dependencies which again make system inefficient. if we have fully developed model in C with Linux environment then it can be integrated with other system easily.

VI. Future work

After implementation of this work into an application, this application can be used for different purposes. Many researchers are still working for the perfection because some of this type of work are not performing in real time perfectly. There are many difficulties in the implementation but we need to find an efficient way to solve this. Considering these cons we still have a lot to work on.

VII. References

[1] N. Pandeeswari and G. Kumar, ""anomaly detection system in cloud environment using fuzzy clustering based ann,"," *Mobile Netw. Appl*, Jun. 2016.

[2] A. A. Diro and N. Chilamkurti, "'distributed attack detection scheme using deep learning approach for internet of things,"," *Future Gener. Comput. Syst.*, may 2018.

[3] A. Z. A. Aleesa, B. Zaidan and N. M. Sahar, "'review of intrusion detection systems based on deep learning techniques: Coherent taxonomy, challenges, motivations, recommendations, substantial analysis and future directions,"," *Neural Comput. Appl...*, jul,2020.

[4] M. M. M. K. M. S. H. T. K. S. R. M. H. JAN LANSKY, SAQIB ALI and A. M. RAHMANI, "Deep learning-based intrusion detection systems: A systematic review," in *IEEE Digital Object Identifier 10.1109/ACCESS.2021.3097247*, July 14, 2021.

[5] J. L. Y. Yu and Z. Cai, "'session-based network intrusion detection using a deep learning architecture,"," *Modeling Decisions for Artificial Intelligence. Cham, Switzerland: Springer.*, Springer.2017.

[6] R. H. G. S. Y. Y. G. Loukas, T. Vuong and D. Gan, "'cloud-based cyber-physical intrusion detection for vehicles using deep learning"," *IEEE Access,..*, 2018.

[7] C. W. Y. Yang, K. Zheng and Y. Yang, "'mproving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network"," *Sensors,.*, june,2019.

[8] *https://en.wikipedia.org/wiki/Python(programminglanguage)]W hichiseasiertouseduetoitsuserf riendlyIDSandplatf orms.Butinthi.*

[9] W. E. Lamping U, Sharpe R, "Wireshark user's guide: for wireshark," 2014.

[10] *https://medium.com/swlh/machine-learning-in-rust-smartcore-2f472d1ce83,.*

[11] *https://medium.com/@autumn$_e$ng/about − rust − s − machine − learning − community − 4cda5ec8a790.hvkp56j3f*].

[12] *https://github.com/josephmisiti/awesome-machine-learningc.*

[13] *https://www.wireshark.org/docs/.*

[14] M. L. Dongmin Wu, Yi Deng, "Federated learning for anomaly detection using mixed gaussian variational self-encoding network," *Information Processing Management, vol.59, no.2, pp.102839,*, 2022,.

[15] *https://github.com/bkimminich/it-security-lecture/blob/master/slides/01-04-network$_s$ecurity.md.*

[16] *http://www.packetech.com/showthread.php?2039-How-to-capture-WiFi-traffic-using-Wireshark-on-Windows.*

[17] *https://networkengineering.stackexchange.com/questions/24265/capturing-and-sending-802-11-frames-packets.*

[20]Bharathy, A M Viswa. (2017). A Hybrid Intrusion Detection System Cascading Support Vector Machine and Fuzzy Logic. World Applied Sciences. 35. 104-109. [12][13][14][15][16][17][**?**]