Black-box Adversarial Machine Learning Attack on Network Traffic Classification

Muhammad Usama*, Adnan Qayyum*, Junaid Qadir*, Ala Al-Fuqaha†
*Information Technology University, Punjab, Pakistan.
†Hamad Bin Khalifa University, Doha, Qatar
Email: *(muhammad.usama, adnan.qayyum, junaid.qadir)@itu.edu.pk, †aalfuqaha@hbku.edu.qa

***Abstract*—Deep machine learning techniques have shown promising results in network traffic classification, however, the robustness of these techniques under adversarial threats is still in question. Deep machine learning models are found vulnerable to small carefully crafted adversarial perturbations posing a major question on the performance of deep machine learning techniques. In this paper, we propose a black-box adversarial attack on network traffic classification. The proposed attack successfully evades deep machine learning-based classifiers which highlights the potential security threat of using deep machine learning techniques to realize autonomous networks.**

***Index Terms*—Adversarial machine learning, Black-box adversarial attack, Network traffic classification.**

I. Introduction

Network traffic classification is an important task in network engineering. It provides a method for monitoring, understanding, and quantifying network traffic. With the emergence of 5G, Internet of Things (IoT), and other related technologies, network traffic volume is expected to grow up to 3.3 Zettabyte (ZB) per year or 35 Gigabyte (GB) per capita per month by 2021 [5]. With this exponential growth in network traffic volume and inception of many data-hungry communication applications, network traffic classification becomes a very challenging problem for users and service providers. Classical network traffic classification techniques are based on port and payload inspection but these schemes have their shortcomings in terms of dealing with a large amount of data and different types of network traffic.

Recent advances in machine learning (ML) such as deep learning (DL) techniques have produced exceptional results in many application domains including: computer vision, natural language processing, speech recognition, and system control. DL is a branch of ML-techniques where a hierarchical structure of neural network layers is used to autonomously learn the features and then those learned features are used for classification or prediction. The success of ML/DL in multiple application domains has motivated the networking community to explore the potential benefits of these techniques for building an autonomous control for improving the performance of networking applications such as network traffic classification, anomaly and intrusion detection. In the last few years, many ML/DL based networking solutions have been proposed highlighting the applications, and challenges of using ML techniques in the networking domain [7], [9], [14], [30], [31].

Recently, ML/DL techniques were found to be vulnerable to carefully crafted perturbations in the test examples. These examples are known as adversarial ML examples. Adversarial examples force the ML/DL algorithm to malfunction and produce incorrect results. DL schemes especially deep neural networks (DNN) are function approximators that have an associated generalization error which makes them vulnerable to adversarial ML attacks. Adversarial ML attacks can be divided into two broader categories (based on the adversary's knowledge); namely, *white-box adversarial attacks* (i.e., perfect knowledge) and *black-box adversarial attacks* (i.e., real-world settings).

In this paper, we take network traffic classification as a functional proxy for ML/DL based autonomous networking applications and propose a black-box adversarial ML attack on network traffic classification. Although the focus of our paper is on network traffic classification, our findings are broadly applicable to similar settings that involve other applications of ML/DL to realize autonomous networks. The purpose of our proposed attack is to compromise the integrity of network traffic classification to shed light on the risks involved in utilizing ML/DL techniques in support of networking applications.

Our results indicate that the current state of the art ML/DL based network traffic classification algorithms do not provide substantial deterrence against adversarial ML attacks. Our experiments utilize the highly cited Tor-nonTor dataset provided by Habibi et al. [18] to perform the proposed black-box adversarial ML attack on Tor-nonTor traffic classifier to demonstrate that using current ML/DL techniques to realize autonomous networks can be a potential security risk.

Contributions: The contributions of this work are twofold:

- Propose and validate a black-box adversarial ML attack on network traffic classification (Tor-nonTor classification).
- To the best of our knowledge, this is the first black-box adversarial ML attack on network traffic classification to highlight that network traffic classifiers utilizing ML/DL techniques are very vulnerable to adversarial ML attacks.

The rest of the paper is organized as follows: In the next section, we review related research that focuses on network traffic classification (specifically, Tor non-Tor classification) and adversarial attacks on networking applications. Section III describes our research methodology; particularly, with reference to the dataset, the ML/DL model, threat model assumptions, and black-box attack procedure. In Section IV, we present our performance evaluations and discuss the outcomes of out experiments. Finally, Section V concludes our study

978-1-5386-7747-6/19/$31.00 ©2019 IEEE

and provides directions for future research extensions.

II. Related Work

A. ToR traffic classification

Tor [8] is a low latency anonymity preserving system that is based on overlay network that anonymizes the traffic of TCP based applications. It is also known as Onion Routing technique for traffic anonymization. In Tor, messages are encrypted and transmitted through distributed Onion routers, where each router uses a symmetric key to decrypt the messages and learn the routing details (i.e., Next Onion Router), the same process goes on in each router and this process obscures the actual transmitter. From the users' perspective, Tor provides security and privacy. Whereas, from service providers' perspective, detection and classification of the Tor traffic becomes a very difficult challenge [26].

Tor traffic classification is a key component of networking architecture. It is expected and experimentally validated that deep ML techniques outperform classical Tor traffic classification algorithms. Alsabah et al. [2] used a decision tree, Bayesian networks, and naive Bayes techniques for classifying Tor network traffic into two classes; namely, *interactive traffic* (e.g., browsing) and *bulk traffic* (e.g., torrent downloading). The purpose of their classification was to improve the Quality of Service (QoS) of the Tor network. In their two-way classification experiments, they managed to achieve 95% accuracy and 75% improvement in Tor responsiveness. Ling et al. [19] proposed TorWard, a malware detection and classification technique for Tor traffic which improves the Tor performance. He et al. [13] described a hidden Markov model based on Tor traffic classification schemes, their presented model classifies the Tor traffic into four categories; namely, P2P, FTP, IM, and web with 92% accuracy.

Unsupervised ML learning schemes such as gravitational clustering have been used for Tor traffic classification and the results were compared with classic clustering schemes in [25]. Hodo et al. [15] used Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs) for binary classification of the Tor-nonTor dataset and demonstrated 99% accuracy. Habibi et al. [18] used decision trees, KNN, and random forest techniques to perform binary classification on the Tor-nonTor dataset. Pescape et al. [24] proposed a traffic classification technique using multinomial naive Bayes and random forest techniques.

B. Adversarial ML Attacks

Since adversarial ML attacks have not yet been thoroughly explored in the networking domain, we first review their applications and effects in other domains. Deep ML techniques, especially DNNs, were demonstrated to produce the best classification results in many application domains but they also learn counter-intuitive and uninterpretable properties due to discontinuity in the learning process and generalization error [27]. These counter-intuitive properties can be exploited to form adversarial attacks that deteriorate the performance of DNN based classifiers.

Goodfellow et al. [11] proposed an adversarial perturbation generation method to fool DNN based classifiers called *fast gradient sign method* (FGSM). In FGSM, an adversarial perturbation is calculated by computing the gradient of the cost function with respect to the input itself. An extension of FGSM is proposed in [17] where FGSM was iteratively applied with a smaller step size to fool the DNN, this method is known as the *basic iterative method*. Papernot et al. [23] proposed a forward derivative based approach for crafting adversarial perturbations known as *Jacobian saliency map based attack* (JSMA). Carlini et al. [4] proposed three adversarial perturbation crafting methods for evading robust ML classifiers by exploiting three different distance matrices (L_1, L_2, and L_∞). Moosavi et al. [21] proposed *Deepfool* to evade ML classifiers, where adversarial perturbations were generated through the iterative linearization of the classifier. Transferability of adversarial ML examples such as logistic regression and SVM has been studied in [22], where it is highlighted that SVM is less prone to adversarial perturbation due to its training specificity and decision boundary learning process. More details about adversarial ML attacks are described in [1], [3], [20].

Adversarial attacks on network traffic classification have not yet been covered duly in literature, we will cover some general networking applications where basic adversarial ML research has been conducted. In our previous work, we have used FGSM, BIM, and JSMA based attacks to highlight the vulnerability of using ML in cognitive self-organizing networks under white-box settings (i.e., the adversary knows details about training data, training process, and model architecture) [29]. Corona et al. [6] highlighted challenges an research opportunities of adversarial attacks for network intrusion detection. Grosse et al. [12] used the JSMA attack to evade malware classification. Generative adversarial networks (GAN)-based black-box adversarial ML attacks on malware classification are presented in [16], where the condition of preserving the functional behavior was not ensured. In this paper, we propose an adversarial ML attack on network traffic classification considering black-box settings (i.e., the adversary does not have any knowledge about DNN training or architecture, the adversary can only query the deployed model for labels). In the next section, we will provide the details of adversarial ML attack on Tor traffic classification.

Table I
Notation Used

Symbol	Meaning
F	Function learned by DNN
x	Input traffic sample
y	Input traffic sample label
S	Substitute DNN model architecture
S_{approx}	Trained substitute DNN model
Q	Synthetic traffic samples
Y	Synthetic traffic labels
D	Synthetic dataset dictionary
MI	Mutual Information

III. Methodology

In this section, we describe the approach that we followed to design a black-box adversarial ML attack on deep ML-based Tor traffic classification, which is used as a proxy to represent other network functional areas. To the best of our knowledge, no standardized deep ML-based Tor traffic classification method has been proposed yet. Although as mentioned in Section II, neural networks have outperformed other ML/DL techniques for Tor traffic classification, we utilize both DNNs and SVM for Tor-traffic classification. We utilize the mutual information (MI) for crafting adversarial perturbations and substitute model training to perform the black-box adversarial attack. Before discussing the design of the deep ML classifier and the adversarial attack, we describe the threat model and related assumptions.

A. Threat Model

In this subsection, we delineate the threat model assumptions. Table I provides a summary of the notation used in our black-box attack procedure.

1) Adversary Knowledge: The proposed attack in this paper only considers black-box adversarial ML attack model, where the adversary has no knowledge of training data, number of layers in the model, training hyperparameters, type of layers, and training data. The adversary can only access the output of the deep ML model (i.e., DNN in this case) in a query-response manner. The adversary can send an input to the model and collect the label as a response. These query-response pairs are later used for crafting an adversarial attack. We assume that the adversary can only perform an adversarial attack during the test time. Other attacks, such as poisoning attacks, are not within the scope of this study.

2) Adversarial Goals: The goal of the adversary is to compromise the integrity and availability of the Tor traffic classifier by minimally altering the test examples. In adversarial attacks on computer vision and natural language processing applications, the fundamental restriction on adversarial examples is to preserve the visual representation or semantic meaning of the image or text, respectively. This restriction is replaced with a more difficult one in the context of networking where the adversary has to ensure that the applied adversarial perturbation does not affect the functional behavior of the network traffic.

B. Tor Classification Model

In this work, We use DNN and SVM for Tor traffic classification. SVM is a well known ML technique used for classification and regression. Whereas, DNNs are well-known for being capable of solving complex classification tasks by extracting hierarchical representation from their input. DNN consists of multiple layers of neurons (smallest computational unit), each layer's output is the input of the next layer. The non-linear activation function is used to ensure that each neuron also learns nonlinear information. The output layer of the DNN uses softmax as activation function to produce the classification probability vector.

We use the DNN and SVM classifiers for binary and multi-class classification, where binary classification is performed between Tor and non-Tor traffic and multi-class classification is performed between different Tor traffic classes. We used stochastic gradient descent with a batch size of 100 for training the classifier. We used a 70:30 ratio of the data for training and validation purposes. We achieved 96% and 93.54% accuracy for binary class classification using DNN and SVM, respectively. For multi-class classification, we achieved 90.60% and 80.60% accuracy using DNN and SVM, respectively. Classification accuracy of both models can be improved by carefully choosing hyperparameters for DNN training.

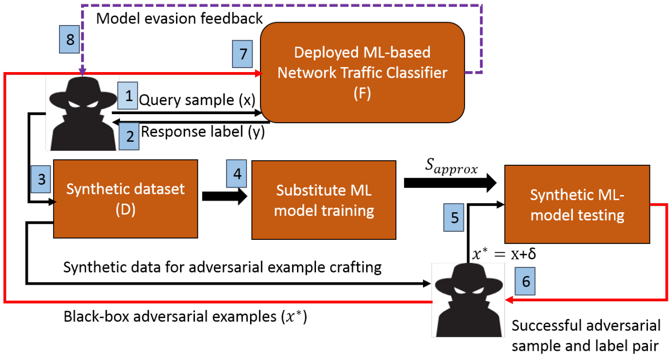

Model evasion feedback
8
7
Deployed ML-based Network Traffic Classifier (F)
1 Query sample (x)
2 Response label (y)
3
Synthetic dataset (D)
4
Substitute ML model training
S_{approx}
Synthetic ML-model testing
5
$x^* = x+\delta$
6
Synthetic data for adversarial example crafting
Black-box adversarial examples (x^*)
Successful adversarial sample and label pair

Figure 1. Steps for designing a black-box adversarial attack on network traffic classification processes through synthetic data generation, substitute model generation, and adversarial sample crafting.

C. Black-box Attack Procedure

In this subsection, we delineate the procedure used to perform the black-box adversarial ML attack on Tor traffic classification models. The proposed adversarial attack is performed in two steps; namely, *substitute model training* and *adversarial sample crafting*. Figure 1 provides a detailed description of the process used to perform a black-box adversarial attack on Tor traffic classification.

1) Substitute Model Training: According to the assumptions provided in III-A, the adversary can only query the deployed ML/DL model F with synthetic input data Q to get a label as a response Y. These query-response pairs are then used for training a substitute model architecture S. The goal in training S is to mimic the decision boundary of the deployed classifier F. This process is divided into two components; namely, *substitute model architecture design & synthetic dataset collection* and *substitute DNN training on synthetic dataset*.

Substitute model architecture design and synthetic dataset collection are very challenging tasks. Since the adversary has no information about F's architecture and training process, selection of appropriate architecture for S and training procedures are performed heuristically. In our experiments, we selected DNN as our substitute architecture. The adversarial attack proposed in this paper is also applicable to other ML/DL architectures with some modifications to the training process. The adversary can also train multiple ML/DL models to find the best-trained substitute model S_{approx}. Substitute model S is trained using synthetic data samples Q prepared by querying F for labels Y. We used a moderate number of queries to develop synthetic data for training S. Initially, the adversary sends queries to F from a set "Q" of synthetic traffic samples obtained by using Tor and a regular browser to get labels Y.

Each query along with its response label is stored as a synthetic data pair in synthetic dataset dictionary D.

Once we obtain a moderate amount of synthetic data—in our case 2644 query-response pairs—the substitute DNN is trained on this synthetic dataset. Binary cross entropy and categorical cross entropy loss functions are used in binary and multi-class classification. We utilized the stochastic gradient descent algorithm for training the substitute model S. The complete training process is described in Algorithm 1.

Algorithm 1 Substitute DNN Training

```
Input: $F$, $S$ and $Q$
Output: $S_{\text{approx}}$ and $D$
Initialize: $Y = \{\}$
for all $x \in Q$ do
    $y \leftarrow F(x)$
    $Y \leftarrow y$
end for
$D = \{Q, Y\}$
$S_{\text{approx}} = \min_{S} \quad \mathcal{L}(\mathcal{D}; \mathcal{S})$
return $S_{\text{approx}}$, $D$
```

D. Adversarial Sample Crafting

Once the substitute classifier S is trained, it is used to generate the adversarial attack. Since the adversary knows every detail about the trained substitute model S_{approx}, it is very easy to generate an adversarial perturbation for S_{approx}. To perturb S_{approx}, an adversary needs to find the most discriminant feature and to slightly perturb it. Formally speaking, an adversarial perturbation for network traffic is an input that when added to the actual input does not lose its functional behavior but gets classified in a different class. In our binary class classification case, a small perturbation in the Tor sample will force the classifier to classify Tor sample in the non-Tor category.

正式地说，网络流量的对抗性扰动是一种输入，当添加到实际输入中时，它不会失去其功能行为，但会被归类到不同的类别中。 在我们的二元类分类案例中，Tor 样本中的一个小扰动将迫使分类器将 Tor 样本分类到非 Tor 类别中

We use MI (mathematical notation: $I(X;Y)$) for most discriminative feature detection for adversarial sample crafting. The MI $I(X;Y)$ is defined as the measure of statistical dependence between two random variables. The MI between two random variables X and Y is given as

$$I(X;Y) = \sum_{X,Y} p(x,y) \log(p(x,y)/p(x)p(y)) \qquad (1)$$

To select the most discriminative feature from synthetic data "D", we calculate MI between each feature and label pair. The top "n" ("n" can vary between 1 to any moderate number of features while maintaining the functional behavior) features having the highest values of MI are selected as the most discriminant features. MI value of any feature also depicts its influence on the classification procedure. Once the most discriminative features are selected, they are perturbed sparsely using L_1 norm minimization (the perturbation is always kept less than 10^{-2}). The adversarial sample crafting algorithm is provided in Algorithm 2.

从合成数据“ D ”中选择最具辨别力的特征，

Once the adversarial examples crafted using the Algorithm 2 have successfully evaded S_{approx}, according to the adversarial ML transferability property [28], the adversarial examples evading the integrity of S_{approx} are highly likely to compromise the integrity of F. In our experiments, we evaluated the adversarial examples on F and the corresponding results are provided in Section IV, where for both binary and multi-class classification, the crafted black-box adversarial examples have successfully evaded the deployed ML-based network (Tor) traffic classification system.

For an adversarial attack to be practical in cognitive networking, it is important that the original packet's functionality is preserved even though the attacker is perturbing the packet with the intention of tricking the classifier. We assume that the perturbations can be reversed through a middlebox employed by the adversary or that the adversary uses the portions of packets for the perturbation that are otherwise unrelated to the packet's functionality (e.g., through extra padding or using unused control fields).

Algorithm 2 Adversarial Sample Crafting Algorithm

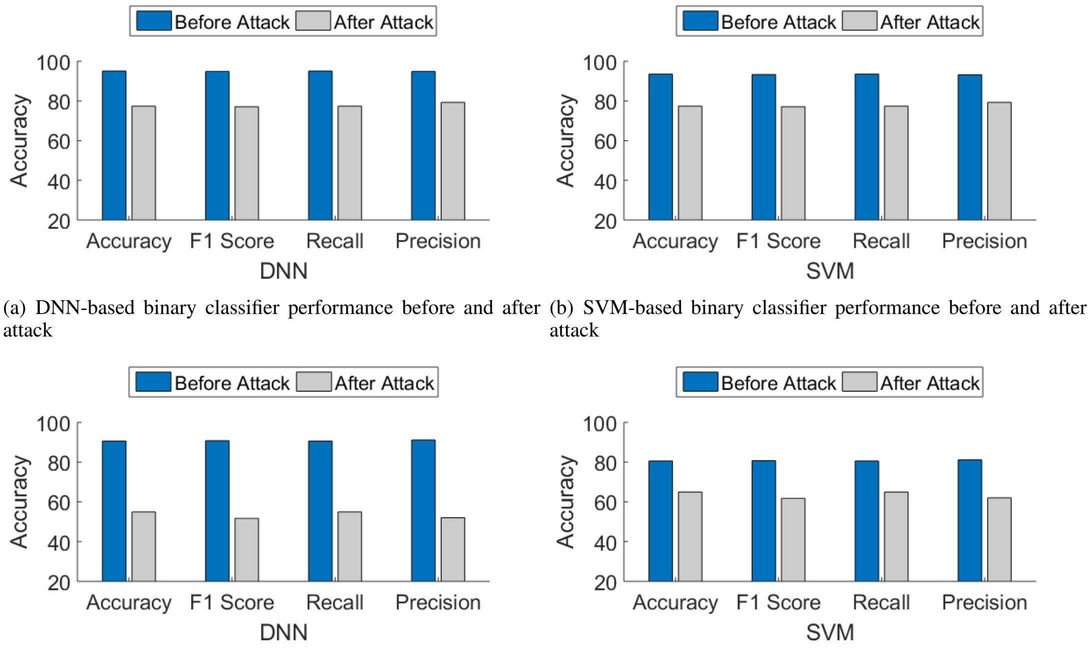
```
Input: $S_{\text{approx}}, D\{Q, Y\}$
Output: $x^*$
Initialize: $x^* = \{\}, p = \{\}, q = \{\}, i = \{\}$
while $\arg\max\{S(x^*) \neq y\}$ s.t $y \in Y,\ x \in Q$ do
    $MI \leftarrow$ Compute $\mathrm{I}(\mathrm{x}_i; y_i)$
    $[a, p] \leftarrow$ Select top$-n$ $MI$ values of target class
    $[b, q] \leftarrow$ Select top$-n$ $MI$ values of other class
    $p^* =$ compute $\arg\min_{p} \|p - q\|_1$
    $\delta = \vec{p} - \vec{p^*}$
    for l = 1: L do
        $x(a(l)) = x(a(l)) + \delta(l)$
    end for
    $x^* \leftarrow x$
end while
```

IV. Performance Evaluation

We conducted systematic experiments to evaluate the Tor traffic classifier against proposed our proposed black-box adversarial attack. Through our experiments, we want to affirm the hypothesis that the black-box adversarial example crafting algorithm proposed in this paper allows us to successfully fool the ML-based Tor traffic classification model. In our experiments, we consider DNN and SVM based binary class and multi-class Tor traffic classifiers. Drop in accuracy is considered as a measure of success for the proposed black-box adversarial sample crafting algorithm. Before discussing our experimental results, we provide a brief description of the dataset used in our experiments.

A. Dataset

We use the UNB-CIC Tor network traffic dataset [18] to validate the proposed black-box adversarial ML attack. The dataset consists of two classification categories: namely, a binary Tor-nonTor classification and multi-class Tor traffic classification. For the binary Tor-nonTor classification, Tor and Non-Tor traffic data samples are provided while for the multi-class classification, Tor traffic of 8 different applications



(a) DNN-based binary classifier performance before and after attack

(b) SVM-based binary classifier performance before and after attack

(c) DNN-based multi-class classifier performance before and after attack

(d) SVM-based multi-class classifier performance before and after attack

Figure 2. Performance of DNN and SVM based binary and multi-class Tor traffic classifiers clearly highlighting the drop in the classifer's accuracy due to the proposed black-box attack.

(namely, *browsing, chat, audio streaming, video-streaming, mail, file transfer, VOIP,* and *P2P*) is provided. The traffic samples were curated using Wireshark and Tcpdump. All necessary information from the traffic data has been extracted using ISCXFlowMeter [10]. More details about the dataset used in this study can be found in [18].

B. Results

We conducted systematic experiments to evaluate the performance of our adversarial attack on network traffic classification using Tor traffic classification as a proxy. In our experiments, we considered binary (Tor-nonTor) and multi-class (8 different Tor applications) classes for the network traffic. Only the top two most discriminant features were perturbed using L_1 norm minimization and the perturbation was limited to being less than 10^{-2}.

In the binary classification case, Tor vs. non-Tor, we achieved 96% and 93.54% classification accuracy on the legitimate samples using DNN and SVM classifiers, respectively. When the proposed black-box adversarial attack was applied, we observed a significant drop in classification performance. We created 2644 adversarial samples of Tor traffic. When these adversarial samples were subjected to the binary classifiers, we observed that the classification accuracy of the DNN-based classifier has dropped from 96% to 77%, and the SVM-based classifier's accuracy has dropped from $93.54\%$to 77.41%. Table II presents the performance of the proposed black-box adversarial sample crafting algorithm in the binary class classification task highlighting the number of successful "Tor" class adversarial samples. Figures 2(a) and 2(b) also depict the F1-score, recall, and precision performance of the binary classifiers before and after the adversarial attack.

In the multi-class classification case, we employed the proposed black-box attack to target the integrity of a single class (i.e., "Chat"). For adversarial sample crafting algorithm, we have considered "chat" vs "non-chat" classification. Once the adversarial sample is crafted for "chat class" it is subjected to a DNN and SVM based multi-class classifier. This process also proves the transferability of adversarial examples in networking domain. This experiment was performed to highlight that the proposed black-box attack can also perform targeted attacks. Legitimate "Chat" samples were classified with 96.3% and 96.4% accuracy by DNN and SVM based classifiers, respectively. After the adversarial attack, the classification accuracy of the same class has suffered a significant drop. For the DNN-based multi-class classifier, the accuracy dropped from 96.3% to 2% which is a drop of nearly 94% in performance confirming our hypothesis that DNN-based network traffic classifiers are very vulnerable to adversarial perturbations. For the SVM-based multi-class classifier, we observed a performance drop from 96.4% to 63.95% which is nearly a 33% drop in accuracy by only perturbing 2 traffic features. Table III provides the performance of proposed black-box adversarial sample crafting algorithm in multi-class classification highlighting the number of successful "Chat" class adversarial samples. Figures 2(C) and 2(d) also depict the F1-score, recall, and precision performance of the multi-class classifiers before and after the adversarial attack.

Our results highlight that the use of ML to realize network functions comes with potential adversarial attack threats.

Table II
Performance of proposed black-box adversarial sample crafting algorithm in binary class Tor traffic classification

ML techniques	Number of adversarial samples crafted	Successfully misclassified adversarial samples
DNN	2644	597
SVM	2644	598

Table III
Performance of proposed black-box adversarial sample crafting algorithm in multi-class classification ("Chat" class adversarial samples).

ML techniques	Number of adversarial samples crafted	Successfully misclassified adversarial samples
DNN	2644	2591
SVM	2644	959

These adversarial attacks can cause serious damage to the performance of networked applications. The design of more sophisticated black-box adversarial attacks on network traffic classification using advanced statistical schemes is left for future work. Similarly, the design of defense mechanisms against adversarial ML attacks to ensure robust ML-based network traffic classification is also left for future work.

V. Conclusions

In this paper, we proposed a method for performing a black-box adversarial ML attack on network traffic classification. We take the Tor traffic classification as a proxy for network traffic classification and demonstrated that deep neural network based network traffic classification schemes are very vulnerable to small carefully crafted perturbations in the test inputs. Our results also indicate that deep machine learning techniques, especially deep neural networks, do not provide any deterrence against adversarial perturbations and utilizing such techniques in networked applications can introduce new security risks to networking applications and infrastructure.

References

[1] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *arXiv preprint arXiv:1801.00553*, 2018.

[2] M. AlSabah, K. Bauer, and I. Goldberg, "Enhancing tor's performance using real-time traffic classification," in *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 2012, pp. 73–84.

[3] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

[4] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 39–57.

[5] Cisco, "The zettabyte era–trends and analysis," *Cisco visual networking white paper*, 2017.

[6] I. Corona, G. Giacinto, and F. Roli, "Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues," *Information Sciences*, vol. 239, pp. 201–225, 2013.

[7] L. Cui, S. Yang, F. Chen, Z. Ming, N. Lu, and J. Qin, "A survey on application of machine learning for internet of things," *International Journal of Machine Learning and Cybernetics*, pp. 1–19, 2018.

[8] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," Naval Research Lab Washington DC, Tech. Rep., 2004.

[9] Z. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2432–2455, 2017.

[10] G. D. Gil, A. H. Lashkari, M. Mamun, and A. A. Ghorbani, "Characterization of encrypted and vpn traffic using time-related features," in *Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP 2016)*, 2016, pp. 407–414.

[11] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples (2014)," *arXiv preprint arXiv:1412.6572*.

[12] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial perturbations against deep neural networks for malware classification," *arXiv preprint arXiv:1606.04435*, 2016.

[13] G. He, M. Yang, J. Luo, and X. Gu, "Inferring application type information from tor encrypted traffic," in *Advanced Cloud and Big Data (CBD), 2014 Second International Conference on*. IEEE, 2014, pp. 220–227.

[14] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and deep networks intrusion detection system: A taxonomy and survey," *arXiv preprint arXiv:1701.02145*, 2017.

[15] E. Hodo, X. Bellekens, E. Iorkyase, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Machine learning approach for detection of nontor traffic," *arXiv preprint arXiv:1708.08725*, 2017.

[16] W. Hu and Y. Tan, "Generating adversarial malware examples for black-box attacks based on gan," *arXiv preprint arXiv:1702.05983*, 2017.

[17] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[18] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy - Volume 1: ICISSP,*, INSTICC. SciTePress, 2017, pp. 253–262.

[19] Z. Ling, J. Luo, K. Wu, W. Yu, and X. Fu, "Torward: Discovery, blocking, and traceback of malicious traffic over tor," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2515–2530, 2015.

[20] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, and V. C. Leung, "A survey on security threats and defensive techniques of machine learning: a data driven view," *IEEE access*, vol. 6, pp. 12 103–12 117, 2018.

[21] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2574–2582.

[22] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 2016.

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

[24] A. Pescape, A. Montieri, G. Aceto, and D. Ciuonzo, "Anonymity services tor, i2p, jondonym: Classifying in the dark (web)," *IEEE Transactions on Dependable and Secure Computing*, 2018.

[25] Z. Rao, W. Niu, X. Zhang, and H. Li, "Tor anonymous traffic identification based on gravitational clustering," *Peer-to-Peer Networking and Applications*, pp. 1–10, 2017.

[26] S. Saleh, J. Qadir, and M. U. Ilyas, "Shedding light on the dark corners of the internet: A survey of tor research," *Journal of Network and Computer Applications*, vol. 114, pp. 1–28, 2018.

[27] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[28] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "The space of transferable adversarial examples," *arXiv preprint arXiv:1704.03453*, 2017.

[29] M. Usama, J. Qadir, and A. Al-Fuqaha, "Adversarial attacks on cognitive self-organizing networks: The challenge and the way forward."

[30] M. Usama, J. Qadir, A. Raza, H. Arif, K.-L. A. Yau, Y. Elkhatib, A. Hussain, and A. Al-Fuqaha, "Unsupervised machine learning for networking: Techniques, applications and research challenges," *arXiv preprint arXiv:1709.06599*, 2017.

[31] M. Wang, Y. Cui, X. Wang, S. Xiao, and J. Jiang, "Machine learning for networking: Workflow, advances and opportunities," *IEEE Network*, vol. 32, no. 2, pp. 92–99, 2018.