Kitsune Dataset Analysis via BigData and Deep Learning Techniques

Igor Zelichenok
St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS)
Saint-Petersburg, Russia
zelichenok@comsec.spb.ru

Igor Kotenko
St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS)
Saint-Petersburg, Russia
ivkote@ comsec.spb.ru

2024 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT) | 979-8-3503-6289-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/USBEREIT61901.2024.10584030

***Abstract*—During the comprehensive digitalization of information processes, the amount of network traffic is growing rapidly, which leads to an increase in the frequency and complexity of network threats. Thus, developing effective network intrusion detection systems (NIDSs) that can detect complex multi-step attacks in a timely and accurate manner is one of the important challenges. In this context, BigData and Machine Learning technologies can greatly improve the accuracy and timeliness of complex attack detection. The paper presents NIDS, which consists of two machine learning models with LSTM layers for analyzing long and short sequences and is capable of processing large amounts of data through the use of big data processing techniques. The main goal of this study is to demonstrate the proposed architecture, as well as validate its characteristics by testing the Short-Term information module on an existing data set. The used machine learning models were trained on the Kitsune dataset on a binary classification task and performed in 93% accuracy and 0.03 losses respectively.**

***Keywords*—network intrusion detection, deep learning, big data, LSTM**

I. Introduction

Every year, the number of devices connected to the global network is growing many times over. The network becomes more complex, the volume of traffic increases, and there are more devices. Due to the spread of the Internet, the quantity of attackers is also growing, carrying out increasingly complex attacks every year. Many globally significant enterprises and objects connected to the network can become the target of attackers. These are banking systems, devices of high-ranking or famous persons, critical infrastructure facilities.

One of the types of significant threats are complex multi-step attacks carried out quietly, in several stages, on many target devices at once. These properties make multi-stage attacks difficult to detect using traditional threat detection methods.

In this regard, there is a growing need for an efficient protection system against such network attacks, capable of automatically analyzing large volumes of information, drawing timely conclusions, and predicting future paths of threat spread.

Thanks to big data processing and machine learning (ML) technologies, it has become possible to identify potential threats among a large flow of network data both over short and long time periods.

However, for ML models to work effectively, they require complete and balanced datasets, otherwise the model will be undertrained or overtrained. To train the models, it was customary to use the Kitsune dataset [1], which was collected on a bench that maximally simulates a real subnetwork with a wide variety of devices included in it.

This paper presents a new NIDS framework, which combines big data processing and ML technologies. It consists of four main components: the client part is responsible for collecting, filtering and sending data to the server, the sequence processing module, which implements the main part of preprocessing and load balancing, the part with deep learning, consisting of two LSTM (Long Short-Term Memory) models, one of which detects a current attack, and the second – over a long period of time, as well as a module for working with big data, consisting of a database cluster and a web service for implementing a dashboard. We experiment with this framework using the Kitsune dataset.

The work is structured as follows. Section 2 provides an analysis of relevant works. Section 3 discusses the architecture of the framework. Section 4 presents the description of the dataset and the results of experiments. The paper concludes with discussion on the results and definition of directions for further research.

II. Relevant papers

Many works based on the analysis of the Kitsune dataset are devoted to the analysis of only one of the 9 presented classes, namely the Mirai botnet, and therefore there are only few articles in the research space that present the implementation of a multi-class model.

The article [2] focuses on developing a methodology for detecting Mirai botnet attacks in the traffic of small-scale networks simulated in Kitsune. The researchers proposed a system using ML methods, including Artificial Neural Networks (ANN), Support Vector Machine (SVM), and K-Nearest Neighbors (K-NN), to analyze and classify network traffic for malicious activity detection. Their approach's uniqueness lies in using bootstrapping techniques to generate samples of varying sizes and comparing the models' performance on these samples. This enabled determining the optimal sample size for training the model with high accuracy and minimal training time, particularly relevant for devices with limited resources. The results showed that a model with a sample size of 10,000 achieves a detection accuracy of 99.56% with minimal training time.

In [3] an evaluation of various ML models for the Kitsune intrusion detection system was conducted with the goal of identifying Mirai botnet attacks. The research revealed that tree algorithm variants, including Simple Tree, Medium Tree, Coarse Tree, RUSBoosted, and Bagged Tree, show similar

This work was supported by the Russian Science Foundation Grant No. RSF 21-71-20078.

Authorized licensed use limited to: Odisha University of Technology and Research. Downloaded on March 11,2025 at 06:22:43 UTC from IEEE Xplore. Restrictions apply.

efficacy but with minor differences in performance. Notably, the Coarse Tree was identified as the most suitable algorithm for detecting Mirai botnet attacks due to its high prediction speed and accuracy, along with relatively low classification error costs and training time. Their accuracy was identical and reached 98-99%, however, the authors presented only binary classification for individual sub-datasets.

The main difference between the two studies [3, 4] lies in their focus on ML models for intrusion detection in the Kitsune system against Mirai botnet attacks. In [3] the authors evaluate the effectiveness of various tree algorithm variants, identifying the Coarse Tree as the most effective for detecting Mirai botnet attacks. In contrast, [4] expands the evaluation to include different datasets within Kitsune, concluding that the Fine Tree algorithm is most suitable for an enhanced version of Kitsune, given its performance across various datasets.

In [5] the application of ML algorithms for predicting ARP spoofing is demonstrated. Researchers used Long Short-Term Memory networks and decision tree classifiers to predict ARP spoofing, achieving accuracies of 99% and 100% respectively. The results indicate that both methods effectively predict intrusions, with the decision tree outperforming LSTM in terms of execution speed.

The study [6] compares three autoencoder-based ML models for anomaly detection and evaluates their resilience to data poisoning attacks. The models include a simple autoencoder, a Deep Autoencoder (DAE), and an ensemble of autoencoders (KitNET). In tests with unpoisoned traffic, all models showed similar effectiveness, with an F1 score around 97% at 1% FPR. However, during data poisoning attacks, DAE proved to be the most resilient, maintaining over 50% F1 score at 10% poisoning, while other models showed significant performance degradation at just 0.5% poisoning.

Table I presents summary characteristics of relevant works, as well as characteristics of the proposed approach.

TABLE I. RELATED WORKS CHARACTERISTICS

Work	Characteristics			
	Clas-ses	***BigData technique***	***Chain analysis***	***ML/DL technique***
Güven et al. [2]	2	Bootstrapping	-	ANN, SVM, K-NN.
Alabdulatif et al. [3]	2	-	-	Tree algorithms
Alabdulatif et al. [4]	4	-	-	Tree algorithms
Usmani et al. [5]	2	-	-	LSTM tree-based
Bovenzi et al. [6]	2	-	-	Autoencoders
The proposed framework	2	DB cluster, parallel preprocessing, client-server app	+	ANN based on LSTM layers

III. PROPOSED FRAMEWORK

A. Architecture

This section presents the proposed architecture of a network intrusion detection system (NIDS) (Fig. 1), it consists of four main components: a client part, a stream processor, a component with ML models, and a part with big data processing.

It allows timely detection of complex attacks in large volumes of data based on the sequence of events, which is an important factor since most systems draw their conclusions based on a single event.

Client-side module, first, collects network logs from the client device, then filters unnecessary information. After that, packets are generated and sent to the server to the sequence processing module via an HTTP connection.

Then the server sends the packets to the Stack. The server checks the Stack at every iteration to make sure it is not full. If the Sequence handler does not have time to pick up the data in the Stack, an additional instance of the handler is created to free it. This load balancing mechanism allows one to process information in a timely manner and submit it to the ML module even under increased loads.

The sequence processor scans the Stack once every 30 seconds, and if new packets appear in it, it takes them, preprocesses them, and then forms a chain of events for subsequent analysis by short-term and long-term threat detection models. At the output, the Thread handler forms an event vector V from the received data, consisting of m events:

$$V = [\, m_1,\, m_2, \ldots,\, m_n] \,. \qquad (1)$$

ML side consists of three components: Training module, responsible for training the model, STIM (Short-term information module), which receives packets from the Thread handler, analyzes them for the presence of an ongoing attack, and sends data about it to the dynamic NoSQL Apache database Cassandra, and LTIM (Long-term information module), which interacts with the PostgreSQL relational database. It analyzes the processed events contained in PostgreSQL, classifies the attack category, and reconstructs the suspected attack scenario.

BigData side consists of three main components: a cluster of three databases, a Long-time alert formatter (LTAF) and visualization tools played by Grafana. The Apache Cassandra NoSQL database acts as a dynamic database that receives alerts from STIM, after which these alerts are collected by LTAF, discards unnecessary information, and loads this data into the PostgreSQL relational database, removing the parsed events from the NoSQL database.

This approach allows information about the system to be stored for a long time and allows the LSTIM module to reconstruct attack scenarios that potentially take place over long periods of time.

B. Model description

To detect multi-step attacks, it is critical to analyze chains of events [7] rather than focusing on single incidents. Multi-step attacks are often carried out in several stages, where each attacker's action may appear legitimate or may not raise suspicion upon superficial inspection.

However, by analyzing the sequence of actions as a whole, it is possible to identify hidden relationships and an overall attack strategy that would not be obvious when analyzing each event separately. Understanding the context and relationships between steps allows security systems to effectively identify and prevent complex attacks [8], providing a higher level of protection for information assets.

Authorized licensed use limited to: Odisha University of Technology and Research. Downloaded on March 11,2025 at 06:22:43 UTC from IEEE Xplore. Restrictions apply.

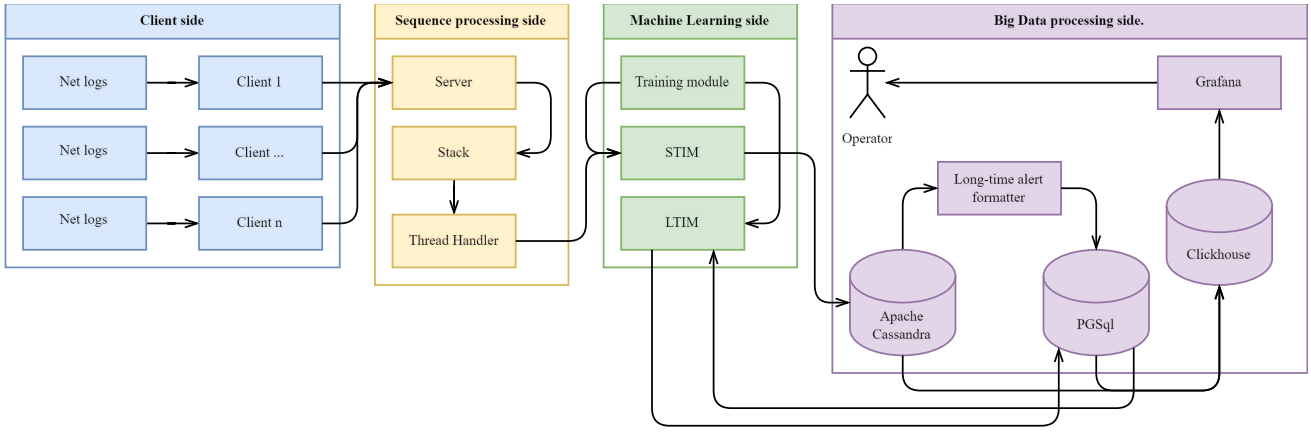

Fig. 1. Proposed NIDS architecture.

ML models with LSTM layers have significant advantages over other recurrent neural networks (RNN) in sequence analysis tasks due to their unique architecture and learning ability. One of the key advantages of LSTMs is their ability to overcome the vanishing gradient problem, which is a significant limitation for traditional RNNs. This is achieved using specialized gates that regulate the flow of information, allowing the network to retain information for long periods of time without significant loss of importance.

LSTM networks are also highly flexible in handling different types of sequences [9], whether fixed or variable length, making them ideal for a wide range of applications including natural language, time series, and more. Due to their ability to accurately model long-range dependencies within sequences, LSTMs outperform other RNNs in tasks that require context understanding and long-term memory. This allows them to learn more efficiently from complex data and achieve better results in tasks such as machine translation, speech recognition and time series prediction.

The proposed solution is based on two ML models, one of which is trained on short chains of sequences, works in binary classification mode and can detect an attack over a period of up to 30 seconds, while the second is designed to detect an attack scenario on data collected for seven days, after which it reconstructs the attack scenario.

In this study, we decided to focus on testing the binary classification model on the Kitsune dataset as one of the most comprehensive.

The model includes an LSTM layer with 16 neurons, two fully connected layers, one of which is an output layer with one neuron for binary classification, and an exclusion layer.

IV. Dataset and Experiments

A. Kitsune dataset description

The Kitsune dataset became available for use in 2019. This dataset contains a variety of traffic scenarios, including both normal network activity and various attacks such as Mirai Botnet, SYN DoS, Active Wiretap, ARP MitM, Fuzzing, OS Scan, SSDP Flood, SSL Renegotiation and Video Injection. Attacks are divided into 4 classes: Recon, Man in the Middle, Denial of Service, and Botnet Malware.

The dataset is divided into sub-datasets, each of which reflects a specific attack. The total size of the set is 64.18 Gb. Each dataset contains approximately 1 million rows, and the number of analyzed features is 115. Datasets are presented both as csv files of normalized logs and as raw pcap files.

This dataset was collected on a real commercial IP video surveillance system, which includes many IoT devices.

Due to the size of the dataset and limited resources, the preprocessing process of this size was a separate task. To solve this problem, it was decided to use an iterative learning process.

Due to insufficient data on the SYN DoS attack, and the unusual nature of the Mirai botnet, it was decided not to train for this type of attack, but they are planned to be included later.

Fig. 2 presents an algorithm for training models. Each sub-dataset was divided into blocks of 30,000 events, then each block was submitted for training. Then, to prevent the "forgetting" effect, when a new block of data arrived, lines from previous blocks were mixed into it, so the output was a trained model with LSTM layers, showing acceptable data on accuracy and losses in both training and test data.

B. Experiments

When creating the ML model, we used Python 3.10, the library for creating machine learning models Tensorflow, libraries for working with data such as Pandas and NumPy, and SciKit-Learn. The experiments were carried out on a home PC with the following configuration: 4.2 GHz, 32 GB RAM with an RTX 3060 graphics processor supporting machine learning calculations.

For training, sequences of 3 steps long events were taken in each dataset, which allows the model to draw conclusions based on sequences rather than one event as a whole.

When assessing the performance of deep learning models working in the proposed framework, classical metrics were used, such as: recall (2), accuracy (3), precision (4) and f-measure (5).

$$R_e = \frac{TP}{TP+FN} \quad , \tag{2}$$

$$accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad , \tag{3}$$

$$precision = \frac{TP}{TP+FP} \quad , \tag{4}$$

Authorized licensed use limited to: Odisha University of Technology and Research. Downloaded on March 11,2025 at 06:22:43 UTC from IEEE Xplore. Restrictions apply.

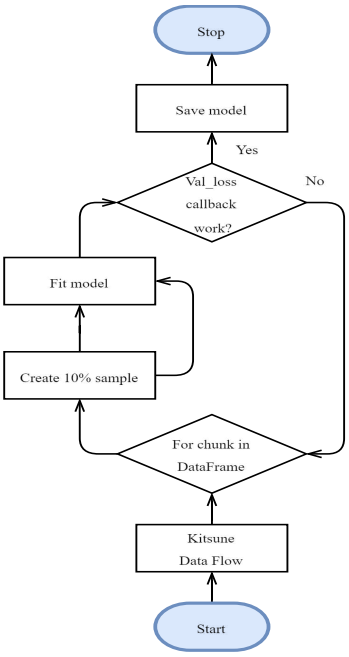

Fig. 2. Algorithm for training models.

$$f1 = 2 * \frac{precision*recall}{precision+recall} \quad . \qquad (5)$$

Recall shows the proportion of positive objects (TP) relative to all positive objects found by the model (TP + FN). Accuracy is the percentage of correct responses from a model relative to all its responses. Precision is calculated by the ratio of correctly predicted positives (TP) to the total number of true positives (TP) and false positives (FP) from the model. The F-measure is the average between Recall and Precision.

The binary classification model achieved accuracy and loss equal to 0.953 and 0.03 in training, and 0.93 and 0.03 in validation. Recall was equal to 0.9353, precision — 0.9383, and f1 was 0.9358.

V. Comparison and Discussion

Although the results obtained in this paper are not very high, relative to [2, 4, 5], they confirm the viability of the presented framework. In the paper the issues of optimizing training with limited resources and large volumes of training data were analyzed discussed. Besides, the differences of the suggested approach from other works were demonstrated, as well as experiments on the implemented framework were conducted.

The main difference between the analyzed NIDS and the one presented by the authors is the fact that the proposed system draws conclusions based on a sequence of events, and not on a single event.

Regarding the multiclass classification model, the work [10] should be mentioned. It is dedicated to the hybrid intrusion detection system COREM2, in which the authors used a ML model as a combination of 2D-CNN, a recurrent neural network with LSTM layers and MLP.

The suggested framework can be used both for binary and multi-class classification. In next our research will be related to conduct experiments for LTIM on all attack classes presented in the dataset.

For big data processing techniques, the authors presented a thread parallelization mode, where an instance of a model is created for each data block, and this leads to fast analysis, but with a high data flow this can lead to system overload.

VI. Conclusion

The paper presented a NIDS framework that implements methods for processing big data and machine learning. NIDS framework consists of the client part, the sequence processing module, the deep learning component, consisting of two LSTM models for detecting attacks for a short and a long period of time, and a big data processing module.

The features of training machine learning models in conditions of limited resources were shown, and the components of the framework were tested on the Kitsune dataset. The model showed an accuracy of 93% on test data.

Directions for future research include measuring the effectivity of the presented architecture, testing LTIM on multi-class classification problems on the Kitsune dataset, and demonstrating the operation of the sequence processing module.

References

[1] Y. Mirsky, T. Doitshman, Y. Elovici and A. Shabtai, “Kitsune: an ensemble of autoencoders for online network intrusion detection,” In Network and Distributed System Security Symposium, 2018.

[2] E. Y. Güven and Z. Gürkaş-Aydin, “Mirai botnet attack detection in low-scale network traffic,” Intelligent Automation & Soft Computing, vol. 37, no. 1, 2023.

[3] A. Alabdulatif, S. Rizvi and M. Hashmani, “Optimal machine learning models for kitsune to detect mirai botnet malware attack,” Journal of Hunan University Natural Sciences, vol. 48, no. 6, 2021.

[4] A. Alabdulatif and S. Rizvi, “Machine learning approach for improvement in kitsune NID,” Intelligent Automation & Soft Computing, vol. 32, no. 2, 2022.

[5] M. Usmani, M. Anwar, K. Farooq, G. Ahmed and S. Siddiqui, “Predicting ARP spoofing with machine learning,” in Proc. 2022 International Conference on Emerging Trends in Smart Technologies (ICETST), pp. 1-6, 2022.

[6] Bovenzi G. et al. “Data poisoning attacks against autoencoder-based anomaly detection models: a robustness analysis,” In Proceedings of ICC 2022-IEEE International Conference on Communications, pp. 427-5432, 2022.

[7] I. Kotenko, “Active Vulnerability Assessment of Computer Networks by Simulation of Complex Remote Attacks,” In Proceedings of 2003 International Conference on Computer Networks and Mobile Computing. ICCNMC-03. 2003. pp.40-47, 1243025.

[8] I. Kotenko and E. Doynikova, “The CAPEC based generator of attack scenarios for network security evaluation,” In Proceedings of the 2015 IEEE 8th International Conference on “Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications” (IDAACS'2015). 2015. pp.436-441, 7340774.

[9] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation. vol. 9, no. 8, pp. 1735-1780, 1997.

[10] A. Psathas, S. L. Lliadis, A. Papaleonidas and D. Bountas, “COREM2 project: a beginning to end approach for cyber intrusion detection,” Neural Computing and Applications, vol. 34, no. 22, pp. 19565-19584, 2022.

Authorized licensed use limited to: Odisha University of Technology and Research. Downloaded on March 11,2025 at 06:22:43 UTC from IEEE Xplore. Restrictions apply.