Network Anomaly Detection

Vasudevan G B[1], Muthu Krishnan K[2], Syed Muhammed Kareemullah Peeran[3], Dr. Shoba rani[4], Uma Maheshwaran[5]
[1, 2, 3] Dept of B.sc Information Science and Cyber Forensics
[4, 5] ProjectCo-Ordinator, Dept of B.sc Information Science and Cyber Forensics
[1, 2, 3, 4, 5] Dr. M.G.R. Educational and Research Institute Chennai, India.

Abstract- *Anomalies could be the threats to the network that have ever/never happened. To protect networks against malicious access is always challenging even though it has been studied for a long time. Due to the evolution of network in both new technologies and fast growth of connected devices, network attacks are getting versatile as well. Comparing to the traditional detection approaches, machine learning is a novel and flexible method to detect intrusions in the network, it is applicable to any network structure. In this paper, we introduce the challenges of anomaly detection in the traditional network, as well as in the next generation network, and review the implementation of machine learning in the anomaly detection under different network contexts. The procedure of each machine learning category is explained, as well as the methodologies and advantages are presented. The comparison of using different machine learning models is also summarised.*

I. INTRODUCTION

Network security has become increasingly critical these days, from the traditional computer network and cellular network to the next generation software defined network (SDN) and Internet of Things (IoT). The rapid growing network brings efficiency and convenience to our life, as well as the demand for high quality of service. Even though the network use case is getting more complex and a network device needs to process more data, users hope to get responses more quickly and show a lower tolerance to the service interruption. Firewalls, deep packet inspection (DPI) systems and intrusion detection systems (IDS) are the typical methods for anomaly detection, however, the cost to deploy these countermeasures and the complexity of system have to be considered [1], [2]. The security issue arises along with the evolution of network, the diversity of network services and applications provides hackers more opportunities to compromise the network than ever before. Especially, the working procedure in the next generation network, is quite different from the legacy network, current anomaly detection methods need upgrade to adapt to the change in these networks. For example, SDN decouples the control plane from the data plane, a centralised controller is usually responsible for the management of multiple data plane devices, besides a higher work load comparing to the control plane of a legacy router/switch, this architecture brings new challenges that the entire network is impacted if the controller is compromised. And the interaction between the control and data plane is no longer within the same hardware, it is mostly going through a network so that the administrator has to consider the security of data transmission, as a command from the controller towards the forwarding device could be modified during transfer. Similarly, data storage in the cloud network is quite different from the past, data has to pass through the network before being stored in a remote server. Also, IoT aims to connect everything from everywhere, the diversity of IoT applications increases the number of devices in the network, as well as complicates the network architecture. Due to the large number of connected devices and the high-speed broadband, anomaly detection requires to process big data over a complex network structure with a prompt reaction. This has become one of the biggest challenges to protect networks [3]–[5].

Machine learning (ML), as an analytical tool based on statistics, has been widely discussed and deployed in various areas. Its capability to make decisions after study and analysis relieves people from processing a huge amount of data, so that ML is normally used to investigate complicated scenarios. Furthermore, its response to abnormal behaviours is usually much quicker than human beings, which is an advantage in early detection. For known attacks, ML gains experience from existing records to understand their characteristics; while for unknown attacks, ML finds the outlier from the intrinsic patterns of data. ML can create diverse models with various algorithms, the way to work with these models also has a big difference. Based on the available dataset, the network operator could choose supervised learning to train a predictor when the size of labelled data is large, or a semi-supervised learning model when the number of labelled data is limited. Even if running the same model to detect the same type of attack, the outcome varies depending on the features that you prefer ML to consider [6], [7]. As a matter of fact, the most difficult step using ML is data preparation, from data collection to annotation, a high-quality dataset is vital to the prediction. Because the output of ML highly relies on the data from which algorithms learn the skill to distinguish normal operations from anomalous behaviours. Thus, in this paper, we introduce ML algorithms, as well as discuss the implementation of ML models in anomaly detection under different network contexts.

The contributions of this article are:

- It presents a comprehensive survey on the ML types.
- Detailed review and discussion of ML techniques in anomaly detection are introduced.
- Various network scenarios employing ML for anomaly detection are analysed.
- Characteristics and advantages of each ML model in anomaly detection are summarised.

II. RELATED WORK

EXISTING SYSTEM

The commonly used system for network anomaly detection is:

Rule – based system: It Utilize predefined rules to identify known patterns of normal and anomalous behavior. Effective for detecting specific types of attacks with well-defined signatures.

Signature-based Detection: Focuses on recognizing known patterns or signatures of known threats. Uses databases of predefined signatures to match against network traffic.

Packet Inspection and Filtering: Inspects individual packets for suspicious content or patterns. Allows for the filtering of packets that match predefined criteria.

Intrusion Detection Systems (IDS): Monitor and analyze network or system activities for signs of malicious behavior. Can be signature-based, anomaly-based, or a combination of both.

DISADVANTAGES IN EXISTING SYSTEM:

- Limited Adaptability
- High False Positive Rates
- Inability to Handle Complex Patterns
- Dependency on Updates
- Resource Intensive
- Lack of Context Awareness
- Static Detection Models
- Difficulty in Handling Large Datasets

PROPOSED SYSTEM:

- The system we proposed for network anomalies detection is based on machine learning models
- The machine learning models that are proposed in this system:
- Random forest classifier
- Decision tree

ADVANTAGES OF PROPOSED SYSTEM:

- Adaptability to changing patterns
- Detection of unknown anomalies
- Scalability
- Reduced False Positives

SCOPE:

The objective of using network anomaly detection is to identify unusual or suspicious patterns of behavior within a computer network. This is crucial for maintaining network security and integrity. Some specific objectives include:

Security Threat Detection: Anomaly detection helps in identifying potential security threats such as hacking attempts, malware infections, denial-of-service (DoS) attacks, or insider threats by detecting deviations from normal network behavior.

Early Warning System: By detecting anomalies in network traffic, systems can provide early warnings of potential security breaches or unusual activities before they escalate into major incidents.

Preventive Measures: Anomaly detection enables organizations to take proactive measures to prevent security breaches or mitigate their impact by identifying vulnerabilities or weak points in the network infrastructure.

Incident Response: When a security incident occurs, anomaly detection can help in incident response by providing insights into the nature and scope of the breach, aiding in containment, eradication, and recovery efforts.

Compliance Requirements: Many industries have regulatory requirements regarding network security and data protection. Anomaly detection helps organizations meet compliance standards by providing continuous monitoring and detection of security threats.

Resource Optimization: By identifying abnormal network behavior, organizations can optimize resource allocation, bandwidth usage, and network performance, leading to better overall efficiency.

Insight Generation: Anomaly detection can provide valuable insights into network usage patterns, trends, and abnormalities, which can be used for improving network design, troubleshooting, and capacity planning.

III. METHODOLOGY

The Network Anomalies Detection module using SVM Algorithm is a critical component of a cybersecurity system. It aims to identify and respond to irregular patterns or anomalies within the network, utilizing the Support Vector Machine (SVM) algorithm, a robust machine learning technique.

Data Collection:

Collects network data, including traffic patterns, packet details, and other relevant information.

Data may be sourced from network logs, packet captures, or real-time monitoring tools.

Data Pre-processing:

Raw network data undergoes pre-processing to extract essential features and prepare it for input into the SVM algorithm.

Involves data cleaning, normalization, and feature engineering to enhance the algorithm's effectiveness.

Training Phase:

Utilizes a labeled dataset comprising instances of both normal and anomalous network behavior.

The SVM algorithm learns to distinguish between normal and abnormal patterns during this phase.

Training involves optimizing the SVM model parameters for accurate anomaly detection.

Anomaly Detection:

The trained SVM model is applied to real-time or historical network data to identify anomalies.

Anomalies are detected based on deviations from the learned normal behavior.

SVM's ability to identify non-linear patterns makes it effective in detecting complex anomalies.

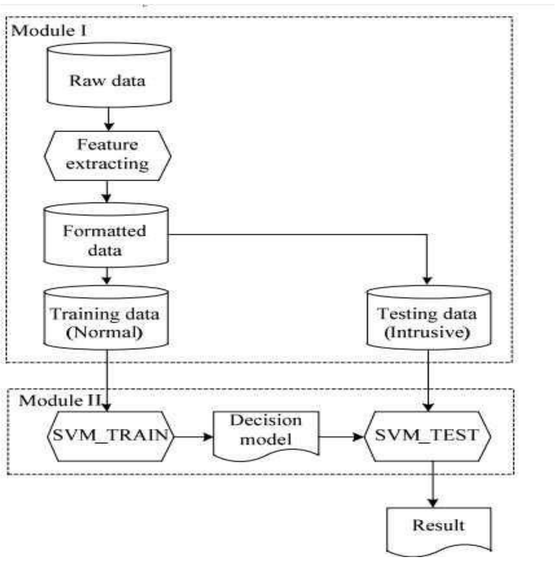
Expected Output:

The anomalies have been detected from the dataset and can be viewed that what type of attack is presented in the network traffic.

SVM BASED ANOMALY DETECTION:

ONE-CLASS SVM BASED ANOMALY DETECTION

In this section, we expound our one-class SVM based intrusion detection model. We first present the framework of the model, and then discuss how each constituent module works.



A. Framework of one-class SVM model

Figure 1. Framework of one-class SVM based model Our one-class SVM based intrusion detection model consists of the following two modules:

Module I: Feature extracting module.

Feature extracting is the necessary step to make the detection module work correctly. Our intrusion detection model integrates a feature extracting module mainly to extract useful features from the raw data and then generates manageable formatted data for the detection module.

Module II: One-class SVM module.

Working as the detection module, one-class SVM involves two procedures. The training procedure accepts the training data and generates a decision model. The testing procedure takes both the decision model and the testing data as inputs, and then produces the detection results.

The framework of one-class SVM based model is illustrated in Figure 1. The details of the two modules are presented in the following sections. 103

B. Feature extracting module

Almost no intrusion detection model can distinguish between intrusive connections and normal connections directly from original packets. They must be inputted with formatted data. Feature extracting is to obtain useful information from raw data and then format it, so that it can be interpreted by the detection module. There is no permanent standard to extract features. It may be better to extract features based on the actual network environment to find whether some attacks are hidden in connections. Extracting proper features helps the detection module to make more accurate predictions. In terms of network intrusions, some frequently-used features need paying attention to, such as the length (number of the seconds) of the connection, the type of the protocol, e.g. tcp, udp, etc., the number of data bytes transferred, the number of “root” accesses and so forth. In our one-class SVM based detection model, the feature extracting module takes the raw data as inputs, and then extracts expected features to form the formatted data.

Moreover, the feature extracting module is charged with dividing the formatted data into two divisions, the training data and the testing data. This process is fairly simple. The normal records comprise the training data and the rest (intrusive) records comprise the testing data. This relates to the detection mechanism of one-class SVM (detailed later).

C. One-class SVM module

$$\min_{\omega,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i$$

$$y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i$$

where ζ_i denotes the distance to the correct margin with $\zeta_i \geq 0, \ i = 1, \dots, n$

where C denotes a regularization parameter

were $w^T w = \|w^2\|$ denotes the normal vector

where $\phi(x_i)$ denotes the transformed input space vector

where *b* denotes a bias parameter

where y_i denotes the i-th target value

The following formula poses the optimization problem that is tackled by SVMs.

The objective is to classify as many data points correctly as possible by maximizing the margin from the *Support Vectors* to the hyperplane while minimizing the term

$$w^T w$$

In other words, the objective can also be explained as finding optimal **w** and **b** that most samples are predicted correctly. Mostly not all data points can be allocated perfectly so that a distance to the correct margin is represented by

$$\zeta_i$$

The *normal vector* creates a line that runs through the coordinate origin. The hyperplanes cut this straight line orthogonal at a distance

$$\frac{b}{\|w\|_2}$$

from the origin.

For the ideal case (Bishop, p.325 ff., 2006)

$$y_i(\omega^T \phi(x_i) + b)$$

would be ≥ 1 and followingly perfectly predicted. Having now data points with distance to their ideal position, lets us correct the ideal case of being ≥ 1 to

$$\geq 1 - \zeta_i$$

Simultaneously a penalty term is introduced in the minimization formula. C acts as a *regularization parameter* and controls how strong the penalty is regarding how many data points have been falsely assigned with a total distance of

$$\sum_{i=1}^{n} \zeta_i$$

- The dual problem

The optimization task can be referred to as a *dual problem*, trying to minimize the parameters, while maximizing the margin. To solve the dual problem, Lagrange multipliers are utilized (alpha≥0).

This leads to a Lagrangian function of (Bishop, p.325 ff., 2006):

$$L(w,b,a) = \frac{1}{2}\|w\|^2 - \sum_{n=1}^{N} a_n\{y_n(w^T\,\phi(x_n) + b) - 1\}$$

with $a = (\alpha_1, \dots., \alpha_N)^T$ representing the Lagrange multipliers

with b being the bias parameter

with w being the normal vector

where $\phi(x_n)$ denotes the transformed feature space

where y_i denotes the i-th target value

Utilizing the following two conditions (Bishop, p.325 ff., 2006):

$$w = \sum_{i=1}^{N} a_i y_i \phi(x_i)$$

$$0 = \sum_{i=1}^{N} a_i y_i$$

w and b can be eliminated from *L(w,b,a)*. This leads to the following Lagrangian function which is maximized as (Bishop, p.325 ff., 2006):

$$\tilde{L}(a) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N} a_n a_m y_n y_m\, k(x_n, x_m)$$

with the constraints $a_n \geq 0$ and $\sum_{n=1}^{N} a_n t_n = 0$

with $k(x_n, x_m)$ denoting the kernel function

Solving the optimization problem, new data points can be classified by using (Bishop, p.325 ff., 2006):

$$\sum_{n=1}^{N} a_n y_n\; k(x_n, x_m) + b$$

For the kernel function *k(x_n,x_m)* the previously explained kernel functions (sigmoid, linear, polynomial, rbf) can be filled in.

And that’s it! If you could follow the math, you understand now the principle behind a support vector machine. It’s easy to understand how to divide a cloud of data points into two classes, but how is it done for multiple classes? Let’s see how this work

Multiclass Classification using Support Vector Machine

In its most simple type SVM are applied on binary classification, dividing data points either in 1 or 0. For multiclass classification, the same principle is utilized. The multiclass problem is broken down to multiple binary classification cases, which is also called *one-vs-one*. In scikit-learn one-vs-one is not default and needs to be selected explicitly (as can be seen further down in the code). *One-vs-rest* is set as default. It basically divides the data points in class x and rest. Consecutively a certain class is distinguished from all other classes.

The number of classifiers necessary for *one-vs-one multiclass classification* can be retrieved with the following formula (with n being the number of classes):

$$\frac{n * (n-1)}{2}$$

In the one-vs-one approach, each classifier separates points of two different classes and comprising all one-vs-one classifiers leads to a multiclass classifier.

IV. CONCLUSION

Machine learning is trying to prove itself in multiple fields, among which anomaly detection is a feasible application that attracts lots of attentions. No matter what is the network scenario, people still have numerous options from ML models. Hence, we present a comprehensive review on the ML in network anomaly detection. From SL to RL, each category processes data in a different style, which leads to a large gap in the outcome. Supervised, unsupervised or semi-supervised learning model is picked based on the dataset on hand, the proportion of labelled data is a key factor in selecting a model. By contrast, RL is a totally different style, it allows the model to try all the state-action pairs so as to identify the best solution. In addition to the model selection, data quality is the most vital part for anomaly detection, it directly links to the prediction performance. Most of the solutions are validated by public datasets or in the simulation, it will be better to verify these models in the real network. And the resource consumption, such as training time and CPU utilisation, of the model is rarely discussed, this shall also be considered and studied to reflect the efficiency. In the future, we would like to explore more for the application of deep learning techniques in the next generation network, such as SDN and IoT.

REFERENCES

[1] https://www.researchgate.net/publication/377640205_Background_separation_network_for_video_anomaly_detection

[2] https://www.researchgate.net/publication/376110604_Enhanced_Memory_Adversarial_Network_for_Anomaly_Detection

[3] https://www.researchgate.net/publication/377596021_Distributed_and_explainable_GHSOM_for_anomaly_detection_in_sensor_networks

[4] https://www.researchgate.net/journal/Machine-Learning-1573-0565/publication/377596021_Distributed_and_explainable_GHSOM_for_anomaly_detection_in_sensor_networks/links/65af2f026c7ad06ab4232f5a/Distributed-and-explainable-GHSOM-for-anomaly-detection-in-sensor-networks.pdf?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIiwicHJldmlvdXNQYWdlIjoiX2RpcmVjdCJ9fQ

[5] https://circuitdigest.com/article/gridiq-energy-usage-optimisation-with-iot-assisted-machine-learning

[6] https://www.techtarget.com/search/query?q=Network+anomaly+detection

[7] https://www.link-labs.com/hs-searchresults?term=Network+anomaly+detection

[8] https://ieeexplore.ieee.org/document/9353774/

[9] https://link.springer.com/article/10.1007/s11235-018-0475-8

[10] https://www.usenix.org/legacy/event/sysml07/tech/full_papers/ahmed/ahmed.pdf?ref=driverlayer.com/web