Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Hardening machine learning denial of service (DoS) defences against adversarial attacks in IoT smart home networks

Eirini Anthi[*], *Lowri Williams, Amir Javed, Pete Burnap*

Cardiff University, School of Computer Science & Informatics, Cardiff, UK

ARTICLE INFO

Article history:
Received 20 January 2021
Revised 16 May 2021
Accepted 24 May 2021
Available online 1 June 2021

Keywords:
Internet of things (IoT)
Smart homes
Networking
Supervised machine learning
Adversarial machine learning
Attack detection
Intrusion detection systems

ABSTRACT

Machine learning based Intrusion Detection Systems (IDS) allow flexible and efficient automated detection of cyberattacks in Internet of Things (IoT) networks. However, this has also created an additional attack vector; the machine learning models which support the IDS's decisions may also be subject to cyberattacks known as Adversarial Machine Learning (AML). In the context of IoT, AML can be used to manipulate data and network traffic that traverse through such devices. These perturbations increase the confusion in the decision boundaries of the machine learning classifier, where malicious network packets are often miss-classified as being benign. Consequently, such errors are bypassed by machine learning based detectors, which increases the potential of significantly delaying attack detection and further consequences such as personal information leakage, damaged hardware, and financial loss. Given the impact that these attacks may have, this paper proposes a rule-based approach towards generating AML attack samples and explores how they can be used to target a range of supervised machine learning classifiers used for detecting Denial of Service attacks in an IoT smart home network. The analysis explores which DoS packet features to perturb and how such adversarial samples can support increasing the robustness of supervised models using adversarial training. The results demonstrated that the performance of all the top performing classifiers were affected, decreasing a maximum of 47.2 percentage points when adversarial samples were present. Their performances improved following adversarial training, demonstrating their robustness towards such attacks.

1. Introduction

The proliferation in Internet of Things (IoT) devices, which routinely collect sensitive information, is demonstrated by their prominence in our daily lives. Although such devices simplify and automate everyday tasks, they also introduce tremendous security flaws. Current insufficient security measures employed to defend smart devices make IoT the 'weakest' link to breaking into a secure infrastructure, and therefore an attractive target to attackers.

As the number of IoT devices increases exponentially (Gubbi et al., 2013), the number of unknown vulnerabilities and threats also increases, resulting in perimeter defences becoming weaker. Intrusion Detection Systems (IDSs) have emerged as successful attack detection and identification methods in IoT networks. In particular, due to the rapid increase in the development of IoT devices, their heterogeneity, and the amount of data that is produced from such tech-

[*] Corresponding author.
E-mail address: anthies@cardiff.ac.uk (E. Anthi).

https://doi.org/10.1016/j.cose.2021.102352

nologies, machine learning techniques have been integrated to support IDSs in IoT networks to defend against a greater array of attacks (e.g. Amouri et al., 2018; Anthi et al., 2018; Doshi et al., 2018; McDermott et al., 2018; Meidan et al., 2018; Shukla, 2017). Many of these approaches employ supervised machine learning to support the detection of malicious behaviour in IoT. In particular, a recent study by da Costa et al. (2019) reviewing state-of-the-art IDSs for IoT reported that the majority of these systems utilise supervised approaches, such as Support Vector Machines (SVM), Random Forest, and Decision Trees. A recent supervised IDS evaluated using real network data derived from a typical IoT testbed presented by Anthi et al. (2018) also demonstrated that a Decision Tree was the best performing classifier for detecting cyber attacks in IoT.

However, the trained models which support such systems may also be subject to attacks and thus introduce a new attack vector. Attacks that target the machine learning models within these systems are known as Adversarial Machine Learning (AML). The aim is to exploit the weaknesses of the pre-trained model by manipulating data and network traffic that traverse through IoT devices. These perturbations increase the confusion in the decision boundaries of the machine learning classifier, where malicious network packets are often miss-classified as being benign. Consequently, the model's effectiveness can be reduced and such errors are bypassed by the machine learning based detectors, which increases the potential of significantly delaying attack detection and further consequences.

Subsequently, the existence of such techniques suggests that machine learning based detectors may be at risk. More specifically, in the context of IoT, AML can be used to manipulate data from network traffic or data collected from the devices/sensors. From an adversary's perspective, AML can also include perturbations to malicious data to cause an increase in misclassification, consequently bypassing the IDS. As machine learning based detection mechanisms become increasingly common, it is understandable that the adversary's motivation to bypass them also increases. Consequently, machine learning based detectors must be further evaluated against AML attacks.

The experiments presented in this paper focus on hardening Denial of Service (DoS) defences against AML. DoS attacks are considered as being one of the most severe attacks against IoT (Chen et al., 2018; Doshi et al., 2018; Verma and Ranga, 2019). Such attacks affect the services of small networks, such as smart homes (Verma and Ranga, 2019), by targeting the smart devices within such environments (e.g. smart light bulbs, smart door locks, smart televisions) and making them unavailable to the intended users (Dhanjani, 2013; Notra et al., 2014; Ronen and Shamir, 2016; Sivaraman et al., 2015). In this case, securing such devices from DoS attacks has been the main focus in several recent studies (e.g. Anthi et al., 2018; Syed et al., 2020; Vaccari, Aiello, Cambiaso, 2020). An important feature of DoS attacks is that it is feasible to deploy by crafting custom packets. In the context of AML, and as DoS attacks are self-contained, an adversary can manipulate various DoS packet features without voiding the attack. Network packets from other attack types may also be manipulated; however, such packet behaviours are more sensitive to perturbation as they affect the validity of the packet and, subsequently, the attack itself.

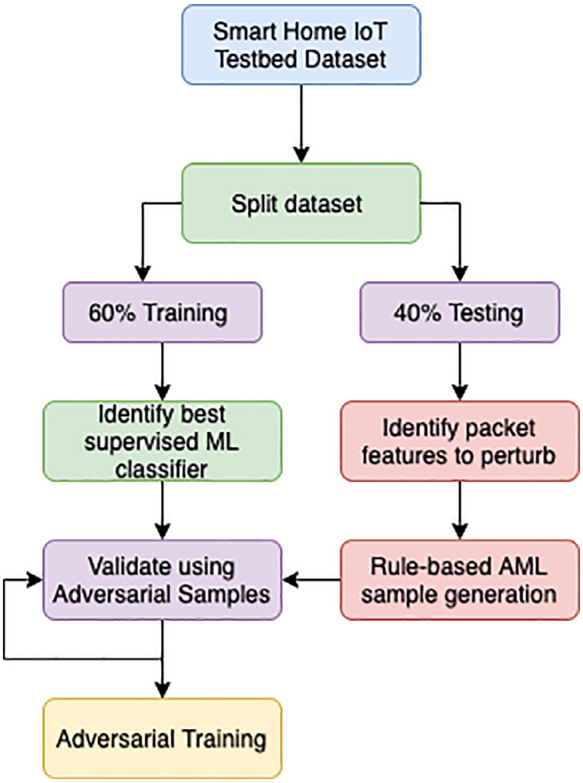
Smart Home IoT Testbed Dataset
Split dataset
60% Training
40% Testing
Identify best supervised ML classifier
Identify packet features to perturb
Validate using Adversarial Samples
Rule-based AML sample generation
Adversarial Training

Fig. 1 – An overview of the study design.

To the best of our knowledge, this is the first investigation into the behaviour of a supervised IDS against an AML attack in the context of IoT using network packet data. The work presented herein considers a realistic attacker model, as well as a dataset collected from a representative smart home IoT testbed. The main contributions of the work presented in this paper are the empirical investigations into:

- Generating adversarial samples from a smart home IoT network dataset
- Investigate the behaviour of a range of supervised classification algorithms used for IDSs in IoT networks against these adversarial samples
- Explore how adversarial training can be used to increase the robustness of such models

The study was designed as follows (see Fig. 1): 1) randomly split the smart home IoT network dataset into training and testing set, each containing 60% and 40% data points respectively, 2) evaluate a range of supervised classification algorithms and identify which are the best performing, 3) generate malicious adversarial DoS packets using a rule-based approach, 4) evaluate the performance of the trained model in 2 on the generated adversarial samples in 3, 5) re-train and evaluate the most affected model using a new training dataset which includes a percentage of adversarial samples from 3.

The remainder of this paper is divided into the following main sections: Section 2 presents the related work, Section 3 discusses the data collected as part of the evaluation of a state-of-the-art IDS presented by Anthi et al. (2018) which is used to support the AML experiments herein, Section 4 discusses AML attack types and approaches, Section 5 presents an approach to generate malicious adversarial DoS packets, Section 6 evaluates the performance of the model against AML samples, Section 7 evaluates the performance of the model following adversarial training, and finally Section 8 concludes the paper.

2. Related work

Due to the advancement in machine learning, there has been a substantial increase in IDSs which use such techniques for IoT networks. Nevertheless, there has been significantly less focus on AML in this context. In the field of cybersecurity, the current research surrounding AML focuses on email spam classifiers, malware detection, and very recently, there has been interest in AML against network IDSs for traditional networks and ICS (Anthi, Williams, Rhode, Burnap, Wedgbury, 2021) .

In more detail, in the context of spam classifiers, both Nelson et al. (2008) and Zhou et al. (2012) demonstrated that an adversary can successfully exploit and bypass machine learning methods by including perturbations to a small percentage of the original training data. In addition, Grosse et al. (2017) evaluated the robustness of a neural network trained on the DREBIN Android malware dataset. They reported that the model misclassified the perturbed inputs in the training set. This attack requires the adversary to have some degree of knowledge of both the dataset and its features. Furthermore, Hu and Tan (2017) presented a more sophisticated adversarial technique that uses the concept of GAN to successfully attack malware classifiers without requiring any knowledge of the targeted system or dataset.

In the context of IoT, there exist only a handful of investigations into AML attacks; the majority of which focus on machine learning detection methods for malware. Particularly, Abusnaina et al. (2019) investigated a range of off-the-shelf methods to craft adversarial IoT software and a GEA method. The results show that all adversarial samples were successful in bypassing the detector. Moreover, Han et al. (2019) developed a framework that employs genetic algorithms to generate adversarial samples for IoT Android applications. The framework demonstrated to have a success rate of nearly 100%. Furthermore, there exist a few studies that focus on detecting and defending against adversarial samples in IoT. For instance, Baracaldo et al. (2018) use contextual information about the origin and the transformation of data points in the training set to identify perturbed data in a sensors' measurement dataset.

Furthermore, recent work has focused on AML against traditional network IDSs and ICS. More specifically, Rigaki (2017) use the KDD'99 dataset to generate adversarial samples and demonstrate the effectiveness of AML against supervised algorithms. Moreover, Zizzo et al. (2019) showcase a simple AML attack against an LSTM classifier which was applied on an ICS dataset. This attack required the manual identification of features that needed to be perturbed in order to generate adversarial samples. Yaghoubi and Fainekos (2019) evaluate a gradient-based search approach on a Simulink model from a steam condenser. This approach demonstrated only to be efficient against a handful of systems that employ RNN with smooth activation functions. Erba et al. (2019) present two types of real-time evasion attacks, using RNN models and an autoencoder to generate adversarial samples.

As a result, the work cited above focuses on other areas of cybersecurity, including email spam classifiers and traditional malware detection. In the context of IoT, AML has been used to target IoT software, Android applications, and sensor metric data, and has yet to address the subject of how AML may affect supervised machine learning-based IDSs trained on IoT network traffic data. In the same context, there has yet to be an investigation into how to defend such IDS systems against AML attacks.

3. Attacking a supervised machine learning detector

To support the experiments presented in this paper, as well as to demonstrate how AML can affect relevant supervised machine learning-based detectors, the data collected as part of the evaluation of a state-of-the-art IDS presented by Anthi et al. (2018) was used. In particular, the IDS utilises a Decision Tree classifier to determine whether network packets are malicious, the type of the attack which has occurred, and which device is affected. The focus of this paper is on how AML can be used to generate adversarial DoS packets to bypass supervised models. The following Sections discuss the features present in the smart home IoT network dataset and presents the methodology behind generating adversarial samples and evaluating the best performing supervised classifiers.

3.1. *Dataset*

To support the AML experiments presented in this paper, an authentic and suitable-sized IoT smart home dataset was used. More specifically, Anthi et al. (2018) assembled an IoT testbed consisting of a range of commercially relevant and representative IoT hardware, including the Belkin NetCam camera, TP-Link NC200 Camera, TP-Link Smart Plug, Samsung Smart Things hub, Amazon Echo Dot, British Gas Hive connected to two sensors: a motion sensor and a window/door sensor, and Lifx Lamp. In addition, a laptop was connected to the network to continually record the network traffic and automatically generate and save the log files, and deploy various attacks. For an illustration of the architecture of the testbed and the components used for generating the datasets, see Anthi et al. (2018).

A dataset containing both benign and malicious data points was generated from the smart home IoT testbed in which 3 weeks worth of benign data and 3 weeks of malicious data was collected using the *tcpdump* (Wir, 2018) tool. To generate the malicious data, Anthi et al. (2018) describes the 5 attack types deployed on the testbed: Denial of Service (DoS), Man-In-The-Middle (MITM)/Spoofing, Reconnaissance, and Replay. To support the AML experiments herein, benign packets, as well as packets that were identified as DoS, were selected. The final dataset consisted of 41,236 DoS and 110,390 benign data points.

4. Adversarial machine learning

As aforementioned, AML aims to automatically add perturbations to data points in order to increase the confusion in the decision boundaries of the machine learning classifier. As a result, malicious network packets can then miss-classified as being benign. The following sections introduce the types of AML attacks, as well as the methods used to automatically generate adversarial samples.

4.1. *Adversarial attack types*

Papernot et al. (2016) categorises adversarial attacks based on:

- Their **complexity**. The consequences of such attacks can vary. Slightly reducing the confidence of a model may be considered as having fewer consequences in comparison to significantly reducing its overall precision.
- The **knowledge** an adversary may have may be categorised into three main types of attacks:
  - *White box* attack: when an adversary has knowledge related to the learning model, such as its architecture, the data it reads, and the features used to support its training.
  - *Black box* attack: when an adversary does not know the internal workings of the target model.
  - *Gray box* attack: when an adversary has some knowledge surrounding the model's architecture or the data it reads.

4.2. *An attacker's motive*

There are many reasons why an adversary may wish to deploy a DoS attack against IoT devices within a smart home. The purpose of a DoS attack is not to get unauthorized access or to obtain sensitive data, but to flood the victim's device in order to make these devices and their services unavailable to the user.

For example, in the context of an IoT smart home, devices such as smart cameras may be used for physical security purposes. Attackers may use DoS to cause a camera to blackout, allowing the coast to be clear to physically access a home without creating digital forensic evidence (OConnor et al., 2019).

4.3. *Attacker model*

The work presented herein considers the following attacker model. It is assumed that the attacker does not have physical access to the IoT devices, but has successfully retrieved the password for the central access point within the smart home network. This type of attacker may be physically located within the wireless range of the targeted user's smart home network. An attacker with control over the wireless router can access devices over the local network and can deploy several different attacks (Vanhoef and Piessens, 2014; 2015; 2016). Such an attacker may have a pre-existing relationship with the victim and was given administrative access to the router/network (OConnor et al., 2019) when they were present in the home.

Subsequently, the attacker has the following capabilities:

- Scan the network.
- Passively eavesdrop on the wireless communications.
- Deploy active attacks such as DoS, MAC/ARP Spoofing, and MITM.

The attacker has the following objectives:

- To collect information about the connected devices (i.e. what devices are connected, what ports are open).
- To make the devices unavailable to the intended user by deploying a DoS attack.

Additionally, it is assumed that the smart home's network is protected by utilising a supervised machine learning IDS.

4.4. *Adversarial sample generation methods*

Various methods exist through which adversarial samples can be generated. Such approaches differ in complexity, speed, and efficiency. The aforementioned methods discussed in Section 2 provide sophisticated approaches for generating adversarial samples.

Two relevant techniques towards automatically generating perturbed samples include the Fast Gradient Sign Method (FGSM) and the Jacobian based Saliency Map Attack (JSMA), presented by Goodfellow et al. (2014a) and Papernot et al. (2016) respectively. Both FGSM and JSMA follow similar methodologies, in that adding small perturbations to the original data can result in such samples exhibiting adversarial characteristics and may be classified differently by the targeted model. Both methods are applied by using a pre-trained Multilayer Perceptron (MLP) network as the underlying model for the adversarial sample generation.

Rigaki (2017) evaluated the aforementioned methods on the NSL-KDD dataset for traditional IT systems and demonstrated that such approaches can successfully generate adversarial samples that reduce the performance of the supervised classifier. In addition, presenting a pre-trained model with AML samples generated from a dataset of industrial IoT device measurements demonstrated to significantly reduce its performance by 20 percentage points (Anthi et al., 2021).

Given measurement data from IoT devices, such as recorded temperatures from a sensor, the aforementioned approaches may be applicable. However, such approaches assume that all features can be equally perturbed by the same predefined constant. Thus, when considering network packet features, this may mean that perturbing these values outside of their valid ranges may jeopardise the validity of the packet, and subsequently the attack. For instance, a flag can only be 0 or 1 and the packet length must have a maximum integer value of 64 Kilobytes. Therefore, the aforementioned methods for generating adversarial samples may be ineffective when applied to network packets.

5. Generating adversarial samples

With the limitations of the approaches discussed in Section 4.4 in mind, this paper proposes a rule-based approach towards generating AML DoS attack samples that aim to target the supervised models which may support IDSs in smart home IoT environments.

The proposed approach is evaluated using malicious DoS packets against IoT devices. The rationale for choosing this type of attack is twofold; 1) DoS is one of the most catastrophic attacks against IoT devices (Chen et al., 2018; Doshi et al., 2018; Verma and Ranga, 2019), and 2) DoS attacks are not connection-based; therefore, the packets are self-contained and their features can be manipulated without voiding the attack.

Inspired by the JSMA and FGSM methods, the proposed approach aims to manipulate DoS attack packet features by considering:

1. **Feature Importance** - identifying the most important features that aid in attack detection.
2. **Practicality** - perturbing packet features that an adversary can modify by changing the attack configurations or by using packet crafting tools such as Scapy. Scapy (2020).
3. **Validity** - given their practicalities, perturbing packet feature values between their valid ranges.

5.1. *Feature selection*

Given the dataset discussed in Section 3.1 all benign and DoS packets were extracted. For this analysis, it is essential to highlight that capture related features (e.g. *caplen, frame.enacp_type, frame.offset_shift, frame.len, frame.cap_len, frame.marked, frame.ignored*) provided by the network sniffer (i.e. *tcpdump*) were omitted from the feature space. The rationale behind this is that such features are not included in the original packet feature space and are generated by the network traffic tool. Therefore, these features cannot be directly manipulated by an adversary. However, the *tcp.delta_time* feature was not omitted as it can be indirectly manipulated by an adversary who may want to increase or delay the time between the sending of DoS packets.

Having removed the aforementioned attributes from the dataset, the Information Gain filter, *InfoGain Ratio Attribute Evaluation*, provided as part of Weka (2020) was used to identify which features best discriminate between the malicious and benign packets. Due to its computational efficiency and simple interpretation, Information Gain is one of the most popular feature selection methods (Tang et al., 2014) and has been used for feature selection in other relevant work (e.g. Alazab et al., 2012; Anthi et al., 2018; Effendy et al., 2017).

This filter evaluates the importance of the features in the training dataset by measuring their information gain with respect to the classes. In more detail, this filter measures how each feature contributes to decreasing the overall entropy - a measure to calculate the degree of disorder or uncertainty. Subsequently, an important feature holds the most information and reduces the entropy the most (Sharma and Dey, 2012). The entropy $H(Class)$ for each class is defined in Eq. (1), where p_i is the probability of randomly selecting an instance of class i from the dataset and log_2 is the base 2 logarithm. The Information Gain is defined in Eq. (2), where $H(Class)$ is the previously defined entropy for each class and $H(Class|Attribute)$ is the sum of the entropies of a specific attribute A for each class. For a working example of how the entropy and Information Gain are calculated, see Omuya et al. (2021).

Table 1 – Feature importance ranking using InfoGain Ratio Attribute Evaluation.

Attribute	Weight
len	0.873
tcp.time_delta	0.731
ip.flags.df	0.675
ip.flags.mf	0.298
ip.frag_offset	0.278
ip.ttl	0.178
tcp.seq	0.169
ip.proto	0.091
icmp.type	0.040
icmp.code	0.040
tcp.window_size	0.021
tcp.flags.urg	0.021
tcp.flags.cwr	0.021
tcp.len	0.021
tcp.flags.ecn	0.021

$$H(Class) = -\sum p_i \, log_2 \, p_i \tag{1}$$

$$I(Class, Attribute) = H(Class) - H(Class|Attribute) \tag{2}$$

Table 1 illustrates the top 15 features which best discriminate between benign and DoS packets with their respective information weight ranking. The remaining features resulted in a much lower importance score. Based on these results, as well as domain knowledge and practicality, the following features were chosen to be manipulated to generate adversarial packets: *len, tcp.time_delta, ip.flags.df, ip.flags.mf, ip.ttl, tcp.flags.urg, tcp.flags.cwr*, and *tcp.flags.ecn*.

More specifically, adversaries may increase the network packet size by introducing padding to the packet header or they may reduce its size by fragmenting a single packet into more packets (Kirda and Trachtenberg, 2009). The *tcp.time_delta* measures how much time has elapsed between the arrival of the prior packet and the current packet. Lower values of delta times correspond to higher rates of transmitted packets which may indicate that a DoS attack has occurred. Although this is a feature calculated by the network sniffer tool, it will be used in this work to explore how a lower rate flow of packets can affect the supervised classifier.

The *ip.flags.df* can be set to indicate that a packet cannot be fragmented for transmission. The *ip.flags.mf* can be set to indicate that the packet contains more fragments. Time To Live (TTL) refers to the amount of time or number of hops a packet is set to exist inside a network before being discarded by a router. When crafting or manipulating packet features, the TTL value can be specified and set between 0 and 255. The TCP flags can also be set or unset, the *tcp.flags.urg*

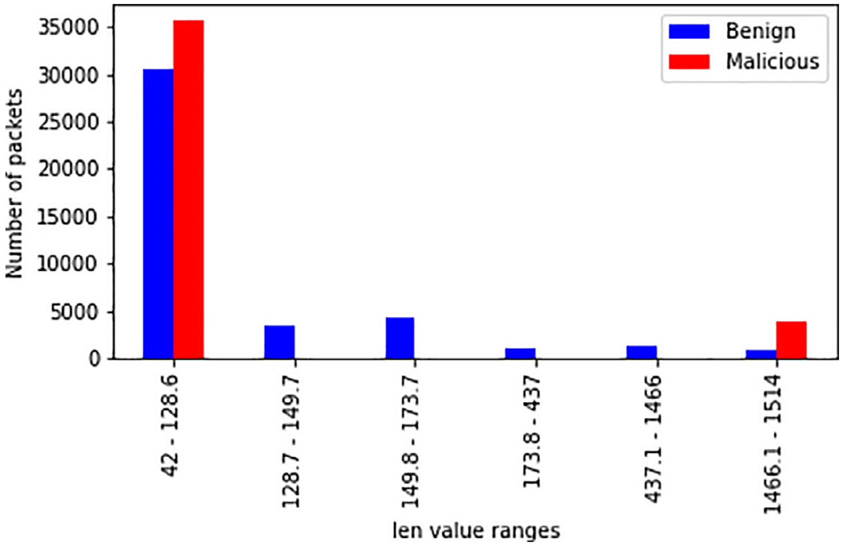

Fig. 2 – Distribution of *len* values for both benign and malicious packets.

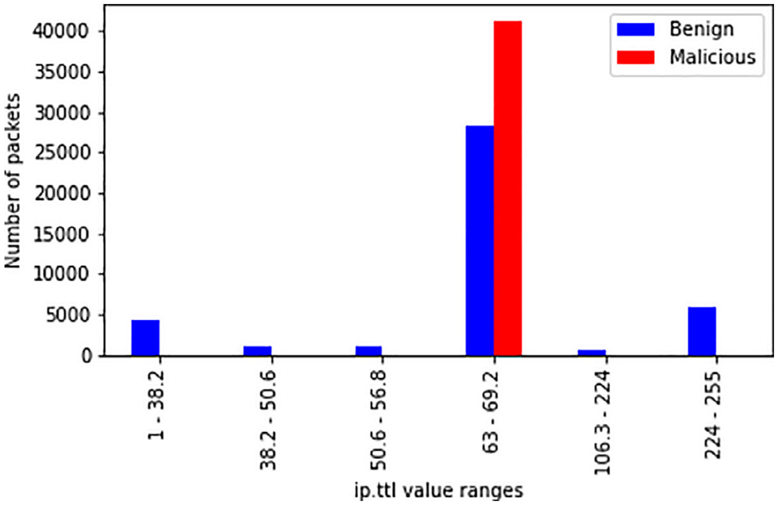

Fig. 3 – Distribution of *ip.ttl* values for both benign and malicious packets.

is used to indicate whether to abort other segments so that the given segment is given priority, the *tcp.flags.cwr* indicates that the host received a TCP segment with the ECE flag set and had responded in congestion control mechanism and the *tcp.flags.ecn* flag is used to echo back the congestion indication. An adversary can craft packets where an invalid or unusual combination of flags can be set.

To better understand the structure of benign IoT network packets, and subsequently, define the ranges in which these features can be perturbed, the distribution of the values of the *len* and *ip.ttl* features in the benign packets were analysed. Figure 2 reports the distributions of the values for the *len* feature for both packet types. The minimum *len* value for benign packets was reported as 52, with the maximum value being 1,514. A significantly large number of *len* values for the benign packets (22,921) fall between the ranges of 54 and 194.

Figure 3 reports the distributions of the values for the *ip.ttl* feature for both packet types. The minimum *ip.ttl* value for benign packets was reported as 1, with the maximum value being 255. A significantly large number of *ip.ttl* values for the benign packets (20,633) fall between the ranges of 30 and 70.

5.2. *Model training*

Given the uneven number of classes, the dataset described in Section 3.1 was balanced to consist of 41,236 samples of both packet types. Subsequently, a random subset of approximately 60% of the dataset was selected for training, with 24,741 samples of each class. The remaining 40% of the dataset was used for testing, with 16,495 samples of each class.

Previous work by Anthi et al. (2018) reported that Weka's implementation of Ross Quinlan's C4.5 algorithm (Quinlan, 2014), the J48 Decision Tree method with no pruning, was the best performing classifier in discriminating between different cyber attacks. In this paper, to explore how well supervised machine learning algorithms can detect DoS attacks

Table 2 – Weighted average results following 10-fold cross-validation.

Classifier	P	R	F
J48 Decision Tree	0.999	0.999	0.999
Random Forest	0.999	0.999	0.999
Naive Bayes	0.997	0.997	0.997
Bayesian Network	0.999	0.999	0.999
SVM	0.999	0.999	0.999
Zero R	0.500	0.500	0.405
One R	0.972	0.970	0.970

in an IoT environment, the corresponding smart home dataset was used to evaluate a range of state-of-the-art classifiers distributed as part of Weka.

Classifiers included generative models that consider conditional dependencies in the dataset or assume conditional independence (e.g. Bayesian Network, Naive Bayes) and discriminative models that aim to maximise information gain or directly map data to their respective classes without modeling any underlying probability or structure of the data (e.g. J48 Decision Tree, SVM).

Table 2 demonstrates the results following classification, reporting models with the highest performances, with each using their default parameters. The overall performance represents weighted-averaged of precision (P), recall (R), and F1-score (F) for all experiments.

Overall, the classification performance for detecting DoS-specific attacks across each classifier achieved a high result. This is intuitive to DoS attacks as such packets have distinct values (i.e. *len* and *ip.ttl* values) in comparison to those that are benign. In particular, the classification performances of the J48 Decision Tree with no pruning, Random Forest, Bayesian Network, and SVM achieved the best performances, resulting in F1-scores of 99.9%.

5.3. *Generating perturbed samples*

Based on the observations in Section 5.1 and to support the initial AML experiments within this paper, a range of feature combinations were perturbed. Firstly, to investigate how perturbing individual features may affect the classifier, adversarial samples were generated where only one of the features was modified at a time.

This approach aims to mask adversarial samples to benign packets as closely as possible. Given the distributions in Section 5.1, the *len* and *ip.ttl* feature values of the malicious packets were perturbed as being a random value between the ranges of 54 and 194 and 30 and 70 respectively. For the flag features, the adversarial samples were generated by randomly setting the flag (1) and unsetting the flag (0). To explore whether a lower rate of packet flow can affect the classifier's performance, the *tcp.time_delta* feature was altered by increasing their values incrementally by five percent up to 50%. Finally, to explore whether DoS packets with lower *tcp.time_delta* values are misclassified as benign, adversarial samples were generated when all features, excluding *tcp.time_delta*, were perturbed.

To avoid bias, and by drawing inspiration from the cross-validation method (Refaeilzadeh et al., 2009), 20 iterations of perturbed samples for each feature-set were generated. Table 3 shows an example of how a malicious DoS packet may be modified when all features, excluding *tcp.time_delta*, are perturbed during the first 5 iterations given this approach.

It is worth highlighting, although such method of perturbation may be considered forceful, this level of perturbation is possible to be achieved by an adversary, specifically in IoT network environments. This is because the behaviour of the devices are not considered as being variable and do not have extreme deviations. As a result, an adversary can employ passive sniffing techniques to observe the activity of the IoT network, and thus craft and deploy adversarial attacks.

6. Evaluating the model on adversarial samples

The J48 Decision Tree, Random Forest, Bayesian Network, and SVM classifiers were first evaluated on the training dataset using 10-fold cross-validation and applied to the original testing dataset. The F1-score achieved by each classifier was 99.9%. The confusion matrix in Table 5 shows how the predicted classes in the original testing dataset compare against the actual ones following 10-fold cross-validation using the J48 Decision Tree.

To explore the effects of the AML attack on the pre-trained classifier, adversarial samples were generated for all malicious DoS data points present in the testing data by individually perturbing each of the features discussed in Section 5.1, as well as perturbing all features, excluding *tcp.time_delta*. The rationale behind this is to investigate the model's behaviour when the adversary only alters packet features and not the rate of the attack. The original malicious packets were excluded from the testing data. The adversarial samples were subsequently included along with the benign testing data points and presented to the trained model. Table 4 therefore reports the average weighted Precision, Recall, and F1-score following these 20 iterations for each classifier.

When the *tcp.flags.cwr, tcp.flags.ecn, tcp.flags.urg, len* and *tcp.time_delta* were perturbed individually, each of the models' performances were unaffected. This may be explained by the fact that such features have a lower importance score (see Table 1) and also may rely on the values of other features to distinctly discriminate between both packet types.

When all features, excluding *tcp.time_delta*, were perturbed, the classification performance of all the models were affected. In particular, the J48 model achieved an F1-score of 52.7%, the highest decrease across the models (a difference of 47.2 percentage points in comparison to its performance when classifying the original testing data). This may be because the malicious DoS packets were significantly modified, therefore their similarity to the benign packets was increased. In addition, when perturbing the *ip.flags.df* and *ip.ttl* features individually, the J48 model's performance achieved an F1-score of 73.3% and 68.2% respectively; again, the highest decrease across the models (a difference in 26.6 and 31.7 percentage points). This may be explained by the fact that the majority of the benign packets and a small number of DoS packets had the *iip.flags.df*

Table 3 – An example of how malicious packet features are perturbed.

Packet	len	ip.ttl	ip.flag.mf	ip.flag.df	tcp.flags.cwr
Original Packet	94	64	0	0	1
Iteration 1	147	37	1	0	1
Iteration 2	64	36	0	0	1
Iteration 3	129	66	0	0	1
Iteration 4	185	30	0	0	1
Iteration 5	171	48	0	0	0

Table 4 – Classification performances when applied to generated adversarial samples.

	J48 Decision Tree			*Random Forest*			*Bayesian Network*			*SVM*		
Perturbed Features	P	R	F	P	R	F	P	R	F	P	R	F
All features (excluding *tcp.time_delta*)	77.8	60.2	52.7	85.7	80.5	79.8	91.3	89.5	89.3	81.5	70.5	67.7
ip.flags.df	83.3	75.0	73.3	92.8	91.6	91.5	99.5	99.5	99.5	83.3	75.0	73.4
ip.flags.mf	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	91.3	89.4	89.3
tcp.flags.cwr	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9
tcp.flags.ecn	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9
tcp.flags.urg	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9
ip.ttl	81.6	70.9	68.2	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9
len	99.1	99.1	99.1	99.7	99.7	99.7	96.8	96.6	96.6	99.9	99.9	99.9
tcp.time_delta	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9

Table 5 – Confusion matrix for the original testing dataset following 10-fold cross-validation using the J48 Decision Tree.

			Predicted	
			a	b
Actual	DoS	a	16,495	0
	Benign	b	1	16,494

set. Due to the tools used to deploy the DoS attacks, the default TTL value for these packets were set as 64. As a result, perturbing the TTL value between the aforementioned ranges significantly altered the distribution of the feature values. Subsequently, this demonstrated to impact the classifier's performance. Given these results, to achieve the most impact, an adversary would have to perturb all selected features excluding modifying the rate of the attack (i.e. *tcp.time_delta*) to successfully reduce the performance of a machine learning based IDS that uses either of the four classifiers to support the classification of DoS packets, and subsequently divert malicious data points.

The confusion matrices in Tables 6, 7, and 8 provide a better insight into the performance of the J48 Decision Tree across the experiments. In comparison to the original performance in Table 5, the model demonstrates a significant increase in false positives when all features, excluding *tcp.time_delta*, and when only *ip.ttl* are perturbed. In addition to these results, when the *ip.flags.df* feature is perturbed, the model reports a higher false positive rate of almost 50%.

Table 6 – Confusion matrix after perturbing all select features (excluding *tcp.time_delta*) following 10-fold cross-validation using the J48 Decision Tree.

			Predicted	
			a	b
Actual	DoS	a	3,345	13,150
	Benign	b	1	16,494

Table 7 – Confusion matrix after perturbing *ip.ttl* following 10-fold cross-validation using the J48 Decision Tree.

			Predicted	
			a	b
Actual	DoS	a	6,901	9,594
	Benign	b	1	16,494

Table 8 – Confusion matrix after perturbing *ip.flags.df* following 10-fold cross-validation using the J48 Decision Tree.

			Predicted	
			a	b
Actual	DoS	a	8,245	8,250
	Benign	b	1	16,494

Table 9 – Perturbed data iteration with highest impact on the model's performance.

Classifier	Iteration	F1-score
J48 Decision Tree	1	52.6
Random Forest	7	79.5
Bayesian Network	19	89.0
SVM	20	67.2

7. Defending against adversarial machine learning

There exist a few methods that attempt to defend against AML attacks. Two of the most common approaches include adversarial training and adversarial sample detection. Goodfellow et al. (2014b) demonstrated that re-training a model on a dataset containing both the original and adversarial data samples significantly improves its efficiency against adversarial samples. The second method involves developing mechanisms of detecting adversarial samples using direct classification, neural network uncertainty, or input processing (Zizzo et al., 2019). However, such detection mechanisms have not demonstrated as being robust enough in defending again AML (Athalye et al., 2018; Zizzo et al., 2019).

Subsequently, given the positive findings of how AML affects supervised detectors, the classifiers were further evaluated using adversarial training. In this case, a random sample of 10% of the adversarial data points (1,650 packets) when all features, excluding *tcp.time_delta*, were perturbed and achieved the highest decrease in the model's performance were included in the original training dataset. Table 9 reports the iteration of data, as well as the F1-score, achieved when each model were applied to the perturbed data.

The experiments described in Section 6 were repeated by retraining the models on the newly generated training datasets and applying them on the unseen adversarial samples generated in the remaining iterations. Table 10 reports the average precision, recall, and F1-score following 20 iterations which included newly selected random perturbed samples in the training set following 10-fold cross-validation.

The results demonstrate that including adversarial samples in the training data increased the performances of each model. For each combination of features, the classification performance achieved an F1-score of over 90%, an increase of over 25 percentage points in comparison to the classification performances reported in Table 4. These results are intuitive, as, during adversarial training, the classifiers are trained to recognise the extended decision boundaries of the features which discriminate between benign and malicious packets.

8. Conclusion

Machine learning based IDSs are known as being fundamental methods for detecting cyber attacks in IoT systems due to their reliability and versatility. However, as shown by the results presented herein, it is evident that machine learning based detectors are vulnerable to attacks that may severely undermine or mislead their capabilities. Adversarial Machine Learning (AML) may have significant repercussions for IoT infrastructures, as adversaries may alter malicious DoS data points to bypass the IDS, causing delayed detection of threats, leakage of confidential information, and severe harm. Therefore, in order to develop more robust machine learning based IDSs, it is apparent that understanding the applicability of AML attacks in IoT systems is crucial.

This paper explored how adversarial attacks can be used to target supervised classifiers by presenting generated adversarial DoS samples to a trained model and understanding their classification behaviours. To support the experiments in this paper, an IoT network dataset containing benign and DoS packets were used to train and test a selection of state-of-the-art supervised classifiers, including the J48 Decision Tree, the best performing classifier for detecting malicious and benign packets in the IDS presented by Anthi et al. Anthi et al. (2018). The experiments herein focused on DoS attack packets as it is one of the most severe attacks against IoT devices, it is feasible to deploy by crafting custom packets, and finally, due to the nature of DoS, an adversary can manipulate packet features without voiding the attack.

To identify which features can be manipulated, the importance of the features for discriminating against both packet types was measured. Based on these results, the top-ranked features were selected for perturbation to generate adversarial packets. Firstly, to investigate how individual features may affect the classifier, adversarial samples were generated where only one of the features were modified at a time. An adver-

Table 10 – Classification performances following adversarial training.

	J48 Decision Tree			*Random Forest*			*Bayesian Network*			*SVM*		
Perturbed Features	P	R	F	P	R	F	P	R	F	P	R	F
All features (excluding *tcp.time_delta*)	99.5	99.5	99.5	99.9	99.9	99.9	99.9	99.9	99.9	99.1	99.1	99.1
ip.flags.df	99.7	99.7	99.7	99.9	99.9	99.9	99.9	99.9	99.9	94.6	94.0	94.0
ip.flags.mf	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.8	99.8	99.8
tcp.flags.cwr	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9
tcp.flags.ecn	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9
tcp.flags.urg	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9
ip.ttl	99.8	99.8	99.8	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9
len	99.1	99.1	99.1	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9
tcp.time_delta	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9

sarial dataset was generated where all features, excluding *tcp.time_delta*, were perturbed. Such samples were evaluated against four trained models, including J48 Decision Tree, Random Forest, Bayesian Network, and SVM. The results demonstrate that perturbing all features, excluding *tcp.time_delta*, achieved the highest impact on the J48 model as the classification performance decreased by 47.2 percentage points (from 99.9% to 52.7%).

Given these positive findings, the paper also explores how adversarial samples can enhance the robustness of the models using adversarial training. A random sample of 10% of the generated adversarial data points when all the features were perturbed was included in the original training dataset. The models were retrained and applied to all unseen adversarial samples, excluding the adversarial samples included in the training set. Overall, the classification performances significantly increased when adversarial samples were present in the training datasets.

The results reported herein demonstrate that the proposed approach towards generating adversarial DoS packet samples is effective in reducing the performance of all the top-performing classifiers. Subsequently, this indicates that this method is successful in attacking a range of supervised classifiers of different types, mainly generative and discriminative models. There is scope to expand this approach as part of future work to target other attack types, as well as other types of machine learning, such as unsupervised and deep learning methods.

9. Limitations and Future Work

The experiments outlined herein have shown that adversarial DoS samples can successfully be produced in the context of IoT network traffic and can significantly affect the classification efficiency of a supervised machine learning based IDS. However, it is important to highlight that the approach presented herein has its limitations.

One of the main limitations surrounding this work is the crude approach towards the perturbation of the chosen features. That is, here, we assume that the adversary has full knowledge of the dataset and the trained model. Therefore, following the analysis of the benign packets, the adversary can identify the ranges in which the feature values fall into and subsequently map the malicious packets to mimic the behaviours of the benign. The manual overhead associated with this approach may be addressed by utilising a more sophisticated method of generating perturbed packets (e.g. Iterative Gradient Sign, Carlini Wagner, Generative Adversarial Networks (GANs)), where the attacker does not know the system or the dataset.

The work presented in this paper focuses on perturbing malicious DoS packets to bypass the detector. However, this is only the tip of the iceberg. The applicability of such an approach and other AML approaches of bypassing machine learning-based IDSs need to be further investigated for other attack types.

Lastly, with regards to adversarial training, the results demonstrated the efficiency of such an approach to increase the robustness of the IDS. However, it is important to highlight that this method may not always be sufficient as it is difficult to anticipate all possible types of AML attacks against a given system. Therefore, there is a need to investigate other, more sophisticated defence mechanisms.

Declaration of Competing Interest

Please check the following as appropriate:

- All authors have participated in (a) conception and design, or analysis and interpretation of the data; (b) drafting the article or revising it critically for important intellectual content; and (c) approval of the final version.
- This manuscript has not been submitted to, nor is under review at, another journal or other publishing venue.
- The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript
- The following authors have affiliations with organizations with direct or indirect financial interest in the subject matter discussed in the manuscript:

CRediT authorship contribution statement

Eirini Anthi: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing - original draft, Writing - review & editing. **Lowri Williams:** Methodology, Validation, Formal analysis, Software, Writing - original draft, Writing - review & editing. **Amir Javed:** Validation, Software. **Pete Burnap:** Conceptualization, Resources, Supervision.

Acknowledgements

This work was part-funded by the Engineering and Physical Sciences Research Council (EPSRC) - project number EP/S035362/1 (PETRAS 2).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.cose.2021.102352.

REFERENCES

Abusnaina A, Khormali A, Alasmary H, Park J, Anwar A, Mohaisen A. Adversarial learning attacks on graph-based IoT malware detection systems. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). IEEE; 2019. p. 1296–305.

Alazab A, Hobbs M, Abawajy J, Alazab M. Using feature selection for intrusion detection system. In: 2012 International Symposium on Communications and Information Technologies (ISCIT). IEEE; 2012. p. 296–301.

Amouri A, Alaparthy VT, Morgera SD. Cross layer-based intrusion detection based on network behavior for IoT. In: Wireless and Microwave Technology Conference (WAMICON), 2018 IEEE 19th. IEEE; 2018. p. 1–4.

Anthi E, Williams L, Malgortzata G, Theodorakopoulos G, Burnap P. A supervised intrusion detection system for smart home IoT. IEEE Internet Things J. 2018;78:477–90.

Anthi E, Williams L, Rhode M, Burnap P, Wedgbury A. Adversarial attacks on machine learning cybersecurity defences in industrial control systems. J. Inf. Secur. Appl. 2021;58:102717.

Athalye, A., Carlini, N., Wagner, D., 2018. Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. arXiv preprint arXiv:1802.00420.

Baracaldo N, Chen B, Ludwig H, Safavi A, Zhang R. Detecting poisoning attacks on machine learning in IoT environments. In: 2018 IEEE International Congress on Internet of Things (ICIOT). IEEE; 2018. p. 57–64.

Chen Q, Chen H, Cai Y, Zhang Y, Huang X. Denial of service attack on IoT system. In: 2018 9th International Conference on Information Technology in Medicine and Education (ITME). IEEE; 2018. p. 755–8.

da Costa KA, Papa JP, Lisboa CO, Munoz R, de Albuquerque VHC. Internet of things: a survey on machine learning-based intrusion detection approaches. Comput. Netw. 2019;151:147–57.

Dhanjani N. Hacking lightbulbs: Security evaluation of the Philips hue personal wireless lighting system. Internet Things Secur. Eval. Series 2013.

Doshi, R., Apthorpe, N., Feamster, N., 2018. Machine learning DDoS detection for consumer internet of things devices. arXiv preprint arXiv:1804.04159.

Effendy DA, Kusrini K, Sudarmawan S. Classification of intrusion detection system (IDS) based on computer network. In: 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE). IEEE; 2017. p. 90–4.

Erba, A., Taormina, R., Galelli, S., Pogliani, M., Carminati, M., Zanero, S., Tippenhauer, N. O., 2019. Real-time evasion attacks with physical constraints on deep learning-based anomaly detectors in industrial control systems. arXiv preprint arXiv:1907.07487.

Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. In: Advances in Neural Information Processing Systems; 2014. p. 2672–80.

Goodfellow, I. J., Shlens, J., Szegedy, C., 2014b. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

Grosse K, Papernot N, Manoharan P, Backes M, McDaniel P. Adversarial examples for malware detection. In: European Symposium on Research in Computer Security. Springer; 2017. p. 62–79.

Gubbi J, Buyya R, Marusic S, Palaniswami M. Internet of things (IoT): a vision, architectural elements, and future directions. Future Gener. Comput. Syst. 2013;29(7):1645–60.

Han T, Liu C, Yang W, Jiang D. A novel adversarial learning framework in deep convolutional neural network for intelligent diagnosis of mechanical faults. Knowl.-Based Syst. 2019;165:474–87.

Hu, W., Tan, Y., 2017. Generating adversarial malware examples for black-box attacks based on GAN. arXiv preprint arXiv:1702.05983.

Kirda RLE, Trachtenberg A. Recent advances in intrusion detection. Lect. Notes Comput. Sci. 2009;5758.

McDermott CD, Majdani F, Petrovski AV. Botnet detection in the internet of things using deep learning approaches. In: 2018 International Joint Conference on Neural Networks (IJCNN). IEEE; 2018. p. 1–8.

Meidan Y, Bohadana M, Mathov Y, Mirsky Y, Shabtai A, Breitenbacher D, Elovici Y. N-BaIoT-network-based detection of IoT botnet attacks using deep autoencoders. IEEE Pervasive Comput. 2018;17(3):12–22.

Nelson B, Barreno M, Chi FJ, Joseph AD, Rubinstein BI, Saini U, Sutton CA, Tygar JD, Xia K. Exploiting machine learning to subvert your spam filter.. LEET 2008;8:1–9.

Notra S, Siddiqi M, Gharakheili HH, Sivaraman V, Boreli R. An experimental study of security and privacy risks with emerging household appliances. In: 2014 IEEE Conference on Communications and Network Security. IEEE; 2014. p. 79–84.

OConnor T, Enck W, Reaves B. Blinded and confused: uncovering systemic flaws in device telemetry for smart-home internet of things. In: Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks; 2019. p. 140–50.

Omuya EO, Okeyo GO, Kimwele MW. Feature selection for classification using principal component analysis and information gain. Expert Syst. Appl. 2021;174:114765.

Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A. The limitations of deep learning in adversarial settings. In: 2016 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE; 2016. p. 372–87.

Quinlan JR. C4.5: Programs for Machine Learning. Elsevier; 2014.

Refaeilzadeh P, Tang L, Liu H. Cross-validation.. Encyclopedia Database Syst. 2009;5:532–8.

Rigaki M. Adversarial deep learning against intrusion detection classifiers; 2017.

Ronen E, Shamir A. Extended functionality attacks on IoT devices: the case of smart lights. In: 2016 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE; 2016. p. 3–12.

. Scapy: Packet encapsulation; 2020. https://thepacketgeek.com/scapy-p-04-looking-at-packets/. Accessed on 05/14/

Sharma A, Dey S. Performance investigation of feature selection methods and sentiment lexicons for sentiment analysis. IJCA Special Issue Adv. Comput.Commun. Technol. HPC Appl. 2012;3:15–20.

Shukla P. ML-IDS: a machine learning approach to detect wormhole attacks in internet of things. In: Intelligent Systems Conference (IntelliSys), 2017. IEEE; 2017. p. 234–40.

Sivaraman V, Gharakheili HH, Vishwanath A, Boreli R, Mehani O. Network-level security and privacy control for smart-home IoT devices. In: 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). IEEE; 2015. p. 163–7.

Syed NF, Baig Z, Ibrahim A, Valli C. Denial of service attack detection through machine learning for the IoT. J. Inf. Telecommun. 2020;4(4):482–503.

Tang J, Alelyani S, Liu H. Feature selection for classification: a review. In: Data Classification: Algorithms and Applications; 2014. p. 37.

Vaccari I, Aiello M, Cambiaso E. SlowTT: a slow denial of service against IoT networks. Information 2020;11(9):452.

Vanhoef M, Piessens F. Advanced Wi-Fi attacks using commodity hardware. In: Proceedings of the 30th Annual Computer Security Applications Conference; 2014. p. 256–65.

Vanhoef M, Piessens F. All your biases belong to us: Breaking RC4 in WPA-TKIP and {TLS}. In: 24th {USENIX} Security Symposium ({USENIX} Security 15); 2015. p. 97–112.

Vanhoef M, Piessens F. Predicting, decrypting, and abusing WPA2/802.11 group keys. In: 25th {USENIX} Security Symposium ({USENIX} Security 16); 2016. p. 673–88.

Verma A, Ranga V. Machine learning based intrusion detection systems for IoT applications. Wirel. Pers. Commun. 2019:1–24.

. Weka 3 - data mining with open source machine learning software in java; 2020. https://www.cs.waikato.ac.nz/ml/weka/. (Accessed on 20/20/)

. Wireshark; 2018. https://www.wireshark.org/. (Accessed on 07/18/)

Yaghoubi S, Fainekos G. Gray-box adversarial testing for control systems with machine learning components. In: Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control; 2019. p. 179–84.

Zhou Y, Kantarcioglu M, Thuraisingham B, Xi B. Adversarial support vector machine learning. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2012. p. 1059–67.

Zizzo G, Hankin C, Maffeis S, Jones K. Adversarial machine learning beyond the image domain. In: 2019 56th ACM/IEEE Design Automation Conference (DAC). IEEE; 2019. p. 1–4.

Eirini Anthi received a First Class Honours B.Sc. degree in computer science from Cardiff University, UK, in 2016 and is currently working towards the Ph.D. degree in the field of Cyber Security at the same University. Her research revolves around the security and privacy of Internet of Things devices (IoT). Specifically, her work examines the security issues that come along with these devices and tries to identify methods to make them more secure.

Lowri Williams received her Ph.D in Computer Science from Cardiff University, UK, in 2018. Her research interests include natural language processing, sentiment analysis, data mining, machine learning, and language resources.

Amir Javed is a lecturer in the School of Computer Science & Informatics at Cardiff University. He has been involved in several research projects, as a research associate in Cyber Security Analytics at Cardiff University. His research interests includes cybersecurity, data analytics, machine learning and security related to IoT devices.

Pete Burnap is a Professor at Cardiff University and is seconded to Airbus Group to lead Cyber Security Analytics Research heading projects involving the application of Artificial Intelligence, Machine Learning and Statistical Modeling to Cyber Security problems (most recently malware analysis). Pete obtained his B.Sc. in Computer Science in 2002 and his Ph.D: Advanced Access Control in support of Distributed Collaborative Working and Deperimeterization in 2010, both from Cardiff University. He has published more than 60 academic articles stemming from funded research projects worth over 8m and has advised the Home Affairs Biographical Sketch Select Committee, Home Office and Metropolitan Police on sociotechnical research outcomes associated with cyber risk and evolving cyber threats.