Feature Vulnerability and Robustness Assessment against Adversarial Machine Learning Attacks

Andrew McCarthy, Panagiotis Andriotis, Essam Ghadafi and Phil Legg
Computer Science Research Centre, University of the West of England, Bristol, UK
Email: Andrew6.McCarthy@uwe.ac.uk

***Abstract*—Whilst machine learning has been widely adopted for various domains, it is important to consider how such techniques may be susceptible to malicious users through adversarial attacks. Given a trained classifier, a malicious attack may attempt to craft a data observation whereby the data features purposefully trigger the classifier to yield incorrect responses. This has been observed in various image classification tasks, including falsifying road sign detection and facial recognition, which could have severe consequences in real-world deployment. In this work, we investigate how these attacks could impact on network traffic analysis, and how a system could perform misclassification of common network attacks such as DDoS attacks. Using the CICIDS2017 data, we examine how vulnerable the data features used for intrusion detection are to perturbation attacks using FGSM adversarial examples. As a result, our method provides a defensive approach for assessing feature robustness that seeks to balance between classification accuracy whilst minimising the attack surface of the feature space.**

***Index Terms*—adversarial learning, machine learning, network traffic analysis**

I. Introduction

Computerised systems are a feature of everyday life in all sectors, globally, including defence, energy, finance, health, and government. Adversaries such as criminals and advanced persistent threats deliberately explore network vulnerabilities, often gaining access to systems over computer networks and causing unwanted events. The European Union Agency for Cybersecurity [1] list common network attack scenarios including: Web-based attacks, Denial of Service (DoS), and Botnets. Public and private organisations must inhabit this threat landscape. The recent SolarWinds supply chain attack identified in December 2020 [2] [3] indicates our reliance on intrusion detection software, and the impact of successful attacks. Many Intrustion Detection Systems (IDS) incorporate Machine Learning (ML) to assist automated classification of malicious and benign traffic in a timely manner, to address issues of dealing with large, varied, continual streams of data and where a decision is required in minimal time.

An accurate ML model must be able to correctly classify malicious and benign traffic, whilst also minimising potential false positive and false negative results, which would result in misclassification. In addition to accuracy, ML performance can be assessed based on precision and recall. A model that predicts malicious traffic when it is mildly confident will likely have high recall but low precision, meaning some benign traffic will be flagged as malicious; alternatively a model that predicts malicious traffic only when it is certain will have low recall and high precision, meaning that some malicious packets will not be identified. Therefore, a trade-off between precision and recall can often exist and be exploited.

Szegedy *et al.* [4] explore how imperceptible perturbations of input values can result in significant differences in the output of ML classifiers such as Neural Networks. Neural network systems can therefore be susceptible to attack through carefully-crafted inputs, known as adversarial examples (AEs) [5]. AE are a form of evasion attack that relies on small perturbations to the original data, often undetectable to human observers. AEs can be algorithmically generated. Indeed there are a selection of algorithms that produce adversarial examples, including Fast Gradient Sign Method (FGSM) [5] and Jacobian Saliency Map Attack (JSMA), as well as available implementations of these attacks such as CleverHans [6].

Much of the existing work on adversarial learning is applied to computer vision tasks, such as image classification [7]–[9], and even well-trained models such as Microsoft's Common Objects in Context (COCO) [10] can be susceptible to adversarial attacks [11]. A fundamental issue is that images contain a significantly large amount of data (i.e., pixels) that would be used by the classifier - for example, a single 1080p colour image would have over 6 million input values for a classifier. Furthermore, subtle variations in such values would unlikely be noticeable to humans, due to colour perception issues [12]. A primary focus of our work is how adversarial attacks against ML classifiers translate across other domains, such as cyber security and network traffic analysis. In effect, an attacker could exploit the weaknesses of a modern ML-based Intrusion Detection System (IDS) so that an attack can evade detection and masquerade as benign activity.

In this work we study the security risks introduced through the use of ML-based detection systems. We explore the trade-off between ML accuracy and the classifier attack surface based on the permitted input features provided to the classifier. Using the CICIDS2017 dataset [13], we demonstrate a feature selection framework that can assess how robust the ML performance is both in terms of accuracy and vulnerability to attack. The main contributions of this work are:

- We consider countermeasures against algorithmically generated AEs, and in particular FGSM.
- We demonstrate an inverse relationship between number of features and robustness against AEs.
- We identify that applying systematic feature selection for model training improves model robustness against AE.

II. Related Work

A. Adversarial Attacks

Adversarial attacks can be classified as either white-box or black-box. The former require an attacker to have access to the target model's parameters whereas the latter does not. Black-box attacks may use a different model, or no model at all. Black-box strategies can employ the transferability of AEs, where AEs generated against one model can be successfully used to attack the target model [14]. Ayub *et al.* [11] built a multi-layer perceptron supervised ML model to detect and classify benign and malicious traffic using two distinct network-based IDS datasets (CICIDS2017 [13] and TRAbID [15]). They achieved outstandingly accurate classification results; however, via using Jacobian-based Saliency Map Attack (JSMA) [16] they reduce the accuracy by 22.52% and 29.87% on the CICIDS2017 and TRAbID datasets, respectively. Thus, demonstrating the ease of evading network defence, and the importance of countermeasures.

FGSM is designed to be fast rather than optimal. Therefore, generated AEs may be based on the minimal perturbations. For each feature the gradient of the loss function is used to determine whether increasing or decreasing the feature's intensity would minimize the loss function. All features are shifted simultaneously. Kurakin *et al.* [17] refined FGSM by taking multiple smaller steps.

Quereshi *et al.* [18] aim to understand the impact of AEs. They build a random neural network-based adversarial IDS, before training it on the NSL-KDD dataset. Subsequently they craft AEs using JSMA and the CleverHans python library. Their benchmark was highly accurate (95.6% benign and 96.61% DoS). Under JSMA their system accuracy fell to 47.58% for benign and a minimum of 28.10% for other attack classes. They note the poor results under JSMA were due to class imbalance in the benchmark data. Further they suggest that better feature extraction techniques could improve accuracy.

B. Architectural Defences

Lillicrap *et al.* [19] proposed a mechanism called feedback alignment that introduced a separate feedback path through random fixed synaptic weights. Gradient based attacks rely on the quality of the gradient to determine a possible attack; the use of feedback alignment which does not use weight transport thus increases robustness against AEs.

Amer and Maul [20] propose modifying the architecture of Convolutional Neural Networks (CNN) by adding a weight map layer. Their proposed layer can be easily integrated into existing CNNs. A weight map layer may be inserted between other CNN layers; thus increasing the network's robustness to both noise and gradient-based adversarial attacks, whilst maintaining accuracy.

Dropout is often used during training to improve test accuracy, particularly where over-fitting is seen due to limited training data; however, Wang *et al.* [21] propose defensive dropout at test time to harden deep neural networks against adversarial attacks. There is an inherent trade-off between defensive use of dropout and the test accuracy; however, a relatively small decrease in test accuracy can significantly reduce the attack success rate. Furthermore, larger perturbations to evade defensive dropout could be more readily recognized by humans.

C. Feature Selection

Hamed *et al.* [22] integrate feature selection into an IDS, aiming to select the most informative features to assist in the detection of "zero-day" attacks where few attack samples are available. They consider Recursive Feature Addition (RFA) and bigram technique (using two adjacent elements from a string) training their model on the ISCX 2012 dataset. Their objective was to find combinations of features that do not necessarily give good accuracy results independently, but work very well as part of a set of selected features.

Farahani [23] uses an IDS case study to propose a novel cross-correlation-based feature selection and compare it against the cuttlefish algorithm (CFA), and mutual information-based feature selection (MIFS). The selected features are used with four classifiers: support vector machines, naive bayes, decision tree, and K-nearest neighbour. They use four datasets: KDD Cup 99, NSL-KDD, AWID, and CICIDS2017. Their results show their proposed method has better accuracy, precision, recall, and F1-score when compared against CFA and MIFS.

Almomani [24] proposes a feature selection model for network IDSs utilising genetic algorithm, parallel swarm optimisation, and other bio-inspired algorithms to improve the performance of network IDSs. They use the UNSW-NB15 dataset and evaluate their selection model on support vector machine and J48 classifiers. They show that accuracy can be maintained with fewer features. The best results of their study for F-measure, accuracy and sensitivity were achieved using generated feature-sets of 30 and 13 features.

D. Visual Analytics

Legg *et al.* [25] studied visual analytics-based active learning as a means of assessing robustness in classifier performance with limited samples. Such visual analytic tools can also inform where genuine vulnerabilities in ML performance may be introduced due to weaknesses in the training data.

Yoo *et al.* [26] propose an interactive visual analytics tool to allow users to visually analyze the type, period, traffic, and frequency of attacks answering the challenge of handling and analyzing vast number of logs. They argue that the tool can be useful and show how a DoS attack was successfully identified and subsequently blocked.

E. Our Work

Most previous work aims at understanding the impact of AEs, or improving accuracy under normal conditions sometimes using feature selections. In this work we address a different problem of improving the robustness of ML models against adversarial attacks.

Traffic Type	Number of Samples
BENIGN	20,000
Bot	1,500
DDoS	1,500
DoS GoldenEye	1,500
DoS Hulk	1,500
DoS Slowhttptest	1,500
DoS slowloris	1,500
FTP-Patator	1,500
Heartbleed	11
Infiltration	36
PortScan	1,500
SSH-Patator	1,500
Web Attack Brute Force	1,500
Web Attack SQL Injection	21
Web Attack XSS	652

TABLE I: CICIDS2017: Traffic Types and Number of Samples

Our work utilises the relatively recent CICIDS2017 dataset. We use Principle Component Analysis (PCA), t-Distributed Stochastic Neighbourhood Embedding (t-SNE), Unified Manifold and Projection (UMAP), and parallel co-ordinate plots to examine the dataset. We focus on feature selection using RFE. Our focus is on improving robustness against AE attack. We use FGSM for its speed. We measure our model's robustness against AEs using accuracy. Further we consider perturbation size to determine whether feature selection could force more overt AEs, that could hopefully be more easily noticed by network operations engineers.

III. Method

We propose a generalizable approach for examining the robustness of features against adversarial attacks in the context of a ML classifier. We examine characteristics of the derived features from the data, and assess how these are manipulated by adversarial learning attacks. Based on these observations, we derive a feature selection approach that seeks to maintain classifier accuracy whilst maximising the amount of feature perturbation required to manipulate a classifier, hence improving robustness since the attack can no longer be performed in a subtle and discrete manner.

A. Dataset

We use the CICIDS2017 dataset [13]. One advantage of this dataset is that statistical time-related statistics have been calculated for both forward flows (client to server) and backward flows (server to client). Typical features in each flow are: Destination Port, Protocol, Flow Duration, Packet Statistics, Flow Bytes/s, Flow Packets/s, IAT Statistics, Flags, Header Length, Down/Up Ratio, Bulk Statistics, Subflow Statistics, Init Win bytes, act data pkt fwd, Active Statistics, and Idle Statistics. The flows are labelled with fifteen discrete classifications of traffic as shown in table I.

We focus on the DDoS class. To further understand the difference between benign and malicious data features, we use violin plots for comparative analysis to examine the distribution of each feature for each class as shown in figures 1 and 2. All features are "Normalized" (scaled between zero and one), and then separated by class, so that the scale factor for each feature is comparable for each of the violin plots shown.

Firstly, it can be seen that the distribution of features for the benign class in figure 1 is much greater than in the malicious case in figure 2. Furthermore, there is no visible separation between the two classes. The malicious class is essentially a subset within the distribution of the benign features. Closer inspection based on calculating the numerical difference between features suggests that inter-arrival time (IAT) may be a distinguishing feature between the two classes.

We train a model to distinguish between benign and DDoS traffic. In order to speed the training of the our model, we reduce the size of the DDoS dataset and consider only the first 50,000 samples. Through selecting this reduced number of samples, we create a more balanced dataset with 52% of samples labelled benign and 48% of samples labelled DDoS. This is an improvement over the unmodified dataset percentages (43% benign, 57% DDoS). We further clean the dataset to remove null and not applicable data.

B. Feature Analysis

Dimensionality reduction is a common first step when analysing datasets. For convenience the first one hundred samples of each class in the CICIDS2017 (DDoS) dataset were extracted and grouped as benign or malicious. We believe this sample is sufficiently indicative. We examine the data using dimensionality reduction methods such as PCA, t-SNE, and UMAP, as shown in figure 3. These three methods are commonly used for dimensionality reduction, allowing for visualization of the data on a 2D or 3D plot. PCA [27] is a well known algorithm that works by identifying the hyper-plane lying closest to the data, and projecting the data onto it. Thus, largely retaining the variation in the dataset. The t-SNE algorithm [28] finds clusters in the data, reducing dimensionality whilst aiming to keep similar instances together and dissimilar instances apart [28]. UMAP [29] is an effective algorithm for visualizing clusters of data points, usually providing faster and better visualizations than t-SNE.

In the PCA plot (figure 3a) we see malicious traffic gathered and occupying the same subspace as benign traffic, showing the complexity of the classification problem. More sophisticated methods such as t-SNE (figure 3b) and UMAP (figure 3c) begin to identify the clustering of the two classes in greater detail, however even so, it is noticeable that there is no clear single cluster associated with either class. This is an important observation as there is no single definition of what makes for benign or malicious traffic in respect to the features being studied within the dataset.

C. Parallel Co-ordinates

Therefore, we considered examining the raw features rather than the dimensionally reduced form. We identified IAT features as potentially good indicators of DDoS traffic, and chose to plot the features as parallel coordinates as seen in figure 4.

We select the subset of features that contain the text 'IAT', and use a parallel coordinates plot to examine the relationship

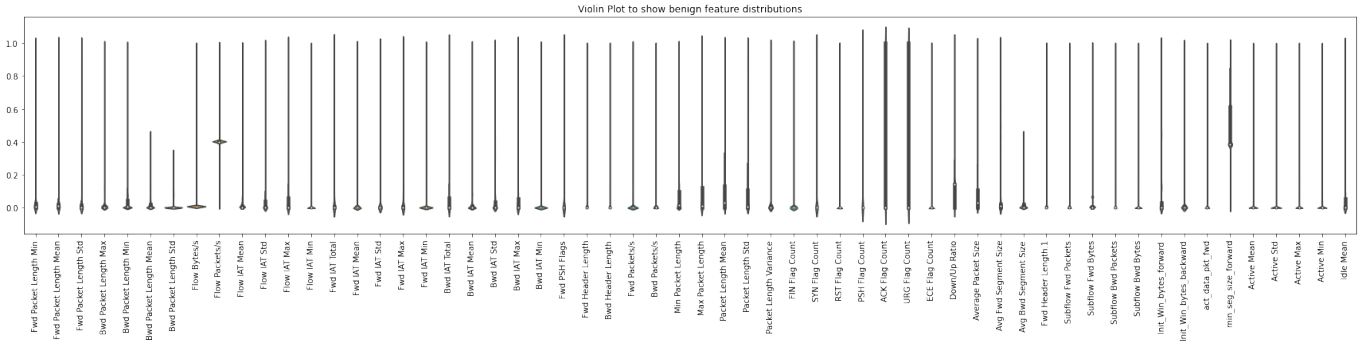


Fig. 1: A violin plot of the distribution of benign features. This violin plot shows the distribution of benign features.

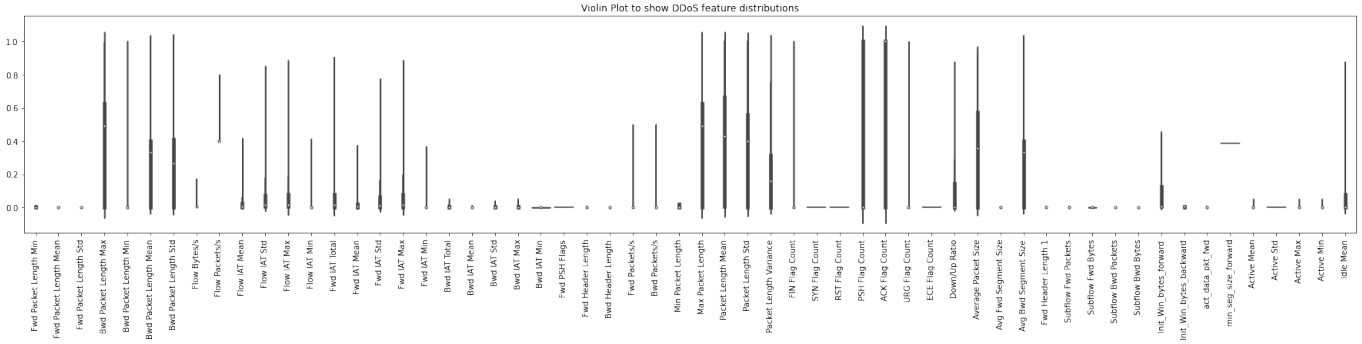
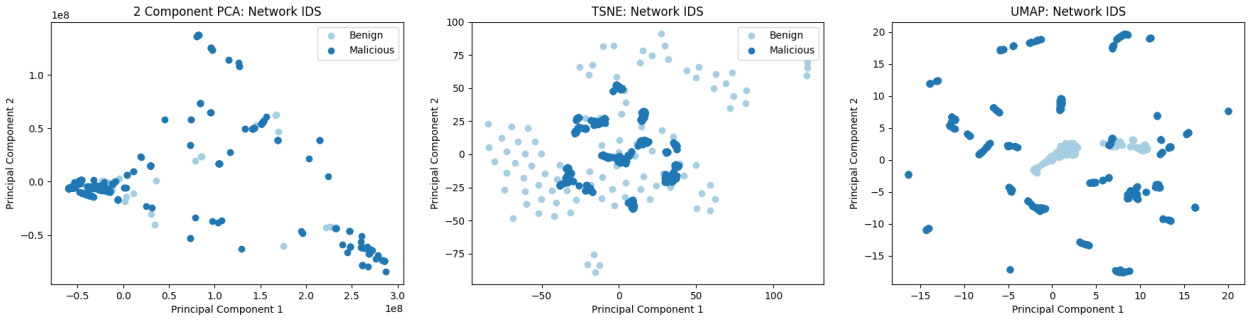


Fig. 2: A violin plot of the distribution of DDoS features. This violin plot shows the distribution of DDoS features.



(a) PCA (b) t-SNE (c) UMAP

Fig. 3: Dimensionality reduction methods to examine the clustering relationship between benign and malicious classes.

between the two classes further. In figure 4 each plotline represents an individual instance from the dataset. We believe this plot provides a clearer depiction between the two classes from the initial overview of the feature distribution.

Having identified some features as better indicators of DDoS traffic. We perform an initial study on accuracy and Mean-Squared Error (MSE). Further, we propose feature selection as a method to improve the classification accuracy against FGSM adversarial attacks. We consider Recursive Feature Elimination (RFE) [30], removing those features with the largest absolute difference under FGSM attack.

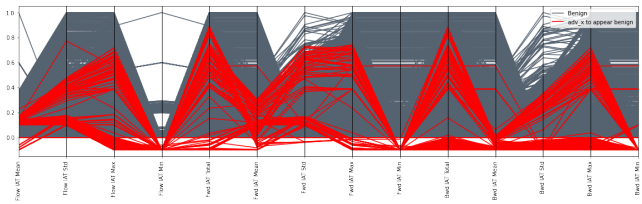
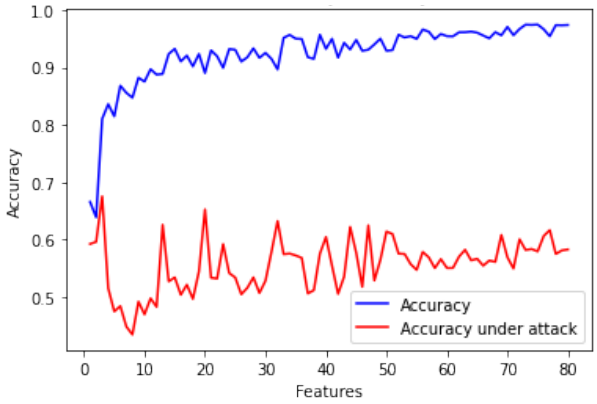


Fig. 4: A parallel coordinates plot of the distribution of benign features and DDoS affected by FGSM.

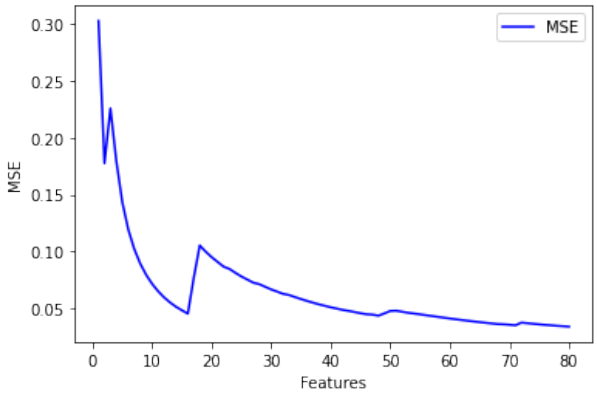
D. Training The Model

We adapted the CICIDS2017 dataset and trained a binary classifier to discriminate between benign and DDoS traffic.

We trained the model using shuffled stratified k-fold ($k = 5$), giving confidence of the validity of our results. We select a k value of 5 aiming to strike a balance between long run times and reduced sample bias. Each iteration was trained with a 80/20 training/test split, showing excellent results (100.00% +/- 0.00%). We consider this well trained model as a baseline; we compare our results against this baseline. We applied FGSM to generate AEs from the DDoS samples, again using k-fold ($k = 5$). Such AEs significantly reduce the accuracy of the classifier, yielding an accuracy of 58.57% (+/-) 15.03%. Using our trained model with x features we perform FGSM, and assess the perturbation of each of the features. We remove the feature with largest perturbation and retrain the classifier with $x-1$ features. We found the optimal solution, maximising accuracy.



(a) Accuracy



(b) MSE

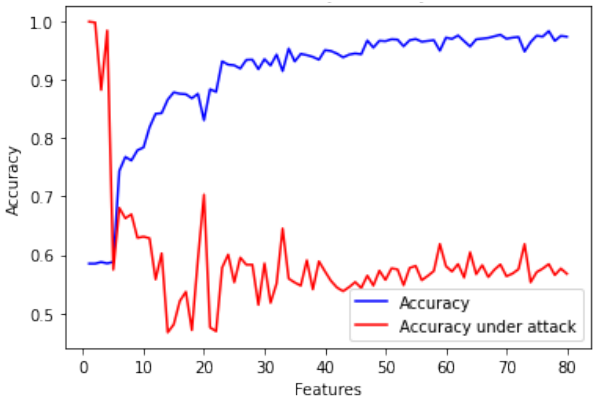
Fig. 5: Features arranged as per original dataset.

IV. Results and Discussion

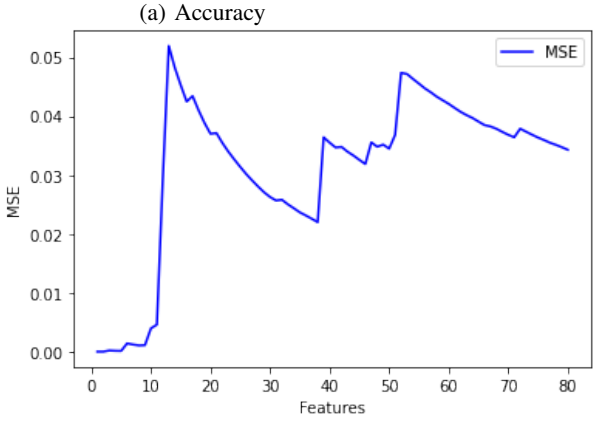
Here we describe and discuss our findings. First we discuss our findings from our initial accuracy and Mean-Square Error study, followed by our findings from our recursive feature elimination experiments.

Figure 5a shows that the model achieves good accuracy (rarely falls below 90%) with fewer features. The model is most accurate when most features are used; however, accuracy under attack rarely exceeds 60%. Showing a well-trained model is susceptible to adversarial attacks such as FGSM. The graphs show an increase in accuracy under normal conditions often correlates with a decrease in accuracy under attack. We note that when Fwd Inter-Arrival (IAT) Total (feature 18) is included, the accuracy under attack improves slightly as shown by a minor peak in accuracy of over 0.5. We consider this accuracy boost a result of the inherent short IAT of DDoS traffic, strongly indicating DoS traffic.

Figure 5b shows the size of perturbation required for a successful attack. The size of perturbation tends to reduce as more features are included. The addition of features increases the attack surface, and allows more subtle adversarial perturbations. The classification results of such systems has serious consequences. Adversaries able to skew the classification accuracy of systems can leverage an advantage by making malicious conditions appear benign. As previously seen with accuracy, we note an increase in perturbation size when Fwd Inter-Arrival (IAT) Total (feature 18) is included. This feature strongly indicates DDoS traffic. We consider the inclusion of important features for classification may also force increases in perturbation size. This in turn means an attack must be more overt.



(a) Accuracy



(b) MSE

Fig. 6: Features arranged from most to least important.

We now consider our experiments with feature-set arranged in order of importance. Figure 6a shows a binary classifier where features have been sorted according to the extracted

No.	Name	Meaning
1	Total_Length_of_Fwd_Packets	Length of Forward Packets
2	_CWE_Flag_Count	Congestion Window Flag
3	_Fwd_Packet_Length_Max	Max Forward Packet Length
4	_Bwd_Avg_Bytes/Bulk	Average No. Backward Bytes/Bulk
5	Subflow_Fwd_Packets	Number of Forward Packets in a Subflow
6	_Flow_IAT_Max	Maximum Inter-Arrival for a flow
7	_Subflow_Bwd_Packets	No. Backward Packets in Subflow
8	_Total_Fwd_Packets	Total Forward Packets
9	_Flow_IAT_Mean	Mean Inter-Arrival
10	Fwd_PSH_Flags	Forward Push Flag
11	_Down/Up_Ratio	Down/Up Ratio
12	_Source_Port	Source Port
13	_Protocol	Protocol
14	_Fwd_Packet_Length_Min	Minimum Forward Packet Length
15	_Flow_IAT_Std	Inter-Arrival Standard Deviation for flow
16	_Flow_IAT_Min	Minimum Inter-Arrival Standard Deviation for flow
17	_URG_Flag_Count	Urgent flag count
18	_Fwd_Avg_Bulk_Rate	Average No. Forward Bytes/Bulk
19	_Subflow_Fwd_Bytes	No. forward bytes in subflow
20	_Bwd_Packet_Length_Min	Minimum backward packet length

TABLE II: Feature-set of 20 most important features.

feature importance. We note with fewer than five features the model predicts the class incorrectly around 40% of the time. This is a poor binary prediction model. The binary FGSM attack aims to flip the recognized class. Therefore for poor classifiers the accuracy can curiously increase under FGSM attack. As more features are included, the accuracy wavers depending on specific properties of those features. We observe a roughly inverse relationship between accuracy and the accuracy under FGSM attack. Where accuracy falls, this coincides with an increase in accuracy under attack and vice versa. We use this graph to determine a set of features providing good accuracy under FGSM attack, whilst retaining acceptable accuracy under normal conditions. We find a promising peak at feature 20 (_Bwd_Packet_Length_Min). The cumulative feature-set of 20 most important features is shown in table II.

This feature-set provides good accuracy under FGSM attack, whist maintaining acceptable accuracy under normal conditions. Focusing our attention either side of this peak we note drops in accuracy with the removal of _Bwd_Packet_Length_Min (19 features), or the addition of _Flow_Duration (21 features). The inclusion of _Bwd_Packet_Length_Min (20 features) gives a local maxima for accuracy under FGSM attack. Whilst maintaining good accuracy ($\approx 85\%$) under normal conditions. The _Bwd_Packet_Length_Min feature may indicate DDoS traffic through the size of returned packets. Each feature in isolation may not be an excellent indicator of DDoS traffic; however in combination a distinct pattern may emerge. For example, a ping flood attack is performed by quickly sending a large multiple of small request packets gaining an equal number of response packets. Packet Length, Number of packets, IAT, and Down/Up Ratio when combined could reveal such traffic. These features are well represented in our generated feature-set (table II).

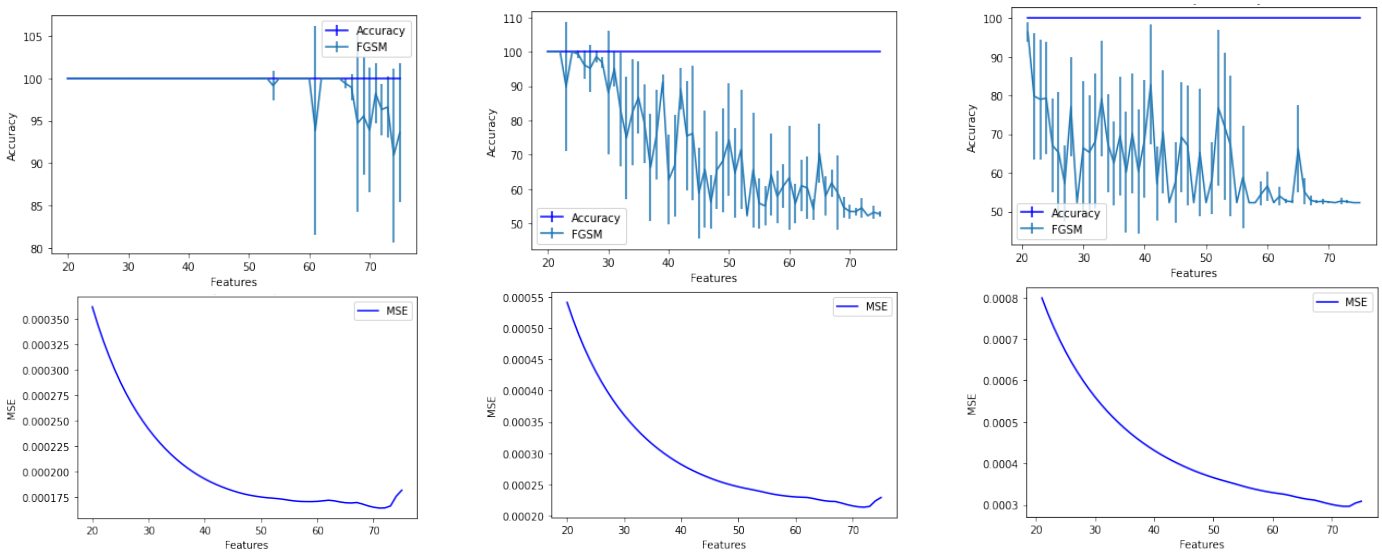
Figure 6b shows the accompanying plot of the perturbation size by number of features, resembling an imperfect saw tooth. We use this graph to determine a feature-set maximising the perturbation size of successful FGSM attacks. Our plot shows relatively small perturbations are necessary until a significant spike occurs with _Source_Port (feature 12), followed by a gradual decline until another spike at _Subflow_Bwd_Bytes (feature 40). Further peaks and gradual declines are seen with _Packet_Length_Variance (feature 50) and _Bwd_IAT_Min (feature 70). Again, individual features may have an effect; however we consider grouping features in a cumulative feature-set effects the perturbation size more. We note that the maximum size of perturbation is smaller when using our feature-set sorted by feature importance (0.05) compared against the maximum perturbation of our original (unsorted) feature-set (0.30). The cumulative feature-set detailed in table II peaks with _Source_Port (feature 12). Looking either side of this feature we note the removal of _Down/Up_Ratio or the addition of _Protocol. The inclusion of _Source_Port gives a local maxima for MSE (>0.05). We further note that the cumulative feature-set of 20 features yields a relatively high MSE (≈ 0.04); however, this value is much lower than the MSE yielded by our original feature-set. The MSE is decreased in our sorted feature-set. The improved accuracy under FGSM attack is an effect of selecting and grouping features. Forced increases in perturbation size may also have a smaller effect.

A. Feature Selection

There are many types of feature importance, which highlight which features may be most/least relevant. Through identifying the relevance of features insights can be gleaned on the dataset and model. We use these insights to improve our predictive model, by discarding features more susceptible to FGSM attack. It is known that reducing the number of features can yield benefits including: reduced time required to train a model [31], improved accuracy, and reduced execution time [24]. We further explore whether robustness can be improved through feature selection. Three common methods for determining feature importance are: model coefficients, decision trees and permutation testing. Our focus is on the latter.

Now we discuss our findings from our RFE experiments. Figure 7 shows plots of accuracy and average perturbation per feature. In FGSM the adversarial "noise" is scaled by a small number (epsilon). The plots illustrate the effect of FGSM on accuracy for different values of epsilon (ϵ). The negative effect on classification accuracy increases with the size of ϵ. For $\epsilon = 0.10$ and $\epsilon = 0.15$ accuracy for the original dataset near 50%. It should be noted that for binary classification tasks an accuracy of 50% equates to a random guess. For all values of epsilon we see an incremental increase in robustness against FGSM. Such AEs can be successfully mitigated with feature selection. Where $\epsilon = 0.05$ a feature-set of approximately 50 features is sufficient to negate the effects of FGSM. Where $\epsilon = 0.10$ a feature-set of approximately 20 features is sufficient to negate the effects of FGSM. Whereas for $\epsilon = 0.15$ a feature set of approximately 20 features is unable to fully negate the effect of FGSM.

All values of ϵ show a similar trend of increased average perturbation per feature; however, we note that as ϵ increases the average perturbation per feature decreases. This can be explained if perturbations are unevenly distributed across features. Large perturbations of a small set of features and small



(a) Accuracy(top) and Average perturbation size per feature(bottom) $\epsilon = 0.05$

(b) Accuracy(top) and Average perturbation size per feature(bottom) $\epsilon = 0.10$

(c) Accuracy(top) and Average perturbation size per feature(bottom): $\epsilon = 0.15$

Fig. 7: Accuracy and average perturbation per feature for feature sets of decreasing size, with ϵ values of: 0.05, 0.10, and 0.15

or no perturbations on other features give smaller average perturbations for the sample.

We have shown that feature selection can improve classification accuracy under adversarial conditions. We used a DDoS case study using the CICIDS2017 dataset. We have shown improvement of FGSM accuracy by recursively removing features most susceptible to large perturbations from the training set. Our research shows an improvement from 58.57% +/- 15.03 to 78.63 % +/- 8.23 (results at the edge of the standard deviation indicate FGSM accuracy 86.86%) with no drop in accuracy for unperturbed samples.

B. Parallel Co-ordinates

Our parallel-coordinates plot in figure 4 shows that DDoS features modified by FGSM fall within the distribution of benign traffic for these features. It is clear that benign and malicious traffic cannot be separated based solely on the range of their features. Instead we theorise that the correlations between features can help separate benign and malicious traffic. Significantly a pattern of peaks and troughs emerge from the FGSM distribution. We theorise that this pattern is more easily concealed in large feature sets where such patterns may be harder to detect.

C. Future Work

We focus on fooling the intrusion detection algorithms, deploying FGSM [17] to produce adversarial examples; however certain features must remain unchanged. For example, destination address and port number must remain unchanged, ensuring the packet is delivered to the target. Furthermore, in network traffic features must remain within reasonable bounds in order to remain inconspicuous. Moreover, many features should ideally remain intrinsically consistent within packets. In other words, counts and other statistics should remain logical and true. Adversaries can control other features and could reasonably and readily change packet length. Successful attacks should constrain which features can be modified in generated adversarial examples. The excellent accuracy results we observed could potentially be a sign of over-fitting which we will consider in future work. We also seek to explore the robustness benefit gained through use of multiple separately trained ML models, perhaps combining them into an ensemble classifier. An investigation into whether an ensemble classifier such as a Random Forest provide greater robustness against adversarial examples. Furthermore, investigation into the transferability of attacks between an Artificial Neural Network (ANN) model and a random forest will be explored. This will be used to determine to what extent our defences are susceptible to transfer attacks.

So far we have examined binary classification. The multi-class problem is more complex. A multi-class classifier must determine which of many classes a suspect sample belongs. Moreover, an adversary can choose the target class for an adversarial example. This could be advantageous: a network analyst would certainly treat a DDoS attack differently than a BotNet or infiltration attempt. Adversaries could gain significant advantage through camouflaging an infiltration attack as a comparatively less serious network intrusion.

V. Conclusion

We have demonstrated a generalisable approach for assessing the vulnerability and robustness of features in a ML context. In particular, adversarial ML attacks seek to identify subtle perturbations of features that can result in mis-classification. Our approach provides researchers with a suitable methodology for assessing how susceptible features

may be towards perturbation attacks, and how we can systematically remove vulnerable features to simultaneously maintain acceptable classifier accuracy whilst eliminating features that may introduce subtle attack vectors. To demonstrate the concept, we applied our approach to a network traffic classification task to distinguish between malicious DDoS activity and benign traffic behaviours. We successfully use feature selection to achieve improvement in accuracy under FGSM attack.

REFERENCES

[1] A. Sfakianakis, C. Douligeris, L. Marinos, M. Lourenço, and O. Raghimi, "Enisa threat landscape report 2018: 15 top cyberthreats and trends," https://www.enisa.europa.eu/publications/enisa-threat-landscape-report-2018, 2019.

[2] R. Satter, "Experts who wrestled with solarwinds hackers say cleanup could take months - or longer," -12-24 2020. [Online]. Available: https://www.reuters.com/article/us-global-cyber-usa-solarwinds-idUSKBN28Y1K3

[3] S. Sirota, "Air force response to solarwinds hack: Preserve commercial partnerships, improve transparency into security efforts," *Inside Cybersecurity*, Jan 12 2021, name - Department of Defense; Copyright - Copyright Inside Washington Publishers Jan 12, 2021; Last updated - 2021-01-13. [Online]. Available: https://search-proquest-com.ezproxy.uwe.ac.uk/trade-journals/air-force-response-solarwinds-hack-preserve/docview/2477182241/se-2?accountid=14785

[4] 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.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014.

[6] N. Papernot, F. Faghri, N. Carlini, I. Goodfellow, R. Feinman, A. Kurakin, C. Xie, Y. Sharma, T. Brown, A. Roy *et al.*, "Technical report on the cleverhans v2. 1.0 adversarial examples library," *arXiv preprint arXiv:1610.00768*, 2016.

[7] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," *arXiv preprint arXiv:1802.00420*, 2018.

[8] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2013, pp. 387–402.

[9] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," IEEE, pp. 39–57, 2017.

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[11] M. A. Ayub, W. A. Johnson, D. A. Talbert, and A. Siraj, "Model evasion attack on intrusion detection systems using adversarial machine learning," in *2020 54th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2020, pp. 1–6.

[12] C. Buckner, "Understanding adversarial examples requires a theory of artefacts for deep learning," *Nature Machine Intelligence*, vol. 2, no. 12, pp. 731–736, 2020. [Online]. Available: https://doi.org/10.1038/s42256-020-00266-y

[13] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." in *ICISSP*, 2018, pp. 108–116.

[14] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "The space of transferable adversarial examples," 2017.

[15] E. K. Viegas, A. O. Santin, and L. S. Oliveira, "Toward a reliable anomaly-based intrusion detection in real-world environments," *Computer Networks*, vol. 127, pp. 200–216, 2017.

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

[17] A. Kurakin, I. Goodfellow, S. Bengio, Y. Dong, F. Liao, M. Liang, T. Pang, J. Zhu, X. Hu, C. Xie *et al.*, "Adversarial attacks and defences competition," in *The NIPS'17 Competition: Building Intelligent Systems*. Springer, 2018, pp. 195–231.

[18] A. U. H. Qureshi, H. Larijani, M. Yousefi, A. Adeel, and N. Mtetwa, "An adversarial approach for intrusion detection systems using jacobian saliency map attacks (jsma) algorithm," *Computers*, vol. 9, no. 3, p. 58, 2020.

[19] T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman, "Random synaptic feedback weights support error backpropagation for deep learning," *Nature communications*, vol. 7, no. 1, pp. 1–10, 2016.

[20] M. Amer and T. Maul, "Weight map layer for noise and adversarial attack robustness," *arXiv preprint arXiv:1905.00568*, 2019.

[21] S. Wang, X. Wang, P. Zhao, W. Wen, D. Kaeli, P. Chin, and X. Lin, "Defensive dropout for hardening deep neural networks under adversarial attacks," in *Proceedings of the International Conference on Computer-Aided Design*, 2018, pp. 1–8.

[22] T. Hamed, R. Dara, and S. C. Kremer, "Network intrusion detection system based on recursive feature addition and bigram technique," *computers & security*, vol. 73, pp. 137–155, 2018.

[23] G. Farahani, "Feature selection based on cross-correlation for the intrusion detection system," *Security and Communication Networks*, vol. 2020, 2020.

[24] O. Almomani, "A feature selection model for network intrusion detection system based on pso, gwo, ffa and ga algorithms," *Symmetry*, vol. 12, no. 6, p. 1046, 2020.

[25] P. Legg, J. Smith, and A. Downing, "Visual analytics for collaborative human-machine confidence in human-centric active learning tasks," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, p. 5, Feb 2019.

[26] S. Yoo, J. Jo, B. Kim, and J. Seo, "Hyperion: A visual analytics tool for an intrusion detection and prevention system," *IEEE Access*, vol. 8, pp. 133 865–133 881, 2020.

[27] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews. Computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[28] L. van der Maaten, L. van der Maaten, G. Hinton, and G. Hinton, "Visualizing non-metric similarities in multiple maps," *Machine learning*, vol. 87, no. 1, pp. 33–55, 2012.

[29] L. McInnes, J. Healy, N. Saul, and L. Großberger, "Umap: Uniform manifold approximation and projection," *Journal of open source software*, vol. 3, no. 29, p. 861, 2018.

[30] S. Ustebay, Z. Turgut, and M. A. Aydin, "Intrusion detection system with recursive feature elimination by using random forest and deep learning classifier." IEEE, 2018, pp. 71–76.

[31] S. M. Kasongo and Y. Sun, "Performance analysis of intrusion detection systems using a feature selection method on the unsw-nb15 dataset," *Journal of Big Data*, vol. 7, no. 1, pp. 1–20, 2020.