International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

ELSEVIER
SSRN

IJIEMR Transactions, online available on 06th Feb 2023. Link :http://www.ijiemr.org/downloads.php?vol=Volume-12&issue=ISSUE-02

DOI: 10.48047/IJIEMR/V12/ISSUE 02/05

Title **Stochastic Study on Machine Learning and AI-based Intrusion Detection System**

Volume 12, Issue 02, Pages: 39-45

Paper Authors
Y Vijaya

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Stochastic Study on Machine Learning and AI-based Intrusion Detection System

Y Vijaya
Associate Professor & HOD
Department of IT, Vijaya Institute of Technology for Women
vijaya.vitw@gmail.com

Abstract - By scanning the network and server functions and informing the analyst if any suspicious activity is detected in network traffic, secure automated threat detection and prevention is a more effective method of reducing analyst workload. It constantly monitors the system and responds based on the threat environment. Depending on the phase, this response action varies. Suspicious activities are detected using artificial intelligence, which works alongside a network intrusion detection system as a virtual analyst to defend against the threat environment and take appropriate measures with the analyst's permission. Finally, packet analysis is performed to look for attack vectors and classify supervised and unsupervised data. When unsupervised data is decoded or converted to supervised data with the help of analyst feedback, the algorithm is automatically updated (Virtual Analyst Algorithm). As a result, the algorithm (with Active Learning Mechanism) improves in efficiency and power over time. As a result, it can defend against attacks that are similar or identical.

Key Words: Artificial Intelligence, Intrusion Detection System, Network Security, Machine Learning

Introduction

There are numerous threats on the internet, including malware and DDOS attacks. An intrusion detection system can guard against such attacks. When an intrusion is detected by an IDS system, an alert is generated. This intrusion detection system examines all network traffic. For large datacenters, this is a difficult task. A massive amount of data flows through the network of a data centre. Traditional intrusion detection systems are incapable of detecting all traffic.

One solution is to use IP flows to regenerate packet data. By utilising IP flows, an intrusion detection system can inspect all traffic. Furthermore, intrusion detection systems require extensive maintenance. Of course, this is conditional and comes at a high price. In addition, sensitive data is increasingly being stored digitally. All of these new services may have security flaws that expose private information such as passwords or other sensitive information. Security flaws are becoming increasingly important because they can cause so much damage. It is critical to safeguard a computer or network against malware and sensitive data leakage.

Given this, the ability to detect and prevent network system attacks becomes even more critical. This is accomplished by employing intrusion detection systems. Administrators can be notified of suspicious activity by an intrusion detection system. To function properly, most intrusion detection systems necessitate extensive manual maintenance. The purpose of this thesis is to determine whether an intrusion detection system can function properly

right out of the box. This is accomplished by employing machine learning algorithms. These are algorithms that can recognise and learn from input patterns. Machine learning algorithms appear to be promising solutions to the automatic intrusion detection problem. As a result, this thesis attempts to argue that an intrusion detection system may perform well out of the box. This is accomplished by employing machine learning algorithms. These are algorithms that are capable of learning from patterns and data. This appears to be well applicable to the intrusion detection problem; however, the algorithms may or may not work.

Attacks Classification

A useful classification is to distinguish between internal and external malicious behaviour. This improves human comprehension. The IDS is capable of working with a wide range of classifications. However, the IDS must notify an administrator of any detections. It is easier to distinguish between internal and external malicious behaviour. Each type of malicious behaviour is distinguished by different characteristics. Knowing these features allows you to fine-tune the IDS for better identification.

External Abnormal Behavior

System attacks are examples of external abnormal behaviour. There are many different types of attacks. Physical attacks, buffer overflows, DDoS attacks, brute-force attacks, vulnerability scans, and man-in-the-middle attacks are all possible.

Internal Abnormal Behavior

Malware is defined as abnormal internal behaviour. Malware can take many different forms. Malware is divided into four categories. Botnets, viruses, and other forms of malware do exist.

Trojan horses and worms. Malware are programmes that infect a system and perform a specific task. The task of the malware determines which category it belongs in.

Detection

A network intrusion detection system (NIDS) does nothing more than monitor the network. As a result, an NIDS will not be able to detect every attack. Only attacks that use the network are detectable. Flow-based intrusion detection systems can also only use flow data. This reduces the number of detectable attacks. DDOS, Vulnerability Scans, Worms, and Botnets are examples of attacks that flow-based network intrusion detection systems can detect.

Intrusion Detection System

An intrusion detection system is a system that detects intrusions within a system to determine whether or not it is under attack. IDS is an abbreviation for intrusion detection system. Attacks and anomalies are other terms for intrusions. This is accomplished by keeping track of network or system activity. The method of intrusion detection is one method of categorising IDSs.

Host-Based Intrusion Detection System

Host-based intrusion detection systems are intrusion detection systems that monitor the device on which they are installed or directly connected. They can monitor the system in a variety of ways, ranging from audit log monitoring to monitoring programme execution. HIDS can become constrained by audit logs because they rely so heavily on them. Another potential issue is the sheer volume of audit logs. Every monitored log must be parsed, which means that if the HIDS is installed on the host system, it can have a significant impact on its performance.

Another disadvantage is that any vulnerability that causes the audit files to be changed also compromises the integrity of the HIDS. If an audit file is changed, the HIDS cannot see or detect the change.

Network-Based Intrusion Detection System

Network-based intrusion detection systems monitor traffic from and to network devices and are installed at strategic points throughout a network. They operate on the same basis as

wiretapping. They "tap" into a network and monitor all communications. Although the intruder may attempt to limit his network activity, the risk is reduced. Furthermore, NIDS are more portable than HIDS. They monitor network traffic and are unaffected by the operating system on which they are installed. The system can use a variety of techniques to determine whether the data is malicious. There are two methods for analysing network data.In packet-based analysis, the entire packet, including the headers and payload, is used. A packet-based network intrusion detection system detects intrusions using packet-based analysis.

This type of analysis benefits from a large amount of data to work with. Each byte of the packet could be used to determine whether it is malicious or not. Flow-based analysis relies on aggregated data about network flows rather than individual packets. A flow-based network intrusion detection system detects intrusions using flow-based analysis. A flow is a single connection established between a host and another device.

Intrusion Prevention System

An intrusion prevention system, also known as an IPS/IDPS, is a detection and prevention system for intrusions. An IDS must be able to detect attacks at the exact moment they occur, which is preferable. In order to detect and prevent attacks in real time, an intrusion prevention system (IPS) must be capable of doing so. Preventive measures for network attacks could include closing the connection, blocking an IP address, or limiting data throughput.

The requirement that attacks be detected in real time can have a significant impact on the detection methods used. For example, an intrusion detection system (IDS) may issue an alert even if the IDS is unsure that what it is alerting on is an anomaly. An IPS must be certain before taking action. Otherwise, the IPS may perform actions that the company employing the IPS does not want the IPS to perform.

Detection

Intruders can be detected using a variety of methods. There are signature-based methods and anomaly-based methods. Signature-based methods compare so-called "signatures" to a signature database that already exists. A packet or flow record is divided into features that form a signature. If the signature of an incoming flow or packet matches one in the database, it is marked as malicious. Because they only attempt to match incoming signatures to known signatures in the database, signature-based methods have low computational and preprocessing overhead. Because it only compares signatures, it is simple to deploy within a network. It is not necessary for the system to learn how to analyse network traffic. Signature-based methods are extremely effective against known attacks. The signature database must be updated in order to detect new attacks. Attackers can also avoid detection using signature-based methods; all that is required is a minor modification to the "signature" to bypass the exact matching.

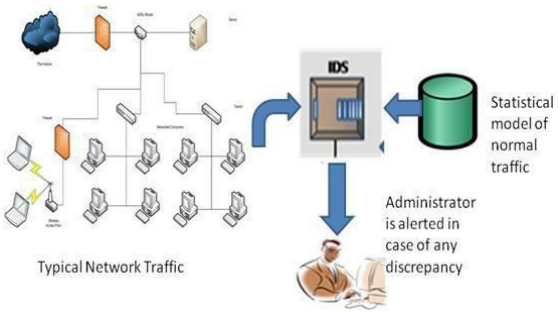

Fig-1: Signature Based IDS

Anomaly-based methods, also known as Behaviour-based methods, involve the IDS attempting to model network traffic behaviour. When an incoming packet deviates from this model, an alert is generated. Because they use a statistical model of normal behaviour, they should be able to detect all deviations from normal behaviour. As a result, new attacks are detected that deviate significantly from normal behaviour.

The system cannot be deployed into a network and expected to work because a network traffic model must be created. The system must gain knowledge of

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

network traffic behaviour. When training data contains errors, such as misclassifications, issues such as a high number of false positive alarms can arise. Machine learning algorithms can be used as an anomaly-based method. Machine learning techniques can learn from data and detect malicious data.

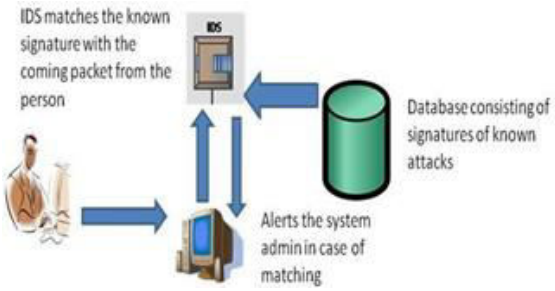

Fig-2: Anomaly Based IDS

Machine Learning

Machine learning is a subfield of computer science. It is a type of Artificial Intelligence in which programmes can learn and recognise patterns in data. Machine learning is the study of algorithms that can learn and predict from data. These are referred to as machine learning algorithms. A machine learning algorithm must learn before it can make predictions on data. The algorithm must be shown several examples of data as well as the correct predictions for these examples in order to learn. The algorithm may require thousands of examples to be presented to it.

Following data learning, the machine learning algorithm can be used to predict new data. Machine learning, for example, can be used to monitor the heart rates of hospital patients. During the learning phase, the machine learning algorithm is shown the patient's heart rate and the current time. Based on the current time, the machine learning algorithm can predict what the patient's heart rate should be after learning. This can be used to determine whether the patient's heart rate is normal by comparing the predicted and actual heart rates.

There are two kinds of machine learning algorithms. Unsupervised learning and supervised learning are the two types of learning. Data that has been labelled is used to train supervised learning. Unsupervised learning relies on unlabeled data. A training set is a collection of data that is used to train machine learning algorithms.

Evaluating ML for an IDS

When using a machine learning algorithm, the F-score can be used to evaluate performance. However, this is insufficient for intrusion detection systems. The F-score is based on the assumption that recall and precision are equally important. This is not always the case when evaluating intrusion detection systems.

When a Normal sample is misclassified as an Intrusion, this results in a false positive. A false negative occurs when a sample is actually an Intrusion but is classified as Normal. False negatives are undesirable because they indicate that an intrusion did not occur. Most intrusion detection systems, on the other hand, are used in a layered fashion. This means that if one layer misses an intrusion, another may.

The layered approach may also yield unexpected results. Perhaps the first layer tries to detect as many anomalies as it can (while maintaining a low recall) before passing the data indicating which anomalies were detected to subsequent layers. This method implies that having a low recall is not always a bad thing. The scoring for machine learning-based intrusion detection systems is determined by how the IDS will be used.

Using ML for IDS

Data must be processed before it can be used in a machine learning algorithm. This means that features must be selected. Some characteristics are obvious, while others must be discovered through trial and error.

Using all of a dataset's features does not guarantee the best IDS performance. It may increase the computational cost as well as the error rate of the system. This is because some features are redundant or ineffective for class differentiation.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Implementation

Technology Stack

The primary library used was Scikit-learn. Scikit-learn is a powerful machine learning library written in Python. It is built on top of the NumPy, SciPy, and matplotlib libraries. It is also open source and commercially usable under the BSD licence. learn. This library was chosen because it includes documentation as well as the most important algorithms. Scikit-learn also includes methods for visualising machine learning algorithms, such as a learning curve graph. These can be useful for evaluating the performance of machine learning algorithms. It also includes F-score calculation methods. This is beneficial because it reduces the likelihood of F-score calculation errors.

Program Execution

The implementation is carried out in stages. The program's elements are defined in a JSON configuration file. This is the data that will be used for learning, testing, and the machine learning algorithm, among other things. The programme can begin the training phase after reading the configuration file. During this phase, the specified algorithm is used and trained using the provided data. After that, the prediction phase begins. This phase collects all results and makes use of the prediction data. The structure and modules of the programme reflect these various phases.

Structure

The implementation is intended to be modular. The first is the machine learning module. This module contains all of the machine learning algorithms that are currently available. There is also a feature module. This module contains the classes that can be used to extract features from flows. A loader module includes all of the classes needed to load data from different datasets.

A training module contains all of the classes used for training. To send data to the machine learning algorithm, these classes use a loader class. They define which data will be used (for example, using only abnormal behaviour and leaving out the normal behaviour). Finally, there is a results module. This module receives the output of the machine learning algorithm and must log or visualise it.

Datasets

The implementation and algorithms were tested using a variety of datasets. Each dataset is used to test various aspects of machine learning algorithms. To begin, a subset of a dataset must be chosen for learning by machine learning algorithms. The method is then used to test the algorithm on another subset of the same dataset.

In the following step, the algorithms are tested using labelled real-world data. In the fourth step, the algorithms are tested using raw, unlabeled real-world data. This ensures that the algorithm performs well when tested on raw real-world data. To put the machine learning algorithms to the test, several datasets were used.

Scen.	Total Flows	Botnet Flows	Normal Flows	C&C Flows	Background Flows
1	2,824,636	39,933(1.41%)	30,387(1.07%)	1,026(0.03%)	2,753,290(97.47%)
2	1,808,122	18,839(1.04%)	9,120(0.5%)	2,102(0.11%)	1,778,061(98.33%)
3	4,710,638	26,759(0.56%)	116,887(2.48%)	63(0.001%)	4,566,929(96.94%)
4	1,121,076	1,719(0.15%)	25,268(2.25%)	49(0.004%)	1,094,040(97.58%)
5	129,832	695(0.53%)	4,679(3.6%)	206(1.15%)	124,252(95.7%)
6	558,919	4,431(0.79%)	7,494(1.34%)	199(0.03%)	546,795(97.83%)
7	114,077	37(0.03%)	1,677(1.47%)	26(0.02%)	112,337(98.47%)
8	2,954,230	5,052(0.17%)	72,822(2.46%)	1,074(2.4%)	2,875,282(97.32%)
9	2,753,884	179,880(6.5%)	43,340(1.57%)	5,099(0.18%)	2,525,565(91.7%)
10	1,309,791	106,315(8.11%)	15,847(1.2%)	37(0.002%)	1,187,592(90.67%)
11	107,251	8,161(7.6%)	2,718(2.53%)	3(0.002%)	96,369(89.85%)
12	325,471	2,143(0.65%)	7,628(2.34%)	25(0.007%)	315,675(96.99%)
13	1,925,149	38,791(2.01%)	31,939(1.65%)	1,202(0.06%)	1,853,217(96.26%)

Fig-3: Distribution of labels in CTU 13 Dataset.

The CTU-13 dataset was used for machine learning algorithm testing steps one through three. This dataset has been annotated. It encompasses botnet activity as well as normal and background traffic. The information was gathered at CTU University in the Czech Republic in 2011. It consists of thirteen distinct captures, each with its own botnet malware. Figure 4 shows how much data is contained in each capture. It is worth noting that the data was only collected for a few hours. Flows in the dataset contain additional information. Each capture contains only a small number of botnet samples. Background flows constitute the vast majority of flows.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Because it does not generate a lot of network traffic, this is typical botnet behaviour. Each flow is labelled with its exact location. This can range from Google Analytics to Google Webmail to a Windows update. The dataset's flows only include the standard information found in net flow. The abnormal behaviour in this dataset is internal abnormal behaviour. In the evaluation chapter, this dataset is referred to as the CTU dataset.

Id	Duration(hrs)	# Packets	#NetFlows	Size	Bot	#Bots
1	6.15	71,971,482	2,824,637	52GB	Neris	1
2	4.21	71,851,300	1,808,123	60GB	Neris	1
3	66.85	167,730,395	4,710,639	121GB	Rbot	1
4	4.21	62,089,135	1,121,077	53GB	Rbot	1
5	11.63	4,481,167	129,833	37.6GB	Virut	1
6	2.18	38,764,357	558,920	30GB	Menti	1
7	0.38	7,467,139	114,078	5.8GB	Sogou	1
8	19.5	155,207,799	2,954,231	123GB	Murlo	1
9	5.18	115,415,321	2,753,885	94GB	Neris	10
10	4.75	90,389,782	1,309,792	73GB	Rbot	10
11	0.26	6,337,202	107,252	5.2GB	Rbot	3
12	1.21	13,212,268	325,472	8.3GB	NSIS.ay	3
13	16.36	50,888,256	1,925,150	34GB	Virut	1

Fig-4: Amount of data and botnet type for each capture.

Algorithm Selection

Both supervised and unsupervised algorithms were used. These are the most widely used algorithms. Simpler and more general algorithms should be tested before employing more complex algorithms such as deep neural networks.

Unsupervised Learning

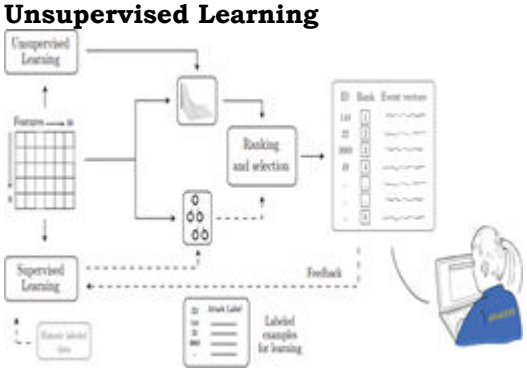

Fig-5: The structure of AI^2 system

K-Means clustering is used to determine whether clustering algorithms are capable of producing results. K-means is a simple clustering algorithm that already indicates whether or not a problem can be solved using clustering. There was, however, no method for determining whether the clusters formed by the K-means algorithm were correct.

One-class Support Vector machines are used for binary classification. They move very quickly. They were used to see if it was feasible to preprocess incoming data and see if a One-class Support Vector Machine detected abnormal behaviour before passing it to other algorithms.

Supervised learning

Support vector machines were used in the implementation. It is a well-known algorithm that can perform both linear and nonlinear classification, making it a promising candidate for implementation testing.

K-nearest Neighbors was the most promising algorithm. This algorithm is heavily used during implementation and testing. The fact that classification occurs based on different neighbours rather than attempting to create a classifier seemed to better fit the feature data.

Through the study of various machine learning algorithms, decision tree algorithms and Bayesian algorithms have also been discussed. They appeared to be less promising in terms of the intrusion detection problem. The distinction between a normal and abnormal flow is very subtle, and these algorithms appeared to make more mistakes. They are still used in the implementation to see if this assumption is correct.

To detect new attacks, their system attempts to combine the expertise of security experts with the speed and ability of machine learning. They use unsupervised machine learning in particular. Because labelled data is scarce and attacks are constantly evolving, they preferred unsupervised machine learning. They generate their own labels in the system and use these labels with a supervised learning algorithm. A big data processing system can extract properties of various entities from raw data. The outlier detection engine is a system for unsupervised learning. It employs features commonly found in large data processing systems. They use three methods: density, matrix decomposition, and replicator neural networks. The output of this unsupervised system is processed and displayed to a security analyst. The security analyst can verify or

refute the output. The feedback is fed into a supervised learning algorithm. Based on this feedback, the supervised learning algorithm trains a model to better predict whether a new event is normal or abnormal. The system improves as more feedback is provided.

Conclusion

The purpose of this thesis was to introduce machine learning algorithms and demonstrate their application in an intrusion detection system. Not all machine learning algorithms are created equal. The most difficult problem during the thesis was locating good labelled datasets that could be used to train machine learning algorithms. By using a good training dataset to train a machine learning algorithm, it is possible to create an intrusion detection system that performs well right out of the box. Much is dependent on the training dataset's quality. If there aren't enough samples of the various intrusions in the practise dataset, the machine learning algorithm will generate a lot of false positives and false negatives. All others were outperformed by K-Nearest Neighbors. It performs admirably in both the evaluation and real-world scenarios. It is critical to pay close attention to the value of k chosen and the distance metric used when using an algorithm like K-Nearest Neighbors. Unsupervised learning algorithms are not immediately effective. Before they can be used for intrusion detection, they require a significant amount of manual interference.

References

[1] AI2 : Training a big data machine to defend- Big Data Security on Cloud (Big Data Security), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), 2016 IEEE 2nd International Conference.

[2] Intrusion Detection Based On Artificial Intelligence Technique – International Journal of Computer Science Trends and Technology (IJCST) – Volume 2 Issue 4, July-Aug 2014

[3] Application of Artificial Intelligence in Network Intrusion Detection -A Succinct Review, World Applied Programming, Vol (2), No (3), March 2012. 158-166

[4] Intrusion Detection System (IDS): Anomaly Detection Using Outlier Detection Approach International Conference on Intelligent Computing, Communication & Convergence (ICCC-2014)

[5] D. Ten, S. Manickam, S. Ramadass, and H. A. Bazar, “Study on Advanced Visualization Tools In Network Monitoring Platform,” in Third UKSim European Symposium on Computer Modeling and Simulation, EMS ‘09’, Minden Penang, Malaysia, December 2009.

[6] L. Chang, W.L. Chan, J. Chang, P. Ting, M. Netrakanti, “A network status monitoring system using personal computer,” presented at IEEE Global Telecommunications Conference, August 2002.