CS229

Investigating the Importance of SMEs in InfoSec Machine Learning Projects

Napoleon C. Paxton
Department of Computer Science
Stanford University
ncpaxton@stanford.edu

Abstract

In the field of information security (InfoSec) there is a wealth of audit and log data which is clearly structured with a header and data corresponding to that header. Many machine learning projects treat each header as individual features. In manual InfoSec analysis, domain experts often use their intuition to combine data from different sources as well as different headers within the same data source to identify events on the network. This important step is often left out when conducting an analysis of InfoSec data in academia. In this project I will analyze two attacks with five machine learning algorithms, with the focus of pointing out how automatically generated data from security tools (generic features) are not sufficient for complicated attacks. The goal is to highlight the need for InfoSec subject matter expert (SME) input for feature engineering to improve modeling for complicated events on a network.

1 Introduction

When starting a supervised machine learning project, one of the main issues that arise is how to obtain features which adequately capture the information needed to train classifiers. In the field of InfoSec, tools used to capture data generally do a good job of structuring the results in clearly defined headers, which are then interpreted as stand-alone generic features in most machine learning (ML) based projects in this field. This is both an advantage and detriment for InfoSec projects in ML. It is an advantage, because we can often generate results without much additional work engineering features. It is a detriment, because the generic features are often narrow in their scope of capturing important information about the traffic. Traditional approaches to analyzing InfoSec data involves analysts using their favorite tools to generate insight and then identifying how output from those tools can be combined to illuminate key findings, such as what has happened, who is responsible, and what were they trying to do? Rarely are these analysts able to identify key insights based off one field of data alone. In this project, I will be highlighting the need for SMEs to use their expertise for the identification of engineered features instead of generic features that result from data being recorded from tools and placed into headers. I use the CIDS2017 Intrusion Detection Evaluation Dataset [1,2,3] in this project. This dataset contains 79 features and is fully labeled as benign, brute force, DoS/DDoS, Heartbleed, Web Attack, Infiltration, and Botnet. For the sake of brevity in this project, I limit my analysis to two attacks; a DDoS attack (acceptable for generic features), and an Intrusion attack (not acceptable for generic features). My methodology is to explain the complexity of the attacks and then show the results of running 5 ML algorithms on the datasets (Naïve Bayes for the baseline, Support Vector, Decision Tree, Logistic Regression, and Linear Discriminant Analysis).

CS229: Machine Learning, Winter 2019, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

I then explain why using generic features works perfectly for a simple model but does not capture all the information for more complicated ones. The reminder of this document is structured as follows: I will give a background of InfoSec data, I will then discuss related work in using InfoSec data for ML projects, next I will discuss my methodology which will include the criteria I use to select ML models, I will then present my results, and finally I will end with a discussion on my findings as well as future work.

2 Background of InfoSec Data

Over the last two decades many informative tools have been created to produce data that captures nearly every aspect of an event over the network. These tools are largely not aware of each other and work in silos. Experts are typically called upon to view the output of the tools and then decided how each result fits into the overall story of what has happend on the network. The next two subsets briefly describe some common tools and tool categories.

2.1 PCAP and flow data

Two of the most widely used formats to capture InfoSec data are packet capture (PCAP) and flow data. Data is recorded in the PCAP format by a well structured method that starts with a header to explain where the data is going or coming from. The reminder of the packet has all the information on the message as well as an abundance of metadata to describe the full message and the header. Flow data is composed of a group of packets that represent a communication. Some flow data only captures traffic in one direction (netflow), whereas some flow data captures traffic in both directions (argus). Tools that use these formats are able to extract the information contained in them and output them into files. In most of these file formats the name of the extracted information is represented as a column and the contents are placed in rows under those columns as new examples of PCAP packets or flow data records are observed. Since these files are constructed in this way, it is typical for machine learning practitioners to feed in the data as-is into classifiers for model training.

2.2 Signature and Behavioral Based Tools

When attempting to identify events on a network there are two approaches, signature based approaches or anomaly based approaches. In a signature based approach, in order to detect an event on a network we need to know identifying attributes of the specific event. In a behavioral based approach we need to know details of the environmental reactions which surround the event. Although the behavioral based approach has the potential to identify a wider variety of events, they are typically not used as much as signature based approaches due to issues with false positives. It has been found that recipients of protection from network threats value availability of their system to interruptions regardless of the reason. Because of this, false positives have historically been at or near the top of the list of metrics which need to be kept at a minimum. For this reason, we use false positives as our top metric and speed of analysis as our second metric in evaluating ML algorithms for this project.

3 Related work

As mentioned before, most InfoSec ML projects use data generated from security tools as input to their classifiers. In [4], the authors performed a survey of over forty ML research project. None of the projects discussed feature engineering of the data output of the security tools. Based on my search there were no meaningful projects exploring feature engineering of the data before inputting it into classifiers.

4 Methodology and Criteria

In this project I chose 5 machine learning algorithms to analyze the CICIDS 2017 data. I chose Naive Bayes as a baseline algorithm to start with due to its flexibility and well known ease of use across many ML projects. The remainder of the algorithms were chosen because they each have attributes which intuitively matched analysis for InfoSec data.

Component	Brute Force	DoS	Web Attack	Infiltration	DDoS
1	.5968987828	.7796069874	.6597628326	.6617593979	.7844068687
2	.2803475191	.0928683612	.1657824906	.1916784883	.0821037565
3	.0553456394	.0619252962	.1275157565	.1100498272	.0480309204
4	.0452758944	.0387505630	.0216924438	.0201152398	.0423106293
5	.0098241388	.0159198602	.0100553549	.0073983489	.0288421132

Table 1: Top 5 PCA Results for All Attacks

4.1 Criteria

As mentioned in the signature and behavioral based tools subsection of the introduction, the criteria used to select a model for the attacks were false positive rate and speed of analysis. It is important to note that the CICIDS 2017 dataset has split the data up based on the day and attack. The attacks happened over a 4 day period and on several days there were more than one attack. Each attack has it's own dataset. For instance, on Thursday there were two datasets created one for Infiltration and the other for Web based Attacks. In most projects these attacks would have been combined and therefore one model which performed best on multiple attacks would have been chosen. We discuss this further in the results section below.

5 Model Results

Here I give the details of running the 5 ML algorithms on all of the attacks. I begin by explaining the criteria chosen to select a model for the data and then I discuss the metrics.

5.1 Criteria

I chose the criteria for this project based on intuition of what is expected of a good tool for detecting InfoSec issues. The first metric is minimal false positives. In InfoSec it has traditionally been better to be attacked than to have availability of system or data affected by a false alarm of an attack. Secondly, time for analysis was chosen. For computer based attacks, each second or microsecond that passes can have a significant effect on reducing the effects of the attack and hopefully preventing future attacks of the same methods. The below models are judged on these two metrics.

5.2 Metrics

In this project I utilized all the features for the metric calculation, however I did utilize principal component analysis (PCA) to investigate the variance of each component. In table 1 the results of the top 5 components are listed. As shown by the table, nearly all the information needed to explain the model is captured in the first few components of each attack. This is due to the construction of the dataset.

Table two displays the details of the metrics results for choosing a model.

5.3 Choosing a Model

If we were able to generate a dataset which was as neatly separated as the CICIDS 2017 dataset, we would choose the following:

In the real world we would have datasets that contained a mixture of good and bad data along with multiple attacks. Based on our criteria of no false positives and minimal time for analysis, we would choose Decision Tree for our model if we had to pick one.

6 Discussion

The goal of this project was to point out how generic features are not appropriate for complicated attacks in InfoSec. In figure 2, we see a comparison of the AUC plots for DDoS and Infiltration attacks:

Model	Attack	Timing	F-1 Score	AUC
Decision Tree	Brute Force	6.2200184822	1.00	n/a
	DoS	4.2924901731	1.00	n/a
	Web Attack	7.3940353393	1.00	n/a
	Infiltration	12.857041835	1.00	0.93
	DDoS	2.9670190811	1.00	1.00
LDA	Brute Force	11.3669791221	0.97	n/a
	DoS	4.3929170192	0.96	n/a
	Web Attack	6.1625602245	0.98	n/a
	Infiltration	6.7440230846	1.00	0.64
	DDoS	5.5570180416	0.98	1.00
Logistic Regression	Brute Force	234.32226221	0.99	n/a
	DoS	21.183590193	0.99	n/a
	Web Attack	70.859363555	0.99	n/a
	Infiltration	18.889065980	1.00	1.00
	DDoS	19.271660566	1.00	1.00
Naive Bayes	Brute Force	1.1899881361	0.99	n/a
	DoS	.93432242106	0.95	n/a
	Web Attack	0.5040056705	0.99	n/a
	Infiltration	1.6880071163	0.99	0.99
	DDoS	1.0060040950	0.99	0.99
SVM	Brute Force	493.27308988	0.99	n/a
	DoS	145.22328239	0.99	n/a
	Web Attack	92.743904113	0.99	n/a
	Infiltration	18.279917001	1.00	1.00
	DDoS	142.47998952	1.00	1.00

Table 2: Model Metric Results

Attack	Model	False Positive Rate	Timing
Brute Force	Decision Tree	0	6.22
DoS	Decision Tree	0.01	4.29
Web Attack	Naive Bayes	0	0.50
Infiltration	Naive Bayes	0	1.68
DDoS	Decision Tree	0	2.96

Table 3: Top 5 PCA Results for All Attacks

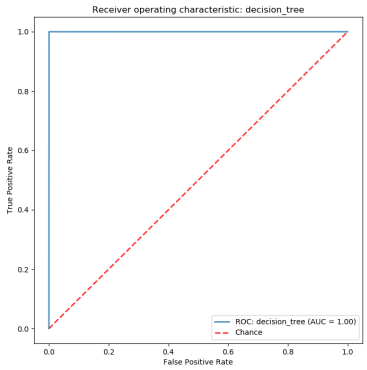

Figure 1: DDoS AUC

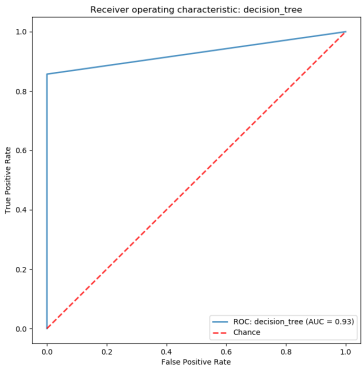

Figure 2: Infiltration AUC

Figure 3: Comparison of AUC Values

In the DDoS AUC we see that there were no false positives or false negatives in our model, but for the Infiltration attack the value is 0.93 which is very bad for this dataset due to its highly pre-processed nature. This can be explained intuitively by examining the nature of the attacks. DDoS attacks are known for their noisy behavior. By this I am referring to the abundance of straight forward signals which highlight the fact that an attack is taking place. Infiltration attacks are more stealthy in nature, meaning signals are harder to come by. InfoSec tools are very good at gathering artifacts or clues of events on a network. In some cases, such as DDoS attacks, these artifacts are enough to capture the appropriate amount of information. In cases such as the Infiltration attack, these artifacts need to be examined by someone who knows how to piece together the artifacts into a more informative feature to capture the meaning behind the occurrence of the artifact. This is the job for the SME.

7 Future

In the future I hope to extend this project by engineering features out of the general features identified in this dataset. I also plan on combining the data to give a more realistic feel to how data would be collected in a more realistic environment. I hope to show that the engineered features capture more information for models and perform better than generic features. I am especially hoping this is the case for the more complicated attacks.

8 Contributions

I completed this project alone. I look forward to hopefully continuing this work in a group in the near future.

9 Code

My code is being provided in a zip file.

References

[1] I. Sharafaldin, et al. "Toward Generating a New Intrusion Detection and Intrusion Traffic Characterization". ICISSP 2018

[2] A. Lashkari, et al. "An evaluation framework for intrusion detection dataset." ICISS 2016

[3] G. Creech and J. Hu. "Generation of a new ids test dataset: Time to retire the kdd collection". WCNC 2013

[4] G. Apruzzese, et al. "On the Effectiveness of Machine and Deep Learning for Cyber Security." 10th International Conference on Cyber Conflict 2018