ANOMALY-BASED NETWORK INTRUSION DETECTION SYSTEM USING BIG DATA SYSTEM USING BIG DATA (ANODE)
DATABOSS

PROJECT TEAM

Beste Aydemir

Berfin Kavşut

Şevki Gavrem Kulkuloğlu

Ege Ozan Özyedek

Meltem Toprak

COMPANY MENTOR

Oğuzhan Karaahmetoğlu

ACADEMIC MENTOR

Prof. Süleyman Serdar Kozat

TEACHING ASSISTANT

Arda Atalık

Abstract. This project is an online anomaly detection system on single source and multi source streaming network traffic data. Different from signature based intrusion detection systems, where the detection is based on pattern recognition of pre-stored anomalies, an anomaly detection based intrusion detection systems allows the system to find novel network attacks. In this project, anomaly based attack detection is achieved in three steps: data acquisition from a network and interpretation of network data in terms of quantitative features, state-of-art anomaly detection algorithms and the user interface. In the data acquisition phase, a large amount of network data packets are collected from multiple sources in the local network. These data packets are then interpreted and converted to meaningful features that can be used in detection algorithms. After the detection algorithms evaluate each packet, the results and alarms are provided to the user in a flexible and practical user interface.The network traffic data will have events per second up to 200-250 with the source number of around 4-5. As expected results in training and testing dataset with labels, it is desired for Area Under Receiver Operating Characteristic curve to be 0.6 at least, F1-Score for all models to be 0.4 at least,and Precision-Recall curve to be above no skill line.

PROJECT DESCRIPTION

ANODE is an online anomaly detection system on single source and multisource real-time streaming data. It is an Intrusion Detection System (IDS) used for anomaly detection on network traffic of interconnected computers network. This system is not integrated into another existing system. Network traffic data is collected packet by packet and relevant features are extracted. With machine learning techniques, anomalies on collected packets are monitored on an admin computer. Results are displayed in UI via live graphs.

With big data, there are countless streaming data coming from multiple sources in real-time. As a result, the number of network attacks has risen up and novel attacks are developed by attackers. The security of networked systems has become an important issue for individuals, companies and governments [2]. Similarly, it is important to produce high-technology cyber security systems in the defense industry.

ANODE is an anomaly-based IDS. There are two types of IDS's: anomaly-based and signature-based. Signature-based IDS is built on pattern recognition of pre-stored anomalies and works with high-accuracy for known anomalies [2]. However, its disadvantage is that it cannot detect any novel attack. Novel attacks should be predicted by developers of IDS beforehand and the efficiency of the system depends on the speed difference of developers and attackers creating new signatures. Hence, signature-based detection techniques are not capable of catching new, self-modifying network behavioural anomalies. Anomaly-based IDS's have the advantage to catch novel attacks [2]. The advantage of our system is its being anomaly-based and ability to catch novel attacks.

Design and Performance Specifications

Autonomous

1. The anomaly detection is performed online on multivariate data.
2. The final product is connected to multiple real-time data sources.
3. The algorithms learn periodically (once a day around six hours) by storing a portion of the incoming streaming data, otherwise, they will predict in the remaining time.
4. Final product is provided to the user via UI/UX on an application.
5. The anomaly statistics are displayed with live charts and histograms.
6. Model fitting of each model speed is at most $\sim$0.3s for each data point.
7. Anomaly detection works in real time, with data processing and anomaly detection performed under $\sim$2s for one network data packet.
8. Event per second (EPS): up to 200-250 with the source number: $\sim$4-5 (EPS observed in network traffic of one host:$\sim$50).
9. The chart's update time for each event detection is under 1 minute.
10. Data acquisition interval is under 2s.
11. Area Under ROC $>$ 0.6, also it is compared with some related anomaly detection works.
12. F1-Score $>$ 0.4.

13. The PR curves of the algorithms are above the no skill line (number of anomalous samples/number of all samples).

User Controlled

14. The user is able to determine a confidence percent threshold between 0-1 to distinguish anomalies depending on their anomaly output score.
15. Algorithm selection option is provided to the user and based on this selection it operates unsupervised anomaly detection.
16. The user is able to determine the hyperparameters of the algorithms, otherwise, the default parameters are used.
17. The user is able to download the results in csv format.
18. The UI is highly customizable in terms of time range and adding thresholds.

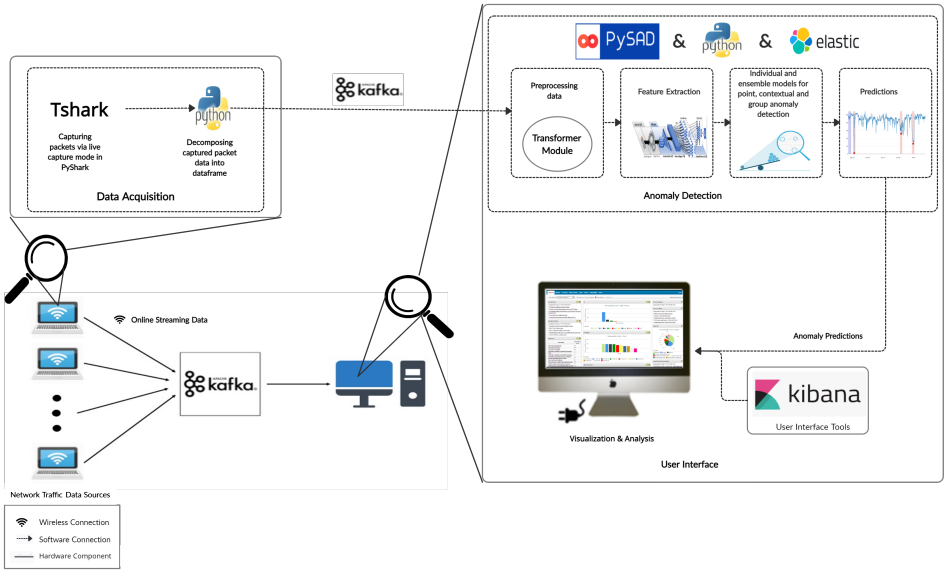

FIGURE 1. Big picture of the project.

Online streaming data coming from multiple network data sources are obtained via wireless connection. Tshark captures the network packets and PyShark extracts the raw features from network packets. Then, the data is sent to the transformer module. In this module, the useful features are extracted and selected to provide appropriate data to anomaly detection models. Some anomaly detection algorithms and their ensembles in PySAD library are used as the detection models. The preprocessed data is given to the models and the anomaly probabilities of the network packets are obtained as the output. The predictions are gathered from all models for each packet. Then, they are sent to Elasticsearch with their raw information. Finally, the packet information and anomaly predictions in Elasticsearh are visualized with live charts, graphs and histograms. The user can add thresholds to the graphs. The system gives alarm when a point is detected as anomaly.

MILESTONES

We reached multiple milestones that are in accordance with the three parts of the project. Four milestones are as follows:

1. **Data Acquisition Module:** This milestone is reached upon the completion of the data acquisition phase. The work packages include the live network data capture, decomposing data to proper data format, sending/acquiring the data to/from a queuing platform called Kafka (to process later), preprocessing, feature selection and extraction, and combining multi-source input. This milestone indicates that the data acquisition module is operating and able to capture and direct the data to following modules.

2. **Anomaly Detection Module:** This is the milestone that belongs to the anomaly detection phase. It covers the implementation of the models for online anomaly detection algorithms, developing the ensemble models, and creating a system flow module that the data instances can go from the beginning to the UI.

3. **Operating UI:** Designing the UI is the goal for these work packages. This milestone is completed after we showed the packet logs and their anomalousness probability on live plot using Elasticsearch's Kibana interface. Deploying the project and relevant parts to Docker platform was another work package for this milestone.

4. **Adjustments and Finalization:** This is the final milestone in which we optimize the anomaly detection algorithms, improve the general system performance in terms of speed and efficiency, get final feedback from our advisors and complete the final product.

DESIGN DESCRIPTION

Data Acquisition Module: PyShark, Python wrapper for TShark, is used for capturing network packets. Decomposition of these packet to proper data format which is Pandas dataframe is done one by one and individual packets are sent to queuing platform Kafka. Transformer module is preprocessing the data coming from Kafka and direct the data feature extraction. Autoencoder, PCA, LSTM autoencoder, and k− means clustering feature extractors are implemented, then data is directed to Anomaly Detection Module. For multi-source input, there are multiple data acquisitions and all of them direct data to Anomaly Detection Module in the admin computer.

Anomaly Detection Module: The anomaly detection algorithms provided by PySAD libray are used. Designed algorithms are to detect anomalies in streaming data. Therefore, it is convenient to use them since the data is real time and streaming in this project. To get more accurate results, the ensembles of the models are

used. This is because the algorithms work differently and they may focus on one aspect in detection. When they are combined, the different aspects in the detection can be focused at a time. The anomaly probabilities are obtained from the individual and ensemble models. To keep anomaly predictions and the raw packet information, an indexing platform called Elasticsearch, which helps the data get visualized in Kibana, is used.

UI/UX Module: To visualize the anomaly predictions to the user the project requires a UI solution. To achieve a customizable, easy to use and multi-functional user interface we have decided on using the Kibana platform. Kibana can be used to visualize it in different ways. These can be viewed in the results section but an example of this visualization include a dashboard, which contains different panels that show different models' predictions of data. Kibana is highly customizable in nature, and hence can be used in a variety of different ways with different visualization options. We can also visualize data as live plots, and the user can change the refresh interval as they please.

Another problem that we needed to solve was to increase user experience as much as we could. We have two solutions for this, one being taking inputs from the user to set certain thresholds and the other being an alarming system that alarms the user via mail if any anomaly is found (i.e. the data point is above the set threshold). For example, keeping in mind the fact that the detection algorithms output a probability value from 0 to 1, the user selects a threshold of 0.7. Then, a data point is identified as having a probability of 0.8 of being an anomaly, from which point on the system sends an alert to the user informing of this activity.

Equipment list: These are the packages used in python and other programs used in the project.

Packages	**Programs**
PySAD	Docker
PyShark	WireShark/TShark
Scikit-learn	Elasticsearch & Kibana
Tensorflow	Confluent Kafka

TABLE 1. Equipment list.

RESULTS AND PERFORMANCE EVALUATION

A flowing system was achieved. In the Figure 2, the Kibana dashboard and the anomaly probability predictions can be observed. In Table 2, time spent on several important steps of the project for a single data evaluation can be found.

Obtaining from Kafka	Data Transformer Speed	Feature Extraction Speed	Fiting Speed	Prediction Speed
0.05s	0.02s	0.07s	1.0s	0.1s

TABLE 2. Time spent on important steps of the project, for a single package.

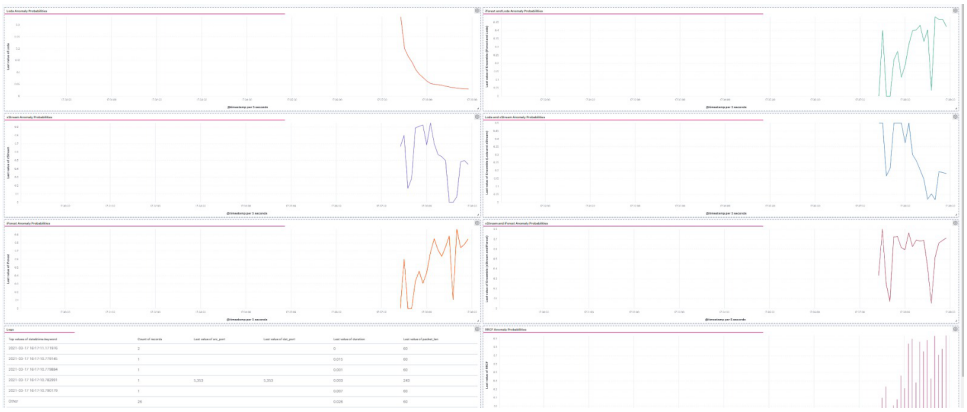

FIGURE 2. The Kibana dashboard that includes several panels.

Simulation Results: The results from one of the ensemble models consisting of xStream and iForest, tested on NSL-KDD data set. According to the figure below, the area under ROC is 0.76 (greater than 0.6, see Performance Specifications),the F1 score is 0.56 (greater than 0.4, see Performance Specifications) and the PR curve is above the No Skill curve.

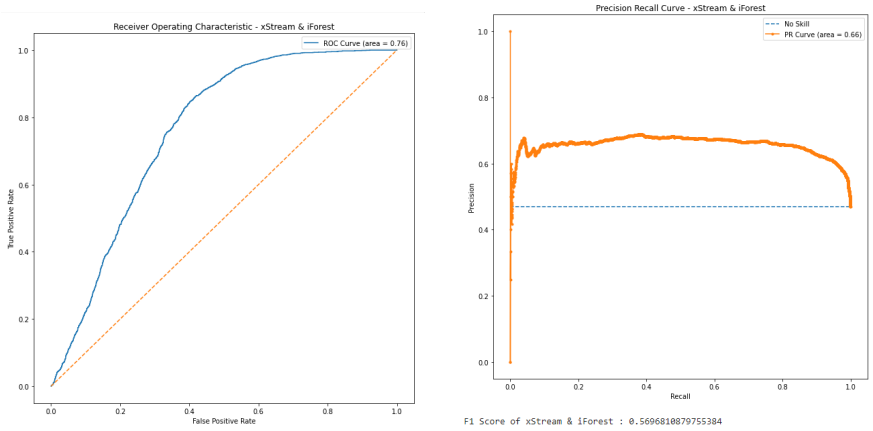

FIGURE 3. AUROC and PR curves for Ensemble model xStream and iForest on NSL-KDD dataset.

CONCLUSIONS AND FUTURE DIRECTIONS

For this project, our goal was to design a system that can detect novel anomalies on a computer network using a Big Data approach. The system we designed consists of three parts: data acquisition, anomaly detection and the user interface. For data acquisition, raw network data is analyzed and transformed into a form that can be used in anomaly detection algorithms. These algorithms include unsupervised learning and anomaly detection algorithms that allow the detection of anomalies in the streaming network system. After anomaly detection, each packet is evaluated in terms of anomalousness and displayed on a user interface for the user to act on the network. The user has the options to visualize the statistics and each packet's anomaly score along with the logs of the network. In addition, the sensitivity of the alerts can be chosen according to a customizable probability threshold.

On a general note, anomaly detection for network systems usually perform signature based detection. These are pattern based detection system for well-studied attacks and anomalies that can match activities on a network to previously determined examples. However, this signature based approach disallows the detection of novel attacks. Outcome of our project shows that a machine learning approach supported by Big Data methods can be used for detection. This project only performs detection of the anomalies, the most obvious future direction for this area is the identification of the anomaly type. Anomaly types can vary from wrong password entry to more serious attacks such as denial-of-service, injection and fuzzing attacks. Although our system detects the anomaly of a packet, another system that reports the type of the anomaly can be developed with the help of supervised learning algorithms to match the anomaly type or clustering algorithms to group the attacks.

Moreover, the core system consisting of anomaly detection algorithms is planned to be developed for other application areas not limited to network traffic. It can possibly be used as a real-time alert system for various purposes including intrusion detection, predictive maintenance and finance later.

REFERENCES

[1] Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, *"Featuresdimensionality reduction approaches for machine learning based network intrusion de-tection"*, Electronics, vol. 8, p. 322, Mar. 2019.doi:10.3390/electronics8030322.

[2] Veeramreddy, V. Prasad, and K. Prasad, *"A review of anomaly based intrusion detectionsystems"*,International Journal of Computer Applications, vol. 28, pp. 26–35, Aug. 2011.doi:10.5120/3399-4730.

BEHIND THE SCENES