Network Anomaly Detection Using Machine Learning

Nguyen Truong Huy, Nguyen Tien Khoi

May 21, 2024

Abstract

Since the beginning of the 20th century, human society has increasingly embraced the application of technology in work and life. More and more technological devices are being connected via the internet to communicate information with one another. This information includes crucial user data that needs to be secured. Consequently, cybersecurity has become increasingly important in the field of technology. New technologies are continually being developed to combat hacker attacks. Recently, we have seen machine learning being applied to various fields with great effectiveness, and it is also being increasingly used in the field of intrusion detection to enhance cybersecurity. However, machine learning is not omnipotent and has its own weaknesses. This study will provide a fresh perspective on the application of machine learning in intrusion detection and the existing limitations.

I. Introduction

1. Cyber Attack

a. What is Cyber Attack

Each expert in the field of information security interprets this term according to their own understanding. For example, "intrusion" is any action that takes the system from a safe state to a dangerous state. This term can be explained as follows: "intrusion" is the destruction of the information security policy or any action that leads to the destruction of the integrity, confidentiality, and availability of the system and the information processed within the system.

An attack is the intentional illegal activity of an individual or group that exploits the vulnerabilities of an information system and disrupts the availability, integrity, and confidentiality of the information system. A cyber attack involves actions or a sequence of actions linked together to exploit the vulnerabilities of information systems, thereby realizing threats by compromising the system's security features.

b. Cyber Attack Models

Traditional Attack Model: This follows the principle of "one-to-one" or "one-to-many," meaning the attack originates from a single source.

Distributed Attack Model: This follows the principle of "many-to-one" or "many-to-many," meaning the attack can originate from multiple distributed sources.

2. Anomaly-Based Intrusion Detection System (AIDS)

This system compares captured events with the normal behavior of the entity. Any action that deviates from normal behavior is considered an intrusion. This means if we establish a dataset of normal activities for a system, we can then flag all states that differ from the newly established data.

The advantage of the system is that it can detect previously unknown attacks. However, this advantage comes at the cost of a high rate of false alarms because, in practice, anomalies do not necessarily indicate intrusions.

As the number of new attacks increases rapidly, it is challenging for the misuse detection approach to maintain a high detection rate. Additionally, as attacks become more sophisticated and prolonged, maintaining the signature database becomes burdensome.

On the other hand, the anomaly detection approach for detecting intrusions through experiential machine learning is relatively easy to maintain. A variety of techniques are used in anomaly detection. They are based on common features, primarily classified into four different models: statistical-based, feature-based, immunity-based, and machine learning (data mining).

Intrusion detection can be considered a binary classification problem because it aims to distinguish between normal and abnormal behavior.

Anomaly Detection System

An anomaly detection system is an Anomaly-Based Intrusion Detection System (AIDS). This system compares captured events with the normal behavior of the entity. Any action that deviates from normal behavior is considered an intrusion. This means if we establish a dataset of normal activities for a system, we can then flag all states that differ from the newly established data.

There is a significant difference between anomaly and misuse detection:

Anomaly detection uses techniques based on characteristics of normal behavior to detect characteristics of bad behavior. Misuse detection relies on previously known bad behaviors to detect repeated bad behaviors. The mechanism of the system is described in the following figure:

The advantage of the system is that it can detect previously unknown attacks. However, this advantage comes at the cost of a high rate of false alarms because, in practice, anomalies do not necessarily indicate intrusions.

As the number of new attacks increases rapidly, it is challenging for the misuse detection approach to maintain a high detection rate. Additionally, as attacks become more sophisticated and prolonged, maintaining the signature database becomes burdensome.

On the other hand, the anomaly detection approach for detecting intrusions through experiential machine learning is relatively easy to maintain. A variety of techniques are used in anomaly detection. They are based on common features,

primarily classified into four different models: statistical-based, feature-based, immunity-based, and machine learning (data mining).

Intrusion detection can be considered a binary classification problem because it aims to distinguish between normal and abnormal behavior.

Machine learning techniques are the most suitable for intrusion detection:

- Machine learning techniques extract knowledge directly from previous data, eliminating the need for manual knowledge extraction.
- They can generate models based on incomplete data.
- Machine learning techniques can represent abstract knowledge, a capability that makes them suitable for handling large amounts of data.

Advantages of the System:

- AIDS can detect abnormal behavior and therefore has the ability to detect signs of attacks without detailed knowledge of those attacks.
- Anomaly detectors can generate information that can be used to define signatures for misuse detectors.

Disadvantages of the System:

- Anomaly detection methods often generate a large number of false alarms due to unpredictable user or network behavior.
- Anomaly detection methods typically require large training datasets.

II.Methodology

1. Material

KDD'99 is the name of a dataset created for the 1999 Knowledge Discovery and Data Mining (KDD) competition and is derived from the data captured in a simulated military network environment. It includes:

- Training Data: The training dataset contains 4,898,431 connection records.
- Test Data: The test dataset contains 311,029 connection records.

Each connection record consists of 41 features that can be classified into three categories:

- Basic Features: Derived from packet headers without inspecting the payload.
- Content Features: Domain knowledge within the payload of the original TCP packets.
- Traffic Features: Computed with respect to a 2-second time window and capturing the characteristics of the connections.

The dataset includes both normal and attack traffic. Attacks fall into four main categories:

- DoS (Denial of Service): A Denial of Service (DoS) attack renders computer resources unavailable to legitimate users. The most common form of DoS attack is overwhelming the computer resources with so many useless requests that legitimate users cannot access them. There are two types of DoS attacks:
  - DoS: An attack from a single entity or a small group of entities.
  - DDoS: A distributed form of DoS attack.

  Types of DoS attacks in KDD Cup 99 include Back, Land, Neptune, Pod, Smurf, and Teardrop.

- R2L (Remote to Local): In this type of attack, hackers attempt to gain access to a computer system by sending packets to the system over a network. The main goal of a Remote to Local Attack is to illegally view or steal data, install viruses or other malware on another computer, network, or system, and cause damage to the targeted computer or network. Types of Remote to Local Attacks in KDD Cup 99 include Ftp-write, Guess-passwd, Imap, Multihop, Phf, Spy, and Warezmaster.

- U2R (User to Root): In this class of attacks, hackers with the privileges of a normal user attempt to gain unauthorized access to the system with the highest privileges (system administrator privileges). In the KDD Cup 99 dataset, there is a type of attack referred to as U2R (User to Root), which includes: buffer-overflow, loadmodule, perl, and rootkit attacks.

- Probe: In this type of attack, hackers scan a network or computer to find vulnerable points that can be exploited to compromise the system. This is similar to monitoring and surveillance of the system. A common method of this type of attack is to perform port scans on a computer system. By doing this, hackers can obtain information about open ports, running services, and other sensitive information such as IP addresses, MAC addresses, firewall rules, and more. Types of probing attacks in KDD Cup 99 include Ipsweep, Nmap, Portsweep, and Satan.

2. Methods

There are three machine learning algorithms that we find suit best solving the network anomaly detection problems.

2.1: Logistic Regression

Logistic regression is one of the simplest techniques used to classify labeled datasets. Logistic regression is well-suited for anomaly detection due to its simplicity and effectiveness in binary classification tasks. It models the probability that a given input belongs to one of two classes (normal or anomalous) using the logistic function, which outputs a value between 0 and 1. The algorithm

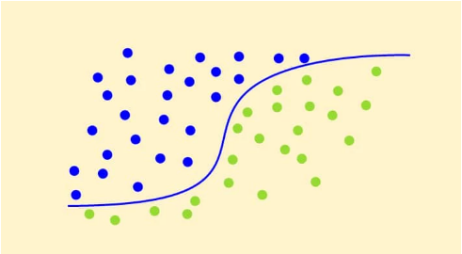

Figure 1: Simple illustration for logistic regression

can be used for both intrusion detection and predicting the likelihood of future anomalies.

2.2: Random Forest Classifier

We use Random Forest Classifier for anomaly detection due to its robustness, accuracy, and ability to handle high-dimensional data. It builds multiple decision trees and merges their results, reducing the risk of overfitting and improving generalization. Random Forests can capture complex interactions between features and are resilient to noisy data, making them effective in identifying subtle anomalies. Additionally, they provide feature importance scores, aiding in understanding the factors contributing to anomalies. Their inherent ensemble nature ensures stability and reliability, crucial for detecting irregularities in cybersecurity and other fields.

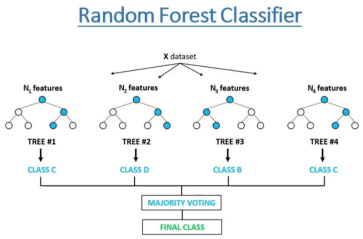

Figure 2: Simple illustration for Random Forest

2.3: Support Vector Machine (SVM)

Support Vector Machines (SVM) are powerful supervised learning models used for classification and regression tasks. SVM is highly effective for anomaly detection because it constructs a hyperplane or set of hyperplanes in a high-dimensional space that separates data points into different classes (normal or anomalous) with a maximum margin.

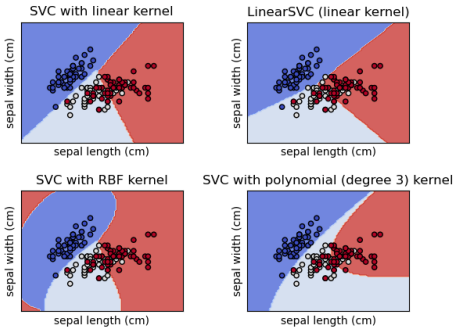

Figure 3: Simple illustration for SVM

III. Experiments

1. Evaluation Metrics

To assess the performance of our models for network intrusion detection, we employed several evaluation metrics, which are crucial for understanding their effectiveness in distinguishing between 'attack' and 'normal' network activities. These metrics provide a comprehensive view of the model's predictive power and its operational utility in a real-world setting.

- **Precision**: Precision is especially significant in scenarios where the cost of false positives (i.e., incorrectly predicting an attack) is high. For our models, high precision for detecting attacks (labeled as 'attack') indicates a high reliability in the predictions of actual attacks. Conversely, the precision for the 'normal' class is low, meaning there is some room for improvement in avoiding false alarms.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- **Recall**: Recall, or sensitivity, measures the model's capability to detect all potential threats. Recall measures the ability of the model to find all the relevant cases (i.e., all actual attacks). High recall is critical in security contexts because missing an attack can be very costly.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- **F1-Score**: The F1-score combines precision and recall into a single metric by taking their harmonic mean. It is particularly useful in uneven class distributions. If a model achieves a high F1-score for all classes, it suggests a balanced performance between precision and recall across both categories.

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Accuracy**: Accuracy reflects the percentage of total correct predictions (both attack and normal) made out of all predictions. This metric, while providing a general sense of performance, should be interpreted in the context of the balanced precision and recall scores given the potentially imbalanced nature of network intrusion datasets.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Predictions}}$$

2. Comparative Results

Table 1 provides a comparative analysis of three different machine learning models—Logistic Regression, Support Vector Machine (SVM), and Random Forest Classifier—used for network intrusion detection. The evaluation metrics include Accuracy, Precision, Recall, and F1-score, which are essential for assessing the performance of these models in distinguishing between 'attack' and 'normal' network activities.

From the results presented in Table 1, we observe the following key points:

- **Accuracy:** The SVM model achieves the highest accuracy at 0.86, indicating it has the highest overall correct predictions among the models tested. Logistic Regression follows with an accuracy of 0.84, and the Random Forest Classifier has the lowest accuracy at 0.82.

- **Precision:** Precision is crucial in scenarios with high costs for false positives. The SVM model demonstrates the highest precision at 0.87, suggesting it has the best performance in correctly identifying 'attack' instances. Logistic Regression and Random Forest Classifier have precision values of 0.85 and 0.84, respectively.

- **Recall:** High recall is vital for security contexts to minimize missed attacks. The SVM model again outperforms the others with a recall of 0.87. Both Logistic Regression and Random Forest Classifier have a recall of 0.85 and 0.84, respectively, indicating room for improvement in detecting all potential threats.

- **F1-score:** The F1-score balances precision and recall. The SVM model has the highest F1-score at 0.86, indicating a balanced and robust performance across both precision and recall. Logistic Regression has an F1-score of 0.84, while Random Forest Classifier has the lowest at 0.82.

In summary, the SVM model outperforms both Logistic Regression and Random Forest Classifier across all evaluation metrics, making it the most effective model for network intrusion detection in this comparative study.

3. Discussion

The SVM model performs the best among the compared models, achieving the highest scores across all evaluation metrics. The superiority of SVM can be attributed to its ability to handle complex decision boundaries between classes, especially when the data is non-linear. SVM optimizes the margin between data

Table 1: Performance Metrics for Different Network Intrusion Detection Models

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.84	0.85	0.85	0.84
SVM	**0.86**	**0.87**	**0.87**	**0.86**
Random Forest Classifier	0.82	0.84	0.84	0.82

points and the decision boundary, leading to higher performance in detecting attack patterns compared to Logistic Regression and Random Forest Classifier. Although Logistic Regression and Random Forest Classifier also demonstrate good performance, they do not achieve the same level of balance and accuracy as SVM. Logistic Regression may be limited in handling complex, non-linear data, while Random Forest Classifier, though robust in many scenarios, may struggle with the diversity and imbalance of attack patterns.

a. Limitations and Future Research Directions

Some limitations of this study include:

- Imbalanced Data: Network intrusion datasets often have imbalanced distributions between 'attack' and 'normal' classes, which can affect model performance. Techniques for handling imbalanced data could be applied to improve performance.
- Generalizability: While the models were trained and tested on a specific dataset, their ability to generalize to other network environments needs further examination.
- Training Time: SVM, although effective, can be time-consuming and resource-intensive to train on large datasets. Optimization methods or variations of SVM could be explored to enhance training efficiency.

b. Future Research Directions

Future research directions could include:

- Integration of Deep Learning Techniques: Deep learning models, such as Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN), could be tested to see if they provide higher performance.
- Advanced Analysis of Attack Patterns: Utilizing advanced analytical techniques to gain a deeper understanding of attack patterns and improve model detection capabilities.
- Hybrid Detection Systems: Combining multiple models to create a more robust detection system, leveraging the strengths of each model.

IV. Conclusion

In this study, we have explored the application of machine learning techniques in network anomaly detection using the KDD'99 dataset. We have focused on three primary algorithms: Logistic Regression, Random Forest Classifier, and Support Vector Machines (SVM). Each of these algorithms has its own strengths and weaknesses, making them suitable for different scenarios of network intrusion detection. Our analysis highlights the importance of selecting the appropriate machine learning technique based on the specific requirements and characteristics of the network environment. Future work can involve the exploration of advanced machine learning models, such as deep learning techniques, to further enhance the accuracy and robustness of network anomaly detection systems.

References

[1] Ali Bou Nassif, Manar Abu Talib, Qassim Nasir, and Fatima Mohamad Dakalbab. "Machine learning for anomaly detection: A systematic review." *IEEE Access* 9 (2021): 78658-78700.

[2] KDD'99 Data set, https://www.kaggle.com/datasets/galaxyh/kdd-cup-1999-data

[3] Song Wang, Juan Fernando Balarezo, Sithamparanathan Kandeepan, Akram Al-Hourani, Karina Gomez Chavez, and Benjamin Rubinstein. "Machine Learning in Network Anomaly Detection: A Survey." *IEEE Access*, vol. 9, pp. 78658-78700, Nov. 2021, doi: 10.1109/ACCESS.2021.3126834.