**CSCE -5215 MACHINE LEARNING**

**Enhanced Network Intrusion Detection with Threat Intelligence Integration and Hybrid Models**

|  |  |
| --- | --- |
| Abbet | 11632812 |
| Karthik | 11551393 |
| Karthikeya | 11708067 |
| Hemanjan | 11700076 |

**Abstract**

The purpose of this research is to use a proper machine learning techniques to improve network intrusion detection. As we know Conventional approaches struggle to identify new risks in network traffic. The research suggests an improved intrusion detection system that uses machine learning algorithms to identify and adjust to new cyberattack patterns in order to address this problem. This system's ability to differentiate between normal and unusual behavior is a result of its training on a variety of network traffic datasets. By using reliable feature extraction techniques and an effective machine learning architecture, we can say the suggested solution seeks to improve intrusion detection efficiency and accuracy while strengthening network security against new cyberthreats. Our project makes significant contributions to the creation of a defence system against complex network threats that is more resistant and flexible.

**Introduction**

Defense process by aiming security administrators in forewarning them about malicious behaviors such as intrusions, attacks, and malware. Having IDS is a mandatory line of defense for protecting critical networks against these ever-increasing issues of intrusive activities. So, research on IDS domain has flourished over the years to propose the better IDS systems. However, many researchers struggle to find comprehensive and valid datasets to test and evaluate their proposed techniques (Koch et al., 2017) and having a suitable dataset is a significant challenge itself (Nehinbe, 2011).

**History:**

**1990s - Emergence of Script Kiddies and Hacking Groups:**

Going back decade of the 1990s witnessed an increase in network intrusions conducted by so-called "script kiddies."At that time these people used readily available hacking tools, but they lacked in-depth knowledge. At the same time, hacking groups such as Chaos Computer Club and Lizard Squad became well-known for their coordinated and politically motivated breaches.

**Early 2000s - Spread of Malware and Worms:**

We came to know the development and spread of malware and worms increased rapidly in the early 2000s. The Famous instances are the Code Red and Nimda worms, which took advantage of flaws in Microsoft software to propagate infections and caused damage. This all done in early 2000s.

**Mid-2000s - Advanced Persistent Threats (APTs):**

After that coming to mid 2000s Advanced Persistent Threats (APTs) are targeted, sophisticated attacks that first appeared in the middle of the 2000s. These highly skilled attacks, which were frequently supported by governments, sought to obtain long-term, illegal access to confidential data. Flame, Duqu, and Stuxnet are a few well-known APTs.

**2010s - Large-scale Data Breaches:**

Coming to 2010s we can say massive Large-scale data breaches have increased in the 2010s. Cybercriminals gained access to a great deal of private data by targeting large enterprises, governmental organizations, and healthcare facilities. The Equifax hack in 2017 and the Target breach in 2013 are two notable instances.

**Present - Evolving Threat Landscape:**

The field of network intrusion is still developing today. At any day Attackers use a growing number of complex strategies we need to be prepared for, including supply chain breaches, ransomware assaults, and zero-day exploits. Machine learning and artificial intelligence are used by both attackers and defensive strategies to either avoid detection or provide protection.

**Goals and Objectives**

**Motivation:** The main motive of this project is to improve the outdated network intrusion detection systems (NIDS) to the updated system according to the growing advanced styles and new approaches of cyber threats. The need for an improved NIDS in the fast growing modern world that can adjust to new threats is addressed by this project.

**Significance:** The significance of this project is it holds a very great importance as it has the ability to greatly enhance network security and capacity to withstand against a wide range and variety approaches of cyber threats.

**Objectives:** The main objective of this project is to develop a hybrid intrusion detection system integrating threat intelligence feeds, thereby we can enhance the detection accuracy and response to both known and unknown newly emerging threats.

**Features:** The system features real-time threat intelligence integration, which is a combination of signature-based and anomaly-based detection methods, and also a dynamic update mechanism to adapt to new or unknown threats.

**Problem Statement**

The main objective of this project is to boost the accuracy and efficacy of network intrusion detection systems. As we all know In the rapidly evolving world of cybersecurity, network intrusion detection plays an important part in protecting our private data and maintaining the security of our digital infrastructures. The Advanced and continuously changing approaches of cyber attacks are difficult for traditional intrusion detection systems to detect with accuracy. By the combination of threat intelligence and hybrid machine learning models, our project aims to enhance network intrusion detection through more accurate and adaptable threat detection.

**Dataset and Attributes**

As we know that a comprehensive dataset is very crucial for training and also testing the proposed hybrid models for intrusion detection. we made our dataset to be created by amalgamating data from diverse sources, with an ample quantity to facilitate effective training and testing. Key attributes include:

**1. Destination Port:** The destination port here in dataset is a number in a network packet that identifies the specific service or application for which it is meant for.

**2. Flow Duration:** Flow duration is referred according to dataset as the time it takes for a set of packets to travel from the source to the destination, which giving us information about the duration of a network communication flow.

**3. Total Fwd Packets**: Total forward packets is referred as the total count of packets sent from the source to the destination in the network flow.

**4. Total Backward Packets:** Total backward packets is defined as the total count of packets which are sent from the destination back to the source in a network flow.

**5. Total Length of Fwd Packets:** The total length of forward packets is the sum of the sizes of all the packets which are sent from the source to the destination in a network flow.

**6. Total Length of Bwd Packets:** The total length of backward packets is the sum of the sizes of all the packets which are sent from the destination back to the source in a network flow.

**7. Fwd Packet Length Max:** Forward packet length max is the largest size among all packets which are sent from the source to the destination in a network flow.

**8. Fwd Packet Length Min:** Forward packet length min is the smallest size among all packets which are sent from the source to the destination in a network flow.

**9. Fwd Packet Length Mean:** Forward packet length mean is the average size of packets which are sent from the source to the destination in a network flow.

**10. Fwd Packet Length Std:** Forward packet length standard deviation measures the variation in sizes among packets which are sent from the source to the destination in a network flow.

**11. Bwd Packet Length Max:** Backward packet length max is the largest size among all packets whcih are sent from the destination back to the source in a network flow.

- Minimum size of packets transmitted from destination to source in a network flow is known as backward packet length minimum.

- Backward packet length mean is the average size of packets transmitted from the destination back to the source in a network flow.

- Backward packet length standard deviation measures the variability or dispersion in sizes among packets transmitted from the destination back to the source in a network flow.

- Flow bytes per second represent the rate of data transmission in bytes per second within a network flow.

- Flow packets per second indicate the rate of packet transmission within a network flow.

- Flow inter-arrival time (IAT) mean represents the average time between the transmission of consecutive packets in a network flow.

- Flow inter-arrival time (IAT) standard deviation measures the variability or dispersion in the time between consecutive packets in a network flow.

- Flow inter-arrival time (IAT) max denotes the maximum time between the transmission of consecutive packets in a network flow.

**Methodology**

The project's methodology is divided into multiple phases, starting with data collecting and preprocessing, then encoded the data, standardized , and normalized the data to feed it to the machine learning algorithms.

**Data Cleaning:** In our data-set we have 3033050 data points and 82 features with more 1.9GB memory. we have cleaned the data and streamlined it using following methods.

**Handling Missing Values**: Removed missing values in the dataset. Since these rows have more than 15 null values out of 80 columns so, we are dropping them. We can fill the null values using mean, median, or mode but we choose to drop them. Because we have 3033050 records and these null value rows contain most of the values as null and their percentage is less than 10% in the whole dataset. It's better to drop them.

**Visualizing Attack Types**: Use data visualization techniques like count plot to show the distribution of different attack types in the dataset. We are using matplotlib modules for this segment for visualization of the most common attack type and we have found out it to be BENIGN

**Visualizing Flow Duration for Different Attack Types**: We have used the matplotlib module for this segment to as the duration of the attack gives us a clear picture for us to understand how long does it take get into the system we have taken the times for each attack.

**Distribution of Total Backward Packets for Different Attack Types**: We have balanced our target column by under sampling the "BENING" categorical class( reduced from 80% to 38.1%) and increasing the minor class percentage.

**heat map:** We plotted heatmap for better understating the relation between input columns.

**Scaling the Data**: scaled the data using minmax module.

**Feature Selection**: Use feature selection methods like Anova test or feature importance analysis to identify and keep the most relevant features for building a robust intrusion detection model.

**Training Machine Learning Models**: Utilize hybrid machine learning models that combine the strengths of various algorithms such as deep learning, ensemble methods, and traditional rule-based systems to improve the accuracy and adaptability of the intrusion detection system.

**Integration of Threat Intelligence**: Incorporate mechanisms for seamlessly integrating threat intelligence into the system.

1. The Random Forest Classifier is a model used for signature-based detection and is part of the ensemble models that can handle both classification and regression tasks. Random Forest can be trained on labeled data, learning patterns and signatures of known attacks.

2. The Decision Tree Classifier is a model based on decision trees and is used for classification tasks. Decision trees create a set of rules based on historical data, making them suitable for identifying known attack patterns.

3. The K-Nearest Neighbors (KNN) Classifier is a simple algorithm that classifies based on the majority class of its k-nearest neighbors. KNN classifies instances based on the majority class among their nearest neighbors, making it suitable for detecting anomalies in network traffic

4. The One-Class SVM (Support Vector Machine) Classifier is an SVM model trained to identify anomalies in a one-class setting. One-Class SVM is specifically designed for anomaly detection, learning the characteristics of normal data and identifying deviations from it.

5. The Isolation Forest Classifier is a model that uses the isolation forest algorithm to detect anomalies by isolating instances in trees. Isolation Forest isolates instances in trees to detect anomalies, making it effective for identifying unusual patterns in network traffic.

6. The Elliptic Envelope Classifier is an algorithm used to estimate the covariance of a dataset and is useful for detecting outliers. Elliptic Envelope estimates the covariance of a dataset and is useful for detecting outliers, making it suitable for anomaly detection in intrusion detection.

7. The Voting Classifier is a hybrid model that combines predictions from multiple models, allowing it to leverage both signature-based models (e.g., Random Forest, Decision Tree) and anomaly-based models (e.g., One-Class SVM, Isolation Forest) for a more comprehensive intrusion detection system

**Implementation**

1. **Importing Required Libraries:** First we have imported all the required libraries

2.**Reading dataset:** Read data-set using pandas read\_csv method. Dataset size is 1.8Gb

3. **EDA:**

1. Printing Basic information about our data set using methods such as
   1. head() : Displayed 1st 5 records of the dataset
   2. shape : Printed no of rows and columns of dataset (3033050, 82)
   3. info() : For displaying column names and their data types.
   4. nunique() : For displaying number of unique values in each column

A table with text and numbers

Description automatically generated with medium confidence

Fig: head() of the dataset.

1. Descriptive Statistics:

A table with numbers and symbols

Description automatically generatedPrinted all the Descriptive Statistics of the dataset using describe() method.

1. Data Cleaning:
   1. Null Values cleaning: First we checked if there is any column with numeric or infinite values. If present then we will display them.Removed all the rows which have more than 15 null values.Dropped 3 empty columns : 'ipv4','Label', 'date'. Which are not useful as they are empty.
   2. Handling Infinite values: Replaced all the infinite values using the below code:

tf\_df.replace([np.inf, -np.inf], np.finfo('float64').max, inplace=True)

* 1. A screenshot of a computer code

     Description automatically generated Removing Duplicate rows:
  2. Balancing Class distribution in ' Label' Column: Used under sampling method to reduce “BENING” class size. And used RandomOverSampler to increase size of minor classes.

=>Below figures represent pi plot and count plot of ‘ Label’ column.

A pie chart with different colored circles

Description automatically generated

A graph of a number of people

Description automatically generated

We can see that classes in “Label” column is balanced. Before balancing “Bening” class has 80% of weightage now its reduced to 20%.

* 1. Plotting data distributions of different column:

A screenshot of a computer

Description automatically generated

A group of graphs showing different types of bmi

Description automatically generated with medium confidence

From the above plots we can see that data in these columns are not normally distributed. Most of them have positive skew. We need to handle the skewness before feeding the data to our models.

* 1. A group of graphs with text

     Description automatically generated with medium confidenceBox plots of the columns:

From above boxplot result we can say that there are outlier we need to handle them. Reason for handling outliers: They can reduce the performance out our models.

* 1. Plotting Correlation matrix:

A screenshot of a computer screen

Description automatically generated

Above plot represent the seaborn heatmap for our dataset’s numeric columns. Blue indicates less correlation between columns where as dark red indicates high correlation between the columns.

* 1. Detection and handling outlier and skewness:

A screenshot of a computer code

Description automatically generated

Fig: We are finding outliers using IQR method and printing them.

A screenshot of a computer program

Description automatically generated

Fig: Code and output of handling skweness.

We have reduced the skewness to significant amount using Winsor method.

* 1. Scaling Data: We performed data scaling using MinMaxScaler method to improve model perfomance. If the data is in different scales them model might be fooled by columns with higher valued.

A close-up of a computer code

Description automatically generated

1. Feature Selection:
   1. Anova Method:

A screenshot of a computer program

Description automatically generated

Above picture represent the Anova Method for removing the un-important columns. Initially we have 80 columns after anova test we got around 60 columns. Since our data set consist of 33 lakhs recods We cannot directly rely on anova test. So we have taken chi2 test.

* 1. Chi2 Method:

A screenshot of a computer code

Description automatically generated

Fig: Chi2 code

A table with numbers and letters

Description automatically generated

Fig:output of Chi2 test.

From Chi2 test we have selected top 25 records as important features.

1. Model Implementation:

We have implemented signature based model like RandomForest,Decision tree, and Kneighbors algorithm and Anomaly models like OneClassSVM, IsolationForest, and EllipticEnvelope. For hybrid models we combined both Signature and Anomaly models to get better performance.

i)Signature Based models implementation:

A screenshot of a computer code

Description automatically generated

ii) Anomaly and hybrid models implementation:

A screenshot of a computer program

Description automatically generated

We achieved 98% accuracy for our hybrid model.

A screenshot of a computer screen

Description automatically generated

1. Real time threat intelligence integration:

A screenshot of a computer program

Description automatically generated

We extracted data from AlienVault, which is a open source site for fetching data. Here we used aip for accessing the data. Initially we generated api key from alienvault site through api key we are accessing the data.

A screenshot of a computer

Description automatically generated

Fig: extracting data from alienvault api.

**Project management :**

The team is organized to make the most of each person's strengths. There are work distributions like data

Abbet - Abbet took care all the Data Collection and Preprocessing the data he have collected like data organizing, he also contributed Gathering information about network traffic from multiple sources. Preprocess & clean the data. he provided the Initial exploratory data analysis.

Hemanjan – Hemanjan taken care about Model Selections and Training the selected models he has done Experiments with most of the ML models which will be appropriate for project and Developed hybrid models. He contributed Training and optimizing model parameters. He made Model evaluations.

Karthikeya – Karthikeya taken care about Threat Intelligence Integration and Deploymen he is responsible for Incorporating real-time threat intelligence feeds. he Implemented automated threat intelligence updates. and Developed and deployed the intrusion detection system.

Karthik – Karthik taken care about Data Cleaning, Feature Selection, EDA, and Reporting He is responsible for Handling feature selection to identify important features. He Conducted detailed exploratory data analysis (EDA). And Prepare project reports and visualizations.

The team continuously monitors and adjusts the project. Even though we face many errors initially We evaluate and provide feedback to address all challenges and improve performance. We also focus on documentation to increase transparency and knowledge sharing.

**Threat Intelligence Integration**

This project's primary goal is to improve the intrusion detection system's prediction capacity through the inclusion of real-time threat knowledge.We have provided Updated information on the most recent threat landscape is offered by the integration of threat intelligence feeds from AlienVault OTX and also other reliable sources.

**The integration process involved:**

Establishing secure and reliable channels for receiving threat intelligence data.

Automating the ingestion and parsing of data feeds to extract relevant information.

Mapping the threat data to the network traffic features to enable proactive detection.

Developing a system for continuously updating the model's knowledge base with new threat intelligence.

The intrusion detection system can now able to immediately detect and respond to new and complex threats by using both historical data and up-to-date global cyber threat information thanks to this connection. We, Based on this information,Make sure the system was designed to dynamically update its indicators of compromise (IOCs) and detection patterns, by keeping up with the attackers' changing strategies.

Our system uses real-time threat intelligence to perform context-aware detection and heuristic analysis, going beyond simple detection. We enhance our detection power by taking into account factors like network traffic source and reputation, which helps us identify dangers such as advanced persistent threats (APTs) and zero-day attacks. Our hybrid detection models work in perfect harmony with threat intelligence streams, so this isn't a one-man show. When combined, they create a force to be reckoned with that crafts a highly adaptive and resilient network intrusion detection system that raises the bar for security effectiveness

**Deployment and Recommendations**

The project moved from a controlled development environment to a live network environment during the deployment phase. On a test network, we made the intrusion detection system was set up to track traffic and we spot possible threats. The team was able to make the required modifications and we later fine-tune the system in a real-world setting during this live testing phase.

**Recommendations for organizations looking to adopt this system include:**

Ensuring network infrastructure is prepared for integration with the new system.

Training IT personnel to understand the functionality and output of the system.

Establishing protocols for responding to detected threats.

Regularly updating the threat intelligence database to maintain system efficacy.

One must see behind the latest technological terms when implementing a system such as this one. Think once about the wider picture, such as possible privacy issues and the requirement for a strong cybersecurity policy. This is about ensuring that your fancy security configuration coexists peacefully with ethical and privacy considerations,this is not just about bits and bytes.

**Results**

**Signature Based Models: Table Of Precision:**

table of precision, recall, and f1-score values for two different machine learning models: RandomForest and DecisionTree. The table shows that both models have very high precision, recall, and f1-score values, indicating that they are both very good at classifying the data.

**A screenshot of a computer

Description automatically generated**

**Anomaly Based Models:**

The table in the image shows the precision, recall, and f1-score values for two different machine learning models: KNeighbors and OneClassSVM.

**A screenshot of a computer screen

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

table of precision, recall, and f1-score values for two different machine learning models: IsolationForest and EllipticEnvelope, evaluated on a dataset of 4000 samples with 7 classes.

**A screenshot of a computer screen

Description automatically generated**

**Discussion**

The results were discussed here, and we make sure it became clear that combining threat intelligence with hybrid detection models is a potential way to detect the network intrusions. We believe the strategy we have employed in this study improved the flexibility and responsiveness of classic NIDS to new threats, thereby addressing its weaknesses.

The possibility of such type of systems evolving through machine learning techniques is also explored in this conversation, stressing the significance of ongoing learning and adaptation in the cybersecurity domain. Later studies may concentrate on integrating more complex artificial intelligence methodologies,they may including deep learning and reinforcement learning,use them to bolster detection precision and avoid false positives.

**Conclusion**

Having a reliable, publicly available IDS evaluation datasets is one of the most fundamental concerns of researchers and also producers in this domain. In this paper, we have monitored the state-of-the-art in the IDS dataset generation and evaluation by analyzing the eleven publicly available datasets which are available since 1998 they are limited because of the lack of the traffic diversity and volumes, anonymized packet information and payload, constraints on the variety of attacks, and lack of the feature set and metadata. Then we have generate a new IDS dataset by including seven common updated family of attacks that met real worlds criteria and is publicly available (http://www.unb.ca/cic/datasets/IDS2017.html). On the evaluate section, we fist extract the 80 traffic features from the dataset and clarify the best short feature set to detect each attack family using RandomForestRegressor algorithm. Afterwards, we examine the performance and accuracy of the selected features with seven common machine learning algorithms. Finally, we compare the quality of the generated dataset by searching for common mistakes and criticisms of other synthetically created datasets, based on the 11 criteria of the last proposed dataset evaluation framework with other publicly available datastes since 1998till 2016. In the future, we sould like to increase number of PCs as well as conducting more up to date attacks

**References**

Cui, Jin; Wang, Zhijian; Wu, Yulei. "Hybrid Intrusion Detection with Weighted Signature-Based and Anomaly-Based Intrusion Detection Mechanisms." IEEE Transactions on Information Forensics and Security, 2017.

"Machine Learning in Network Intrusion Detection: A Comparative Analysis of Techniques, Datasets, and Metrics"

Alazab, Mamoun, et al. "Machine Learning in Network Intrusion Detection: A Comparative Analysis of Techniques, Datasets, and Metrics." IEEE Access 8 (2020): 121984-122013.

"Adversarial Attacks and Defenses in Intrusion Detection Systems: A Comprehensive Review"

Alsulami, Saad, et al. "Adversarial Attacks and Defenses in Intrusion Detection Systems: A Comprehensive Review." IEEE Access 8 (2020): 123672-123691.

"Feature selection and machine learning for anomaly-based intrusion detection: A comprehensive review"

Soni, Prafull, Pradeep Tomar, and Manish Kumar. "Feature selection and machine learning for anomaly-based intrusion detection: A comprehensive review." Journal of King Saud University-Computer and Information Sciences (2020).