Machine learning on network analysis

Susmita Rai - 908928

Contents

1 Project definition **2**
1.1 Introduction 2
1.2 Project aims 3

2 Related work **5**

3 Project management **7**
3.1 Methodology 7
3.2 Risk analysis 8

1 Project definition

1.1 Introduction

With the rapid expansion of the Internet, it has become an essential part of our lives with over half the population connected [1]. However, this results in an increasingly complex and fragile network. Many systems are left vulnerable, waiting to be exploited. By 2021, it is predicted that cost of cyber-attacks will reach $6 trillion [2]. The importance of a good and secure security system is far too crucial.

"Offensive cyber capabilities are developing more rapidly than our ability to deal with hostile incidents" [3]. Attacks are becoming smarter, polymorphic viruses and obscured malwares are passing through current systems. Over a third of organizations believe that the threats they are facing cannot be blocked by their anti-virus [2]. Due to our rapid growth, we have left many openings for an attack, one of them is through the network. Our need for constantly being connected is causing a major gap in security. In 2017, 8 different network attacks dominated the market [4].

1. Browser attacks - malicious users target vulnerable websites to infect, infecting new genuine users.
2. Brute force attack - attempting to guess your way through to the system.
3. Denial of service (DoS) or Distributed Denial of Service (DDoS) – flooding a service by creating many requests in order to slow or crash the system.
4. Worm attacks – self propagating program that spreads through local system through exploitable vulnerabilities.
5. Malware attacks – programs that can take many forms, however their purpose is always malicious.
6. Web attacks – exploiting vulnerabilities found in the website such as SQL injection.
7. Scan attacks – indirect attack to gain knowledge of any vulnerabilities that exist such as an open port.

8. Other attacks – attacks that were out of scope, such as physically attempting to steal device.

Fortunately, methods such as Intrusion detection system (IDS) exist to deter most of these attacks. IDS constantly scan the network for any anomalous activity in the network. Some are even capable of stopping the attack completely rather than just alerting the user.

However, IDS face many issues such as explaining what an anomaly is in the first place. Robustness and accuracy also come into question. How often does an IDS system report false negatives or how many different types of malwares can they detect?

By using machine learning it is possible to overcome these problems. Its ability of learning patterns and understanding different classifications can assist IDS. Todo add more here about machine learning. and what i want to do

I aim to create a system that can detect malwares on a network, and also test its robustness and accuracy rates. todo think more about this

1.2 Project aims

This project aims to create a system that can detect malware on a network by incorporating machine learning to enforce security on the network. There are multiple steps that are required with possible extension.

Data drives machine learning. Immense amounts of data have been collected, ready to be analysed however it is lacking for cyber security, especially for network analysis. Creating a network dataset that is unbiased and realistic proves to be a difficult challenge [5]. Unfortunately, the initial step would be finding a suitable dataset. Despite the issues with collecting network data, there are still many public resources available.

A note to consider is that I have already chosen a dataset which I will expand more under initial work. This is because the entire project depends on the dataset therefore, I had to choose a dataset before the project started.

There are several objectives that I would like to fulfill with this project.

1. Develop a system capable of classifying different malwares using machine learning.

   All IDS systems can detect anomalous activity. This would attempt to create an IDS system using machine learning techniques. The malwares would be anomalies that need to be detected.

2. Create synthetic malware attack patterns

   This depends heavily on the results of step 1. If the system can detect malwares, then it must also learn the pattern of what makes a malware. This would allow me to create synthetic malware attack patterns. Also, able to retrain classifier in step 1 to learn patterns of fake attacks creating a more secure IDS system.

   2.1. Replicate environment and create my own attacks

   Another possibility would also be to create actual malware attacks, not just synthetic. This would allow me to test the classifier on a live network rather than just on pre-collected data.

3. Test robustness of available IDS with synthetic data.

   If synthetic malware data is realistic enough, it would be able to fool IDS systems into thinking an attack has happened. What is the point of this?

   3.1. Create genetic adversarial networks to create even more realistic synthetic malware data.

   By creating GANNs, more realistic synthetic malware data can be generated. Thus, a loop of constantly testing robustness of IDS in step 3. Again, what is the point of this?

2 Related work

i dont fully understand whats supposed to be here

The hunt for a high true positive with low false positive IDS system has been sought for a while. Variety of different algorithms have been implemented in order to find one true solution.

Most attacks have a certain pattern that they follow resulting in two main approaches [6]. Anomaly detection and signature-based detection. Given a normal day-to-day activity, if a deviation occurs then that is recorded as an anomaly. That is the rule that anomaly detectors follow. Whereas for signature-based detection, it's based on the fact that an attack has a known pattern. If a known pattern is detected, there must have been an attack.

However, they both fall short. If a day-to-day activity is malicious, anomaly detectors will fail to flag it. Whereas for misuse detectors, the rise of polymorphic and obfuscated malwares is rendering it useless [6].

Nevertheless, many academics have implemented a range of different algorithms that attempts to detect malware.

One paper introduces a novel method for signature-based intrusion detection [7]. Creating an updating database that stores most frequent signatures to detect. The results show that the rate of false positives lowered, and malware detection speed increased. However, the database requires the administrator to choose whether signature is dangerous or not. Furthermore, other paper criticizes signature-based detection for its lack of ability to detect innovative malwares [8].

By utilizing data mining techniques such as Bayesian networks, to select features of importance, they were able to detect various attacks such as DOS with 100% accuracy. Their accuracy did falter for other attacks such as User2Root at 84%. [6].

Further research has been done on applying algorithms that are capable of learning and understanding features on a massive scope. Genetic algorithms mixed with traditional algorithms resulted in lower false positives and false negatives [8].

Unfortunately, even with many different implements, current IDS systems present different results for the same situation [9]. Judging the results can also be difficult. IDS systems that have high true positive and low false positive rate could lead to a false sense of security, especially if the attacks are over a large portion of traffic [9].

This shows that there's progress still to make when implementing an IDS system that has high positive rate with low false positives. One that is also able to handle the constant face-paced changes of evolving malwares.

3 Project management

3.1 Methodology

For this project, I have chosen to use Scrum agile methodology because of its iterative nature. Scrum framework follows one simple rule, that requirements can change, or otherwise known as requirements volatility [10]. Due to its flexibility, Scrum is ideally designed for a team of ten or fewer [11] and since I will be working alone, some of its fundamentals will be altered

Usually a daily Scrum takes place at the start of the day where the development team discuss the events of yesterday, however since I will be working alone, no daily Scrum will take place. Instead, a meeting with all other undergraduates under the same supervisor will take place every fortnight. This way everyone can discuss the progress of their respective projects.

Whereas for sprints, I will create a meeting with my supervisor every fortnight, which will be the length of one sprint. Instead of having a different meeting for planning, review and retrospective, it will all be merged into one. By doing this, I will still be able to receive some feedback and talk about issues that are/may cause problems.

Since I am the sole person working on this project, I will be responsible for creating a product and sprint backlog. Tasks will be broken down into separate categories, such as "todo" or "in-progress". Each task will also be assigned a priority ranging from "must haves" to "would be nice". I will be using GitHub project board [12] or some alternative that is similar in order to create the task board. Having an online task board allows easy access wherever I am.

This methodology also ties in well with the structure for machine learning projects. The brunt of the work for machine learning projects is always at the start which is data processing. In this step, data must be prepared and cleaned such that it can be fed into a model. This is then followed by model training and finally model testing and evaluation. Figure 1 displays the life cycle of machine learning projects [13]. Depending on how the data is processed, this results in different results for the model. The algorithms used for the model also highly influences the outcome. This ends up being

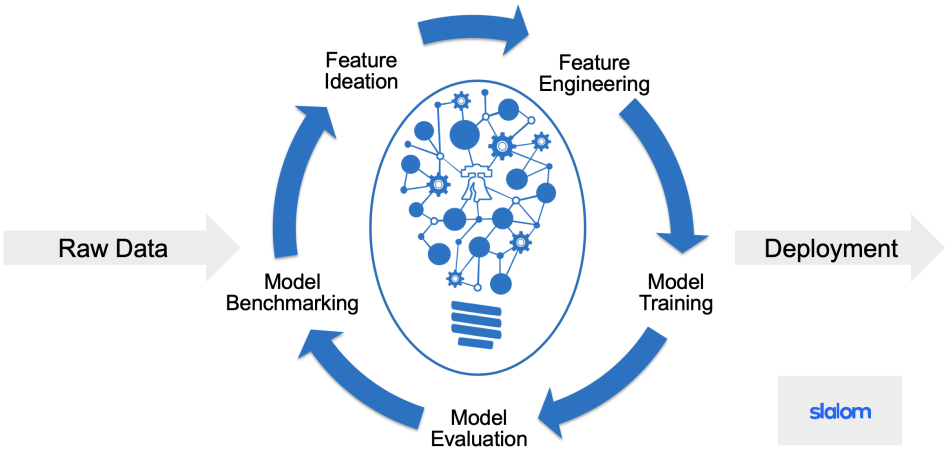

Figure 1: Machine learning life cycle

an iterative process since at each sprint, the goal would be to create the best model possible.

3.2 Risk analysis

1. Time management

   Due to my lack of productivity, unpredicted events or underestimation of the workload, the project may not be completed in time. However, because of my chosen software life cycle, I will be able to create a plan of tasks with their priority and effort levels recorded at the start of every sprint which should help guide the workflow. Also, the roadmap laid out by the Gantt chart contains a timeline which I will follow.

2. Hardware access

   This project deals with big data which requires extensive hardware. Lack of hardware can hinder progress as methods will have to be optimized which will require more time. For example, trying to just load the data will have to be split into various batches as there won't be enough RAM. Training models can also take significantly longer as a lot of processing power will be needed. Hopefully, this should not be an issue as the university has many different devices suited for this project.

3. Lack of domain knowledge

   This project deals with two separate issues, machine learning and security. For this project to succeed, it requires high domain knowledge in both fields which I may not have. However, by planning properly, tasks can be broken down into simpler components. Also, guidance is available from experts in the field.

   3.1. Dataset

   Every machine learning project requires a dataset and since it is far too time consuming to produce my own [14], I will have to select an existing one. I may end up choosing a wrong dataset that is not fit for this project. Before I make a concrete decision on which dataset to choose from, I will ask for feedback from experts in the field.

   3.2. Generalisable

   A lot of IDS systems that have been implemented using machine learning techniques seem to not be generalizable to other networks or system [14]. To avoid this, the model should not overfit the dataset, however if the model ends up underfitting, then the accuracy level will be poor. A balance between the two will be required. Creating my own hack and applying it to the model could also help learn patterns of malwares instead of learning what is the norm and any deviation from the norm is a malware. This could make it more generalizable.

   3.3. Malware – features of importance

   There are many algorithms that can understand what features are important and can find correlation between x and y easily. However, even if I created a list of features of importance, my lack of domain knowledge could make it harder to understand what the algorithm is trying to tell me. The results of the algorithm could be nonsense as well. There are many guides on known malwares, I will have to read up on them. If necessary, ask for expert advice as well.

   3.4. Machine learning - applying right methods

   Machine learning has a wide variety of different algorithms that could be implemented in this project. However, I may end up choosing the wrong ones. For example, when attempting to refine the model, I could end up implementing a strategy that does

not make any sense. It could even lead my model to learn incorrect information. Also, the first crucial step is to preprocess the data. Again, preprocessing it incorrectly could lead the model learning biased or incorrect information. For example, when the data contains a lot of NaNs, the model will not be able to handle this value. There are many ways of dealing with NaNs such as replacing it with a special value, using mean values or outright removing them. Whatever the method, it will affect the dataset which in turn will change the outcome of the model. A lot of background reading will be required and visualization of the dataset.

References

[1] Global internet usage in 2019. `https://wearesocial.com/blog/2019/01/digital-2019-global-internet-use-accelerates`. Accessed: 2019-10-18.

[2] Cybersecurity statistics of 2019. `https://www.varonis.com/blog/cybersecurity-statistics/`. Accessed: 2019-10-18.

[3] World Economic Forum. *The Global Risks Report 2018*, volume 13. 2018.

[4] Top 8 Network Attacks by Type in 2017. `https://www.calyptix.com/top-threats/top-8-network-attacks-type-2017/`. Accessed: 2019-10-18.

[5] J.O. Nehinbe. A critical evaluation of datasets for investigating IDSs and IPSs researches. 2011.

[6] Srilatha Chebrolu, Ajith Abraham, and Johnson P. Thomas. Feature deduction and ensemble design of intrusion detection systems. *Computers & Security*, 24(4), 2005.

[7] A. H. Almutairi and N. T. Abdelmajeed. Innovative signature based intrusion detection system: Parallel processing and minimized database. In *2017 International Conference on the Frontiers and Advances in Data Science (FADS)*, 2017.

[8] T. Mehmood and H. B. M. Rais. Machine learning algorithms in context of intrusion detection. In *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*, 2016.

[9] E. Guillen, D. Padilla, and Y. Colorado. Weaknesses and strengths analysis over network-based intrusion detection and prevention systems. In *2009 IEEE Latin-American Conference on Communications*, 2009.

[10] Joel Henry and Sallie Henry. Quantitative assessment of the software maintenance process and requirements volatility. In *Proceedings of the 1993 ACM Conference on Computer Science*, CSC '93. ACM, 1993.

[11] Ken Schwaber. *Agile Project Management With Scrum*. Microsoft Press, 2004.

[12] About project boards. https://help.github.com/en/articles/about-project-boards. Accessed: 2019-10-20.

[13] How and why to use agile for machine learning. https://medium.com/qash/how-and-why-to-use-agile-for-machine-learning-384b030e67b6. Accessed: 2019-10-20.

[14] S. K. Wagh and S. R. Kolhe. Effective intrusion detection system using semi-supervised learning. In *2014 International Conference on Data Mining and Intelligent Computing (ICDMIC)*, 2014.