JOURNAL OF INFORMATION SYSTEMS APPLIED RESEARCH

Volume 12, No. 1
April 2019
ISSN: 1946-1836

In this issue:

iscap
Information Systems & Computing Academic Professionals

The **Journal of Information Systems Applied Research** (JISAR) is a double-blind peer reviewed academic journal published by ISCAP, Information Systems and Computing Academic Professionals. Publishing frequency is three issues a year. The first date of publication was December 1, 2008.

JISAR is published online (http://jisar.org) in connection with CONISAR, the Conference on Information Systems Applied Research, which is also double-blind peer reviewed. Our sister publication, the Proceedings of CONISAR, features all papers, panels, workshops, and presentations from the conference. (http://conisar.org)

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the conference. At that point papers are divided into award papers (top 15%), other journal papers (top 30%), unsettled papers, and non-journal papers. The unsettled papers are subjected to a second round of blind peer review to establish whether they will be accepted to the journal or not. Those papers that are deemed of sufficient quality are accepted for publication in the JISAR journal. Currently the target acceptance rate for the journal is about 40%.

Questions should be addressed to the editor at editor@jisar.org or the publisher at publisher@jisar.org. Special thanks to members of AITP-EDSIG who perform the editorial and review processes for JISAR.

JOURNAL OF INFORMATION SYSTEMS APPLIED RESEARCH

Editors

2019 JISAR Editorial Board

Adversarial Machine Learning for Cyber Security

Michael J. De Lucia [a,b]
Michael.j.delucia2.civ@mail.mil

Chase Cotton [b]
ccotton@udel.edu

[a] U.S. Army Research Laboratory (ARL)
Aberdeen Proving Ground, MD 21005

[b] Electrical and Computing Engineering Department
University of Delaware
Newark, DE 19716

Abstract

The security of machine learning, also referred to as Adversarial Machine Learning (AML) has come to the forefront in machine learning and is not well understood in the application to the cyber security area. AML has been largely applied to image classification but has been limited in application to the cyber security area. One of the most fundamental components of machine learning, is the features. The disparate features of the cyber security area vary and are different than in image classification. To understand the features of the cyber security area, traffic classification is selected as a use case to focus on. Additionally, we present an example of cyber security AML of a network scanning classifier. A background on AML attack types, Adversarial Knowledge, and Image Classification features is given first. Next a discussion of the Cyber security traffic analysis features and AML of the cyber security area is given. We propose the disparate features of the cyber security area, augmented with ensemble learning could lead to a defense against AML. Future research is proposed for experimentation of AML with a subset of the cyber features discussed and the development of a defense against AML.

Keywords: Adversarial Machine Learning, Cyber Security, Traffic Analysis, Features, Machine Learning

1. INTRODUCTION

The security of machine learning, also referred to as Adversarial Machine Learning (AML) has come to the forefront in machine learning and is not well understood within a cyber security context. Machine Learning has become integrated into many different technologies to include cyber security (i.e. Intrusion Detection Systems (IDS), traffic analysis, malware and network scanning detection). Adversaries will attempt to circumvent and negatively affect the classification decisions, where machine learning has been employed for protection (Laskov & Lippmann, 2010).

AML has largely been applied to image classification and spam filtering with limited understanding within cyber security (Laskov & Lippmann, 2010). AML has also been focused on Deep Neural Networks (DNN) but has also been applied to traditional machine learning algorithms such as Support Vector Machines (SVM) (Papernot, McDaniel, Goodfellow, Jha, Celik, & Swami, 2017). Thus far there has been a limited knowledge of AML to cyber security. The specific

cyber security area that will be focused on will be AML of SVM (machine learning) traffic classification and analysis methods in addition to a network scanning detection scenario.
One of the fundamental components of the employment of machine learning methods to a specific technology area is feature engineering and representation. Features employed within machine learning based cyber security network detection classifier implementations vary greatly and are developed and engineered based on network traffic characteristics. The techniques that an adversary can use to perturb network traffic such that it is misclassified by the defender's IDS or traffic classification varies greatly depending on the machine learning approach and features implemented in the IDS or traffic classification.

We propose, a greater understanding of the importance of features and inclusion of multiple disparate features to improve the defense against AML for cyber security (traffic analysis). First, a background on the attack types, levels of adversarial knowledge, image classification features and AML will be given. Next, a discussion of features of the cyber security area and AML in cyber security, followed by an investigation and results of conducting AML on a network scanning detection classifier. Lastly a conclusion and discussion of future work will be presented.

2. BACKGROUND

AML Attack Types

In AML, there are two different types of attacks an adversary could perform; Evasion and Poisoning attacks (Muñoz-González, Biggio, Demontis, Paudice, Wongrassamee, Lupu, & Roli, 2017). An evasion attack occurs when an adversary perturbs a sample at test (detection) time to cause misclassification. A poisoning attack occurs when an attacker inserts mislabeled bad or perturbed data into the training samples. The focus of this paper will be on evasion attacks.

Adversarial Knowledge

There are varying levels of an adversary's knowledge of a system, which can be leveraged as attack models (Biggio, Corona, Maiorca, Nelson, Šrndić, Laskov, & Roli, 2013). The varying levels of knowledge include Perfect (Complete Knowledge), Limited, and Zero. Perfect level knowledge is defined as the adversary having knowledge of the feature space, type of classifier, and the trained model (Biggio et al., 2013). In the limited knowledge case, the adversary knows feature representation (features included) and the type of classifier, but not the trained model (Biggio et al., 2013). Lastly, zero knowledge is when the adversary does not know any of the details (features, type of classifier, or trained model) of the machine learning system. An adversary's knowledge levels of Perfect, Limited, and Zero are analogous respectively with the traditional cyber security terms of White-box, Grey-box, and Black-box. The terms White-box, Grey-box, and Black-box will be used throughout this work to refer to the adversary's level of knowledge of the machine learning classifier.

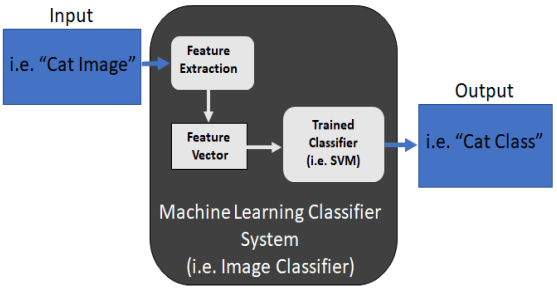

Figure 1- Machine Learning Classifier System View

Recall in the Black-box instance, an attacker has zero knowledge of the machine learning classifier. Therefore, an attacker may only have access to the input and output of the machine learning classifier. In Figure 1, it can be observed that the feature extraction and the classification decision occur within the machine learning classifier's system boundaries. Therefore, the features are unknown to the adversary. As Figure 1 depicts the Machine Learning Classifier System takes an input of the sample instance which is to be classified and the output is the class assigned. As shown in Figure 1 the adversary provides an input image of a cat to the Image Classifier and receives an output of the "Cat Class".

Many machine learning classifiers are open systems, allowing the adversary to view both the inputs (i.e. image) presented to the classifier and the resulting output class assigned (i.e. "Cat", "Not Cat"). However, there are cases where an adversary will have a partial view or no view of the input or output. An example, where an adversary will have no view of the input or output is a machine learning classifier which is executed in an isolated offline environment (not accessible). In a partial view, where only the input can be viewed, the adversary may need to infer the output class based on outside observations or knowledge. A further discussion of a partial view will be provided in a later section of AML for cyber security.

Image Classification AML

To understand transferability of AML from image classification to cyber security, we will give a brief background on the features within image classification. In image classification, an image is composed of a matrix of pixels and channels (e.g. 3 RGB channels), each representing the pixel (i.e. color) intensification (0-255). The pixels are directly extracted from an image as a feature. Additionally, the relationship between neighboring pixels can be extracted as features by using a combination of image gradient, edge detection, orientation, spatial cues, smoothing, and normalization (Zheng & Casari, 2018).

In image classification, AML is the perturbation of an image by adding noise to cause misclassification (Papernot, McDaniel, Jha, Fredrikson, Celik, & Swami, 2016). The perturbation of the image by an adversary must be applied meticulously to cause misclassification by the machine learning classifier, while still being correctly classified by the human eye (Papernot et al., 2016). AML in image classification has been primarily focused on DNN but has been demonstrated to transfer to traditional machine learning methods such as Support Vector Machines (SVM) (Papernot et al., 2017).

3. CYBER SECURITY FEATURES

A fundamental component of the machine learning development process is feature engineering. Feature engineering is defined as the process of transforming raw data into features to better represent the relationship between classes to improve machine learning performance (Susarla & Ozdemir, 2018). The features within cyber security are extracted differently compared to image classification. The features within the SVM based traffic analysis cyber security, are not always based solely on the bits within the network packet. They may be either based on each network packet or the network traffic flow.

There are many options for feature extraction directly from a network packet. Examples of features directly extracted from the network packet include the nested protocol headers or sub-fields or the packet payload (content). Inspection of the payload is often referred to as Deep Packet Inspection (DPI) or Payload based Classification (Kim, Claffy, Fomenkov, Barman, Faloutsos, & Lee, 2008).

An alternative option for feature extraction includes characteristics of a network traffic flow. A network traffic flow is often a group of network packets for a specific conversation between two endpoints. There are many characteristics of a network flow such as connection tuples (source and destination IP and Ports), inter-arrival times, sequence of packet sizes, Transport Layer Security (TLS) record sizes, offered TLS Cipher Suites, and the total bytes transferred in each direction.

Network Packet Features

An example which creates features from the packet payload is the Extremely Lightweight Intrusion Detection (ELiDe) System (Chang, Harang, & Payer, 2013). ELiDE builds an n-gram representation of the bits contained within the network packet payload to create the features for input into a binary linear classifier (Chang et al., 2013). While, the motivation for ELiDE was an Intrusion Detection System (IDS), it could also be used for fingerprinting of the payload for traffic analysis. Similarly, to image classification, an n-gram representation of the bytes contained in the network packet payload are directly extracted from the network packet as features.

However, this approach could be easily influenced by an adversary by encrypting the packet payload, thereby hiding any malicious activities. Therefore, the addition of encryption to the traffic payload protects and hides the malicious activities, resulting in an inability to perform DPI (Dainotti et al., 2012). The inability to perform DPI on an encrypted payload can be attributed to a different output produced each time since a new symmetric key is generated for each session established. For example, in the Transport Layer Security (TLS), which leverages encryption to protect communications, a handshake occurs first, during which a new symmetric key is generated and securely shared between client and server (Dierks & Rescorla, 2008).

As a result, this would allow an adversary to influence the machine learning classifier to cause a misclassification of malicious traffic as benign. This misclassification of an encrypted payload is caused by the fact of the payload n-gram representation features learned during training, not matching the extracted features at test (detection) time. Previously, encryption of the packet payload in of itself could have been an indicator of malicious activity or a signature for traffic classification. However, Internet traffic is increasingly becoming encrypted, as of 2016 approximately 30 percent of the top page search results on Google used HTTPS (SSL/TLS) (Meyers, 2016). According to Google Transparency Report on HTTPS encryption in the Web, 95% of traffic across Google's infrastructure is encrypted and 75% of Windows based Chrome

users browsed to HTTPS encrypted websites as of June 2018 (Google, 2018). The trend of Internet encrypted traffic is on the rise and will become widespread in the future.

Network Flow Features

An alternative traffic analysis mechanism is to use derived characteristics of the packet or network flow of traffic. In this instance traffic analysis is performed at a flow level which contains a sequence of packets which may be a bi-directional (client and server) or unidirectional (single sided) conversation. There exist several characteristics of a network flow such as the unique connection tuple (Source IP, Destination IP, Source Port, and Destination Port), inter-arrival packet times, unique TCP flags set, protocols used, non-conforming protocol use, frequency of communication, packet or protocol sizes, sequences of packet or protocol sizes exchanged, and domain names leveraged. While, these are a few examples of characteristics, the possibilities of different cyber security features are endless.

Appendix A presents even further cyber security feature examples, which demonstrate 52 features from (Muehlstein, Zion, Bahumi, Kirshenboim, Dubin, Dvir, & Pele, 2017), 19 features from (Anderson, Paul, & McGrew, 2016), 3 features from (Wright, Monrose, & Masson, 2006), and 2 features from (Herrmann, Wendolsky, & Federrath, 2009). The examples in Appendix A is merely a brief taxonomy of cyber security features from four different studies, but still displays a large number of features. Hence, the number of cyber security feature possibilities is massive.

Additional features for input to the machine learning classifier could be extracted and represented from these characteristics such as the mean and standard deviation could be taken over the timing and packet sizes over the traffic flows. Additionally, signatures of non-encrypted payloads carried by standard network methods can also be checked against signatures of known malicious payloads.

Another example is the use of data mining approaches such as the term frequency and inverse document frequency to represent the frequency of TLS record sizes within a conversation (De Lucia, 2018). In this case the characteristic is the term frequency of the TLS record sizes, which then forms the feature vector for each conversation. This single characteristic maps to a medium sized feature space of 32,000 unique possibilities, which results in sparse vectors since not all record sizes are present in every conversation. However, the sequence of TLS record sizes could be represented in a multitude of different ways to create features. For example, a possible alternative representation could be the total number of bytes, weighted average, and standard deviation sent in each direction. For an attacker to perturb their traffic flow to be misclassified as another type of traffic flow (malicious vs benign), they would need to modify the sequence of TLS record sizes being exchanged in each direction to match the pattern of another type of traffic.

Yet, another example is the attribution of TLS encrypted malware to a specific malware family (Anderson, Paul, & McGrew, 2016). Attribution using traffic analysis is performed using 19 different features such as identical TLS parameter use, sequence of packet lengths and times, network flow data, byte distribution, the TLS handshake list of offered cipher-suites, list of advertised extensions, and the public key length (Anderson et al., 2016). These features were also used to differentiate benign from malicious TLS clients (Anderson et al., 2016).

In this traffic analysis method, there are many features directly taken from the characteristics and some which are derived. Again, these characteristics could be represented in many ways to form the features which will be input into the machine learning classifier. For an attacker to cause misclassification of their traffic, they would need to modify many different characteristics. As an example, an adversary could modify the list of cipher-suites offered and extensions supported to match that of another traffic flow. However, the adversary may need to perturb several features to accurately cause misclassification.

4. AML CYBER SECURITY

In AML cyber security traffic, the adversary will perturb the malicious network application (i.e. malware, bot-net communication) traffic to appear as benign. For example, an adversary will perturb their Nmap network scanning traffic to appear as benign to a network scanning detector (machine learning classifier), resulting in a misclassification. However, just as in the image classification, there are constraints which are levied on the perturbation performed by the adversary.

There are many constraints within the cyber security area. Some example constraints include adherence to the respective networking (i.e.,

TCP, IP, TLS) protocol widely known standard documents (i.e. RFCs), implementation of offered services (i.e. TLS cipher suites offered in a Client Hello message), allowing the successful transmission of the message contents of a bot-net communication, and not negatively impacting the goal of malware contained within the network traffic. The constraints can change based on the objective and implementations chosen by the adversary (malware, bot-net traffic, or TLS client). For example, an adversary performing perturbation of the network traffic must be done within the bounds of the specific network protocol being leveraged (i.e., non-normal window sizes, improper TCP flags set).

Additionally, there is indirectly a human element for a constraint. Traffic analysis by a machine learning classifier may be also augmented with a human analyst. Therefore, the perturbations of the malicious traffic must be performed in a method which would not be noticeable by an experienced network analyst.

AML Perturbation

To cause misclassification, one of the fundamental components which an adversary must perturb is the features which are leveraged by the targeted SVM cyber security classifier. As discussed earlier, an adversary would need to perturb their malicious network traffic to mimic the features of a legitimate traffic flow, to hide their malicious activities. For example, a bot-net developer would need to perturb the bot-net traffic to look like either another bot-net (misattribution) or look like legitimate application traffic. We are assuming the adversary will leverage encryption, which implies that DPI is unusable, resulting in the need to use traffic analysis features. The next two examples are based on the network flow feature examples discussed in section 3.

Recall the first example network flow features discussed was the use of the TLS record sizes as a feature. The adversary would only need to perturb the single feature of the TLS record sizes. The TLS record sizes of the adversary's malicious traffic would need to be perturbed to mimic the sequence and distribution of TLS record sizes from a legitimate network traffic flow. However, this may have a cascading effect in producing a larger number of packets and increase of latency and inter-arrival times. For example, this increase could be attributed to a larger TLS record size resulting in longer processing times at the end nodes and transmission time of the message or malware to be sent. Much thought must be given by the adversary, as to the effects caused by the perturbation. However, this cascading effect could also be a benefit to the defense against AML. The attacker would also have the constraint of having to perturb the TLS record sizes, while still achieving a malicious goal.

Recall the second example network flow features discussed was the list of cipher suites offered, packet lengths, and timing. The adversary would need to perturb many more features of the malicious network traffic to mimic another legitimate network flow. For example, the attacker would need to perturb the list of cipher suites offered, the packet lengths, and the timing among many other features. The difficulty and cost, in terms of time, of mimicking another traffic flow, increases linearly as the number of disparate features increase. Each of the features is a disparate characteristic which must be manipulated to cause misclassification. Additional characteristic perturbations increase adversary implementation time to achieve misclassification. Additionally, perturbing a single feature may have a detrimental unintended effect on another feature.

As an example, if the two features are the TLS record sizes and the number of cipher suites offered, it will require disparate perturbations to the malicious traffic. An adversary would not only need to mimic the TLS record size sequences, but also the offered cipher suites. To mimic the cipher suites, the adversary would need to not only add it to the list, but also implement these cipher suites in the malicious client software. The additional implementation time to achieve these perturbations, indirectly increases the cost to the adversary.

Adversary Knowledge

Recall the Black-box, Grey-box, and White-box model for adversarial knowledge as discussed in section 2. All three of these models hold for the machine learning based cyber security of traffic analysis. However, there are some differences in the Black-box case, which will be expanded on. Recall in the Black-box case the adversary can only view the input and the output classes. However, in the cyber security traffic analysis, only the input is observed and known by the adversary and the output is not known or observed.

For example, let's assume the traffic analysis is being employed in a passive IDS in an enterprise environment. Traditionally, a passive IDS will raise and write alerts to the log file or notify an administrator for identified malicious network traffic. Therefore, the result is only known to the network administrator and not by the adversary.

To augment this example, let's now assume it is an active IDS within an enterprise environment. In this case, the IDS will act on the identified malicious network traffic, perhaps by blocking it. Again, there is no direct notification to the adversary of the output of the IDS machine learning classification. However, the adversary may be able to infer the classification output, since the adversary will notice their traffic being blocked, since the attack will fail or expected results are not received. The adversary can then infer that their network traffic was classified as malicious. Although, this observation of an attack failing or not receiving expected results may be indicative of some other problem that occurred, while the adversarial network traffic was in fact classified as benign.

In section 4 the discussion of perturbation of network traffic features is based on a Grey-box perspective, where the adversary is aware of the features which are being input into the traffic analysis machine learning classifier. Therefore, the adversary understands which network characteristics of their network flow must be perturbed to cause misclassification. However, the adversary may not know which subset of the features best represent another legitimate traffic class. Additionally, the adversary may not have an awareness of the representation of the network traffic characteristics. Lastly, the perturbation of certain features may cause an inadvertent change to another feature which may nullify the perturbation causing the adversary's network traffic to be correctly classified.

In the Black-box case of perturbation and AML, the features are unknown to the adversary. Therefore, the adversary is not aware of which features should be perturbed to mimic legitimate network traffic. In most cases the adversary would not be able to directly view the output of the traffic analysis machine learning classifier. However, the features could be vastly complex to be inferred even if the adversary were able to view the output. Therefore, in the black-box case, where features are unknown, the vulnerability to traffic misclassification is significantly reduced. As Appendix A, displays a large number of cyber security features from just a few different studies, the massive number of possibilities can be overwhelming to an adversary. Hence, the combination of as few as several different features themselves could be a defense against AML, since the adversary does not know which features to perturb.

5. AML CYBER SECURITY EXAMPLE

Background and Dataset

Earlier we discussed the ability of an adversary to conduct an AML attack in the context of a cyber security network detection classifier. We will now discuss our approach of AML conducted on a network scanning detector classifier and dataset consisting of network flow features originating from benign and malicious (Nmap network scanning) hosts. A notorious network scanning software tool leveraged by attackers is Nmap.

Normally attackers conduct network scanning in the initial phases of an attack to better understand the network and the ports open on a host. The attacker can then perform additional probes to uncover a specific software package and version listening on an open port. The discovery of a specific software package and version will assist the attacker in identifying a vulnerability to leverage in an attack.

The targeted SVM network scanning detector classifier was reconstructed based on the descriptions and features described in (Venkatesan, Sugrim, Izmailov, & Chiang, 2018). We implement the SVM classifier in the python programming language and scikit-learn. Additionally, the dataset leveraged was produced by the same authors (Venkatesan et al., 2018). The initial set of 11 different network flow features was reduced by feature selection to 3, consisting of the percentage of unsuccessful TCP connections, UDP, and ICMP connections (Venkatesan et al., 2018). The detector is trained using these 3 features which are extracted from network flows of benign (no scanning activity) and scanning (Nmap scanning) hosts.

Attacker Goal and Assumptions

The objective of the attacker is to hide (evade detection) the presence of the network scanning activity taking place on a network. Thus, the attacker will need to cause misclassification of a host's traffic flow as benign opposed to scanning. The attacker will need to perform several steps in order to cause a misclassification by the network scanning detector classifier.

The attack will be carried out from a grey-box perspective. In this scenario, the attacker does not have access to the trained target scanning classifier (including hyper parameters) and training dataset. We assume the attacker has knowledge of the specific 3 features being used in the target classifier and has access to a dataset of benign network flows (i.e. contains no scanning activity). The attacker may already have access

to the target network being monitored by the network scanning classifier and can passively collect benign network flows. A benign network flow dataset can also be built offline by an attacker.

Approach

The steps for an attacker to achieve the objective of misclassification (AML) of network scanning as benign traffic will be further described. A prerequisite for an attacker to perform AML is the ability to collect benign network flows and generate a network flows for network scanning activity. The resulting network flows are processed and analyzed to create a labeled (i.e. Benign and Scanning) dataset. Each network flow correlates to a sample in the dataset consisting of the 3 feature values (percentage of unsuccessful TCP connections, UDP, and ICMP connections) required by the network scanning detector classifier.

Using the newly created dataset, the attacker uses the nearest neighbor algorithm to identify the benign sample which is closest to each scanning sample and records the 3 feature values. These feature values are used as a baseline to compute the amount of TCP traffic which must be generated to cause misclassification. The additional TCP traffic will cause the 3 feature values of the scanning sample to decrease and mimic a benign sample. Lastly, based on the proceeding calculations, the attacker must generate additional TCP traffic on the actual host during scanning activities to cause misclassification by the target network scanning classifier.

Results

Experimentation was conducted using the network scanning detector classifier and AML method previously discussed. The dataset was split into 80% and 20% for training and testing respectively. The test dataset consisted of 40 scanning and 45 benign samples. Before introducing the AML attack, the baseline accuracy of the network scanning detector classifier was 100%.

	Accuracy
Baseline	100 %
AML	76 %

Table 2- Baseline vs AML accuracy

The collection of benign network flows was simulated by using benign samples in the test dataset. A total of 20 scanning samples to be perturbed were also selected from the test dataset. During the AML attack, 20 of the 40 scanning samples were perturbed using the method previously described. All 20 of the perturbed scanning samples were misclassified as benign. As a result, the classification accuracy of the network scanning detector reduced to 76% as seen in table 1.

It is expected that all perturbed samples would be misclassified, since the AML attack is mimicking benign sample feature values. Thereby rendering the network scanner detector ineffective. Therefore, a defense against this type of AML attack is required to continue successful detection of network scanning activity.

We propose the addition of features and ensemble techniques as a defense to this AML attack. The addition of features will make an attack incomprehensible as the number of characteristics for an attacker to perturb would grow. While, an ensemble would allow the combination of weak learners to form a stronger learner.

A proposed ensemble learner is composed of a network scanner detector as previously discussed and an anomaly detector for an abnormal amount of traffic originating from a host. Additionally, the introduction of a feature which has a direct relationship with existing features. An introduction of a feature in the anomaly detector of average amount of traffic for a host would cause an increase as an attacker conducts the AML attack. The generation of additional TCP traffic during the AML attack would cause an anomaly detection.

6. CONCLUSION

Summary

Adversarial Influence of Machine Learning (AML) has become the forefront of the security of machine learning but has largely been applied to image classification, which has been established for many years. It is imperative to understand the effects of AML transferability to cyber security in network traffic analysis. Features are a fundamental component of the machine learning classification process. Therefore, the features of the cyber security area must be well understood.

We believe features play a crucial role in the classifier and in developing resiliency. It is important to look at these vulnerabilities from a grey and black box perspective. Even though in the grey-box perspective an adversary will be aware of the features leveraged by the classifier, they will still need to know the subset of features which are representative of their traffic flow. Additionally, a larger number of features to

perturb will result in an increased cost to the adversary and in some cases may not be feasible. Lastly, from a black-box perspective, many disparate features themselves may be a sufficient defense against AML.

Future Work

We propose to conduct further exploration with several disparate features and an SVM for cyber security network detection classifier. Further exploration is expected to reveal the differences in the variety of fundamental feature distributions within a cyber security machine learning implementation in comparison to the image domain. As discussed earlier, the fundamental feature in image classification is the pixel intensity. An adversary need only perturb pixel values in an intelligent manner to achieve misclassification. Whereas in a cyber security machine learning classifier, the adversary would need to perturb disparate features of the network flow to achieve misclassification.

During our experimentation, we will perturb features of the network traffic flow, to achieve misclassification and evaluate the importance of features in an adversarial environment. The proposed experimentation will be evaluated using a representative cyber security network detection machine learning classifier. Lastly, we propose the development of defensive algorithms to protect against misclassification will include the use of ensemble machine learning methods which leverage a variety of disparate features for classification.

7. REFERENCES

Anderson, B., Paul, S., & McGrew, D. (2016). Deciphering Malware's use of TLS (without Decryption). Journal of Computer Virology and Hacking Techniques, 1-17.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., & Roli, F. (2013, September). Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 387-402). Springer, Berlin, Heidelberg.

Chang, R. J., Harang, R. E., & Payer, G. S. (2013). Extremely lightweight intrusion detection (ELIDe) (No. ARL-CR-0730). ARMY RESEARCH LAB ADELPHI MD COMPUTATIONAL AND INFORMATION SCIENCES DIRECTORATE.

Dainotti, A., Pescape, A., & Claffy, K. C. (2012). Issues and future directions in traffic classification. IEEE network, 26(1)

De Lucia, M. J., & Cotton, C. (2018, May). Identifying and detecting applications within TLS traffic. In Cyber Sensing 2018 (Vol. 10630, p. 106300U). International Society for Optics and Photonics.

De Lucia, M. J., & Cotton, C. (2018, Nov). Importance of Features in Adversarial Machine Learning for Cyber Security. In *2018 Proceedings of the Conference on Information Systems Applied Research*, Norfolk, VA. ISSN: 2167-1508.

Dierks, T., & Rescorla, E. (2008). The transport layer security (TLS) protocol version 1.2 (No. RFC 5246) <https://tools.ietf.org/html/rfc5246> (1 March 2018).

Google. "Transparency Report, "HTTPS Encryption on the Web." Retrieved July 13, 2018 from <https://transparencyreport.google.com/https/overview>

Herrmann, D., Wendolsky, R., & Federrath, H. (2009, November). Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier. In *Proceedings of the 2009 ACM workshop on Cloud computing security* (pp. 31-42). ACM.

Kim, H., Claffy, K. C., Fomenkov, M., Barman, D., Faloutsos, M., & Lee, K. (2008, December). Internet traffic classification demystified: myths, caveats, and the best practices. In *Proceedings of the 2008 ACM CoNEXT conference* (p. 11). ACM.

Laskov, P., & Lippmann, R. (2010). Machine learning in adversarial environments. *Machine Learning*, 81(2), pp. 115-119.

Meyers, P. J. "HTTPS Tops 30%: How Google Is Winning the Long War." Moz, (5 July 2016), Retrieved March 6, 2018 from <https://moz.com/blog/https tops-30-how-google-is-winning-the-long-war>

Muehlstein, J., Zion, Y., Bahumi, M., Kirshenboim, I., Dubin, R., Dvir, A., & Pele, O. (2017, January). Analyzing HTTPS encrypted traffic to identify user's operating system, browser and application. *In Consumer Communications & Networking Conference*

(CCNC), 2017 14th IEEE Annual (pp. 1-6). IEEE.

Muñoz-González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E. C., & Roli, F. (2017, November). Towards poisoning of deep learning algorithms with back-gradient optimization. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (pp. 27-38). ACM.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., & Swami, A. (2016, March). The limitations of deep learning in adversarial settings. In Security and Privacy (EuroS&P), 2016 IEEE European Symposium on (pp. 372-387). IEEE.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017, April). Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security* (pp. 506-519). ACM.

Susarla, D., & Ozdemir, S. Feature Engineering Made Easy. Packt Publishing, 2018.

Venkatesan, S., Sugrim, S., Izmailov, R., & Chiang, C.-Y. J. (2018). On Detecting Manifestation of Adversary Characteristics. In Proceedings of the MILCOM 2018, Los Angeles, CA (pp.431-437). IEEE.

Wright, C. V., Monrose, F., & Masson, G. M. (2006). On inferring application protocol behaviors in encrypted network traffic. *Journal of Machine Learning Research*, 7(Dec), 2745-2769.

Zheng, A., & Casari, A. Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists. O'Reilly, 2018.

Appendix A: Cyber Features Taxonomy

# Forward packets	Max throughput of backward peaks
# Forward total bytes	Backward min peak throughput
Min forward interarrival time difference	Backward STD peak throughput
Max forward interarrival time difference	Forward number of bursts
Mean forward interarrival time difference	Backward number of bursts
STD forward inter arrival time difference	Forward min peak throughput
Mean forward packets	Mean throughput of forward peaks
STD forward packets	Forward STD peak throughput
# Backward packets	Mean backward peak inter arrival time diff
# Backward total bytes	Minimum backward peak inter arrival time diff
Min backward interarrival time difference	Maximum backward peak inter arrival time diff
Max backward interarrival time difference	STD backward peak inter arrival time diff
Mean backward interarrival time difference	Mean forward peak inter arrival time diff
STD backward inter arrival time difference	Minimum forward peak inter arrival time diff
Mean backward packets	Maximum forward peak inter arrival time diff
STD backward packets	STD forward peak inter arrival time diff
Mean forward TTL value	# Keep alive packets
Minimum forward packet	TCP Maximum Segment Size
Minimum backward packet	Forward SSL Version
Maximum forward packet	Mean throughput of backward peaks
# Total packets	Forward peak MAX throughput
Minimum packet size	SSL session ID len
Maximum packet size	# SSL cipher methods
Mean packet size	# SSL extension count
Packet size variance	# SSL compression methods
TCP initial window size	TCP window scaling factor

(Muehlstein, Zion, Bahumi, Kirshenboim, Dubin, Dvir, & Pele, 2017)

Inbound bytes	Sequence of packet inter arrival times
Outbound bytes	Byte distribution of packet payload
Inbound packets	TLS version
Outbound packets	Order list of offered cipher suites
Source port	List of supported TLS extensions
Destination port	Selected cipher suite
Total duration of flow in seconds	Selected TLS extensions
Sequence of Packet lengths	Client public key length
Sequence of TLS record lengths	Sequence of TLS record times
Sequence of TLS record types	

(Anderson, Paul, & McGrew, 2016)

TCP packet size	Packet direction	Inter arrival time

(Wright, Monrose, & Masson, 2006)

IP packet size	Packet direction

(Herrmann, Wendolsky, & Federrath, 2009)