Contents lists available at ScienceDirect

Journal of Information Security and Applications

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

Adversarial attacks on machine learning cybersecurity defences in Industrial Control Systems

Eirini Anthi [a,*], Lowri Williams [a], Matilda Rhode [a], Pete Burnap [a], Adam Wedgbury [b]

[a] *Cardiff University, School of Computer Science & Informatics, Cardiff, UK*
[b] *Airbus, Newport, UK*

ARTICLE INFO

Keywords:
Industrial Control Systems
Supervised machine learning
Adversarial Machine Learning
Attack detection
Intrusion Detection System

ABSTRACT

The proliferation and application of machine learning-based Intrusion Detection Systems (IDS) have allowed for more flexibility and efficiency in the automated detection of cyber attacks in Industrial Control Systems (ICS). However, the introduction of such IDSs has also created an additional attack vector; the learning models may also be subject to cyber attacks, otherwise referred to as Adversarial Machine Learning (AML). Such attacks may have severe consequences in ICS systems, as adversaries could potentially bypass the IDS. This could lead to delayed attack detection which may result in infrastructure damages, financial loss, and even loss of life. This paper explores how adversarial learning can be used to target supervised models by generating adversarial samples using the Jacobian-based Saliency Map attack and exploring classification behaviours. The analysis also includes the exploration of how such samples can support the robustness of supervised models using adversarial training. An authentic power system dataset was used to support the experiments presented herein. Overall, the classification performance of two widely used classifiers, Random Forest and J48, decreased by 6 and 11 percentage points when adversarial samples were present. Their performances improved following adversarial training, demonstrating their robustness towards such attacks.

1. Introduction

Industrial Control Systems (ICS) play a key role in Critical National Infrastructure (CNI) concepts such as manufacturing, power/smart grids, water treatment plants, gas and oil refineries, and health-care. Historically, ICS networks and their components were protected from cyber attacks as they ran on proprietary hardware and software and were connected in isolated networks with no external connection to the Internet [1]. However, as the world is becoming more interconnected, there has been a need to connect ICS components and to other networks, allowing remote access and monitoring functionalities. As a result, ICSs are now subject to a range of security vulnerabilities [1].

Given the importance of these systems, they have become an attractive target to an attacker. As these systems control operations in the physical world, the cyber attacks against them may have major consequences for the environment they operate in, and subsequently, its users. It is therefore understandable that the security issues surrounding such systems have become a global issue. Thus, designing robust, secure, and efficient mechanisms for detecting and defending cyber attacks in ICS networks is more important than ever [2].

Although there exist several security mechanisms for traditional IT systems, their integration into ICS systems is challenging mainly for two reasons; (a) ICS devices are resource-constrained, and (b) they include legacy systems and devices that do not support modern security measures. Subsequently, complementary security solutions, such as passive process data monitoring, are promising [3]. This has led to a substantial increase in research focusing on ICS tailored Intrusion Detection Systems (ICS). Such intrusion systems operate by observing the network or sensor data to detect attacks and anomalies that may affect ICS.

Due to their efficiency in detecting attacks, there has been a substantial increase in the application and integration of machine learning within IDSs (e.g. [1,4–10]). However, the introduction of such systems has introduced an additional attack vector; the trained models may also be subject to attacks. The act of deploying attacks towards machine learning-based systems is known as Adversarial Machine Learning (AML). The aim is to exploit the weaknesses of the pre-trained model which has “blind spots" between data points it has seen during training. More specifically, by automatically introducing slight perturbations to the unseen data points the model may cross a decision boundary and classify the data as a different class. As a result, the model's effectiveness can be reduced as it is presented with unseen data points

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

https://doi.org/10.1016/j.jisa.2020.102717

Available online 2 February 2021

that it cannot associate target values to, subsequently increasing the number of misclassifications.

The existence of such techniques means that infrastructures which incorporate machine learning-based IDSs may be at risk of being vulnerable to cyber attacks. In the context of ICS, AML can be used to manipulate data from actuators or other devices by including perturbations to cause malicious data to be classified as being benign, consequently bypassing the IDS. This could lead to delayed attack detection, information leakage, financial loss, and even loss of life. It is therefore understandable that as machine learning-based detection mechanisms become more widely deployed, the adversary incentive for defeating them increases. As a result, it is evident that machine learning-based IDSs must be extensively evaluated against AML attacks.

To the best of our knowledge, this is the first study which investigates the behaviour of supervised models against automatically generated AML attacks, as well as the defence of such attacks in the context of ICS. More importantly, this work considers a realistic attacker model and assumptions, as well as a realistic dataset collected from a representative power system testbed. The main contributions of the work presented in this paper are the empirical investigations into:

- generating adversarial samples from a power system dataset
- the behaviour of supervised machine learning algorithms against adversarial samples for intrusion detection in an ICS system
- how adversarial training can support the robustness of such models

The study was designed as follows (see Fig. 1): (1) randomly split the power system dataset into training and testing set, each containing 60% and 40% data points respectively, (2) evaluate a range of supervised machine learning models and identify which are the best performing, (3) generate adversarial samples using the Jacobian-based Saliency map method, (4) evaluate the performance of the trained models in 2 on the generated adversarial samples in 3, (5) include a percentage of adversarial samples from 3 in the training data and re-train and evaluate the models.

The remainder of this paper is structured as follows: Section 2 discusses the relevant work in this research area, Section 3 discusses the power system testbed and the generated dataset which is used to support the experiments in this paper, Section 4 evaluates the performance of a range of supervised classifiers, Section 5 discusses AML and the methodology followed to generate adversarial samples, Section 7 investigates the effectiveness of adversarial training as a defence mechanism, and finally 8 concludes the paper.

2. Related work

There has been a substantial increase in machine learning-based IDSs for a range of ICS systems. Table 1 presents a summary of the existing ICS systems and associated supervised learning approaches to attack detection and classification in these contexts. To date, there has been less focus on AML in this context. Such research has mainly focused on email spam classifiers, malware detection, and very recently there has been interest in AML against network IDSs for traditional networks (e.g. [11–13]).

More specifically, both Nelson et al. [14] and Zhou et al. [15] demonstrate that an adversary can exploit and successfully bypass the machine learning methods employed in spam filters by modifying a small percentage of the original training data. Moreover, Grosse et al. [16] evaluate the robustness of a neural network trained on the DREBIN Android malware dataset. They report that it is possible to confuse the model by perturbing a small amount of the features in the training set. Such an attack is considered to be a white box attack, as to be successful, the adversary needs to have access or knowledge of the dataset and the features it includes. Additionally, Pierazzi et al. [17] evaluated 170K Android apps between 2017 and 2018 to demonstrate the practical feasibility of evading a state-of-the-art malware classifiers. Their results showed that “adversarial-malware as a service” is a realistic threat, as it was possible to automatically generate thousands of realistic and inconspicuous adversarial applications at scale, where on average it took only a few minutes to generate an adversarial app. Furthermore, Hu and Tan [18] proposed a more advanced adversarial technique which uses the concept of Generative Adversarial Networks (GAN) to successfully attack malware classifiers without requiring any knowledge of the data and the system. This is known as a black box attack. Finally, Appruzzese, Colajanni, and Marchetti [19] deploy realistic adversarial attacks against network intrusion detection systems that focus on identifying botnet traffic through machine learning classifiers. The results showed that such attacks are effective.

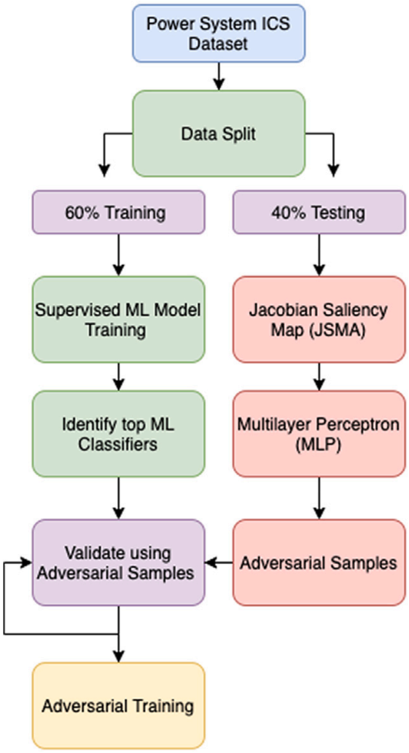
Power System ICS Dataset
Data Split
60% Training
40% Testing
Supervised ML Model Training
Jacobian Saliency Map (JSMA)
Identify top ML Classifiers
Multilayer Perceptron (MLP)
Validate using Adversarial Samples
Adversarial Samples
Adversarial Training

Fig. 1. An overview of the study design.

In the context of ICSs, there exist only a handful of investigations into AML attacks. Specifically, Zizzo et al. [20] showcased a simple AML attack against a Long Short-Term Memory (LSTM) classifier which was applied on an ICS dataset. However, this work is at a preliminary stage as the adversarial samples were generated by manually selecting the feature values to be perturbed. Yaghoubi and Fainekos [21] proposed a gradient-based search approach which was evaluated on a Simulink model of a steam condenser. However, this approach is efficient only against a handful of systems that may specifically employ Recurrent Neural Networks (RNN) with smooth activation functions. Finally, Erba et al. [3] demonstrated two types of real-time evasion attacks, again using Recurrent Neural Network models, and used an autoencoder to generate adversarial samples. Neither of these aforementioned works investigate *defence methods against AML*. Conclusively, it is evident that there is room to investigate AML and the defence against such attacks for current IDSs in ICS systems that are supported by supervised learning. Moreover, as Table 1 shows, Recurrent Neural Networks are yet to gain prominence in attack detection in an ICS context — with algorithms such as Naive Bayes, Random Forest, SVM, and J48 being much more widely used. The experiments, therefore, focus on defending against AML on these methods as the state of the art in ML-driven attack detection methods for ICS.

Table 1
Summary of current work on Intrusion Detection Systems in Industrial Control Systems.

Citation	Publication date	Dataset	Machine learning models
[22]	2019	Power System	Random Forest
[23]	2019	Wind Turbines	SVM
[24]	2019	SCADA Testbed	Long Short Term Memory (RNN)
[25]	2018	Power system (synthetic)	Naive Bayes, Random Forests, SVM
[26]	2018	SWaT	SVM, J48, Random Forest
[27]	2018	Gas Pipeline	SVM, Random Forest
[6]	2018	SCADA Testbed	Random Forest, J48, Logistic Regression, Naive Bayes
[28]	2018	SCADA Testbed	SVM, Decision Tree, and Random Forest
[29]	2018	Power System	SVM, J48, Neural Network
[30]	2018	Wind Turbine	Decision Trees (J48, Random Forest, CART, Ripper, etc.)
[1]	2018	SWaT	1D Convolutional Networks
[31]	2017	SCADA/ICS	J48, Naive Bayes
[32]	2017	SCADA/Modbus	Decision Tree, K-Nearest Neighbour, SVM, OCSVM
[33]	2017	Power Grid, Water Plant, Gas Plant	J48, Random Forest, Naive Bayes, SVM, JRipper + Adaboost
[34]	2017	SCADA Testbed	Decision Tree, Random Forest
[35]	2017	ICS Testbed	Long Short Term Memory (RNN)
[8]	2017	SWaT	Long Short Term Memory (RNN)
[36]	2015	Power System	OneR, Random Forest, Naive Bayes, SVM, JRipper + Adaboost
[37]	2014	SCADA	Naive Bayes, BayesNet, J48
[9]	2014	SCADA Network Traffic	One-Class SVM
[4]	2013	Gas Pipeline	Naive Bayes, Random Forest, SVM, J48, OneR
[10]	2009	ICS Testbed	Neural Network (Error-back propagation and Levenberg–Marquardt)
[5]	2003	SCADA Testbed	Bayesian Network

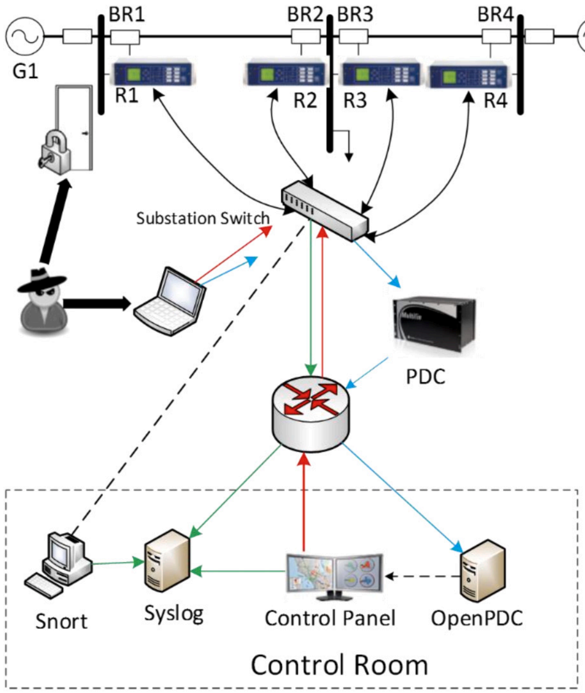

Fig. 2. Power System Framework Testbed used for generating the datasets in which support the experiments herein [39].

3. Industrial control system case study: Power system

Mississippi State University and Oak Ridge National Laboratory implemented a scaled-down version of a power system framework. Although this system is relatively small, it captures the core function and is considered as being a representative example of a larger power system [38]. Fig. 2 illustrates in more detail the power system framework configuration and the components used for generating the datasets in which support the experiments in this paper.

More specifically, the components of the power system as shown in Fig. 2 include:

- G1 and G2 are the main generators.
- R1, R2, R3, and R4 are the Intelligent Electronic Devices (IEDs) responsible for switching the breakers (BR1, BR2, BR3, BR4), which are automatically operated electrical switches designed to protect electrical circuits from damage caused by excess current from an overload or short circuit, on and off.
- Each IED automatically controls one breaker (e.g. R1 controls BR1, R2 controls BR2, etc.)
- The IEDs use a distance protection scheme which trips the breaker on detected faults (whether they are valid or invalid) since they have no internal validation to detect the difference.
- Operators can also manually issue commands to the IEDs to manually trip the breakers. The manual override is used when performing maintenance on the lines or other system components.
- There are also other network monitoring devices connected on the testbed, such as SNORT and Syslog servers.

4. Supervised machine learning

To explore how well supervised classification algorithms can learn to detect cyber attacks in an ICS environment, the performance of supervised machine learning when the corresponding data discussed in Section 4.1 was used to train the classification model and evaluated. The following Sections report the features present in the power systems dataset, as well as describing the methodology behind selecting and training the best performing supervised classifiers.

4.1. Dataset

A dataset containing both benign and malicious data points was generated from the power system testbed by [38]. These data points have been further categorised into three main classes; ‘no event’ instances, ‘natural event’ instances, and ‘attack event’ instances. Both the ‘no event’ and ‘natural event’ instances are grouped to represent benign activity. To generate the malicious data, attacks from 5 scenarios were deployed on the power system. These attacks are described as follows:

(1) **Short-circuit fault**. This is a short in a power line and can occur in various locations along the line. The location is indicated by the percentage range.
(2) **Line maintenance.** One or more relays are disabled on a specific line to do maintenance for that line.

Table 2
Features included as part of the power system dataset.

Feature	Description
PA1–PA3:VH	PA1:VH–PA3:VH Phase A
PM1: V–PM3:V	C Voltage Phase Angle
PA4:IH–PA6:IH	Phase A–C Current Phase Angle
PM4: I–PM6: I	Phase A–C Current Phase Magnitude
PA7:VH–PA9:VH	Pos.–Neg.–Zero Voltage Phase Angle
PM7: V–PM9: V	Pos.–Neg.–Zero Voltage Phase Magnitude
PA10:VH–PA12:VH	Pos.– Neg.–Zero Current Phase Angle
PM10: V - PM1	Pos.–Neg.–Zero Current Phase Magnitude
F	Frequency for relays
DF	Frequency Delta (dF/dt) for relays
PA:Z	Appearance Impedance for relays
PA:ZH	Appearance Impedance Angle for relays
S	Status Flag for relays

(3) **Remote tripping command injection attack**. This is an attack that sends a command to a relay which causes a breaker to open. It can only be done once an attacker has penetrated outside defences.
(4) **Relay setting change attack**. Relays are configured with a distance protection scheme. The attacker changes the setting to disable the relay function so that the relay will not trip for a valid fault or a valid command.
(5) **Data injection attack**. A valid fault is imitated by changing values to parameters such as the current, voltage, and sequence components. This attack aims to blind the operator and causes a blackout.

The final dataset consisted of 55,663 malicious and 22,714 benign data points.

4.2. Feature selection

To perform machine learning classification experiments, it is essential to identify which attributes best describe the dataset. In this case, the data points within the power system dataset contain attributes associated with synchrophasor measurements and basic network security mechanisms. A synchrophasor measurement unit is a device which measures the electrical waves on an electricity grid, using a common time source for synchronisation. The dataset contains a total of 128 features [39]. These features are described in more detail as follows:

- 29 types of measurements from each synchrophasor measurement unit. In this specific power system testbed, there are 4 PMUs. Therefore, the dataset contains a total of 116 synchrophasor measurement columns.
- 12 types of measurements of control panel logs, snort alerts, and relay logs of the 4 synchrophasor measurement unit and relay.

Table 2 summarises the features included in the dataset, as well as their corresponding descriptions. More specifically, the index of each feature is in the form of “R#-Signal Reference”. The “R ‘#’ ” specifies the type of measurement from the synchrophasor measurement unit. For instance, “R1-PA1:VH” corresponds to the “Phase A voltage phase angle” measured by “PMU R1”.

4.3. Model training

To explore how well supervised machine learning algorithms can detect cyber attacks in an ICS environment, the corresponding power system dataset was used to evaluate a range of state-of-the-art classifiers.

The “no free lunch” theorem suggests that there is no universally best learning algorithm [40]. In other words, the choice of an appropriate algorithm should be based on its performance for that particular problem and the properties of data that characterise the problem. In this case, a variety of classifiers distributed as part of Weka [41] were evaluated using 10-fold cross-validation using their default hyper-parameters.

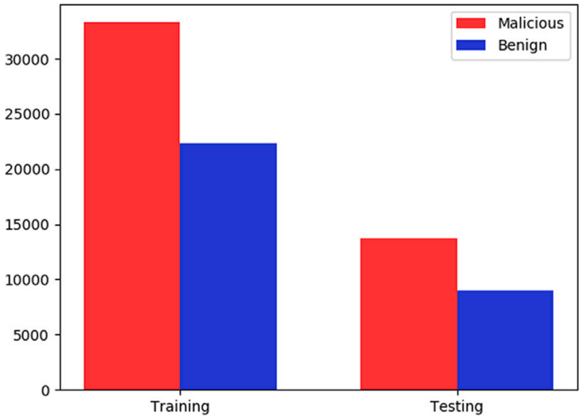

Fig. 3. Distribution of data points across both training and testing datasets.

To conform to other comparable IDSs in ICS systems in Table 1, the classifiers were also selected based on their ability to support a high-dimensional feature space. The classifiers included:

- Generative models that consider conditional dependencies in the dataset or assume conditional independence (e.g. Bayesian Network, Naive Bayes).
- Discriminative models that aim to maximise information gain or directly maps data to their respective classes without modelling any underlying probability or structure of the data (e.g. J48 Decision Tree, Support Vector Machine).

To support classification experiments, a random subset of approximately 60% of the dataset described in Section 4.1 was selected for training, with the remaining 40% selected for testing. Fig. 3 reports the distributions of data points across the target values in both the training and testing datasets.

An uneven balance of class labels across the training dataset has the potential to negatively affect or may bias classification performance. Given the significant uneven balance across the dataset, the class balancing filter available in Weka was applied to balance the distribution of classes within the sample. In this case, the training dataset was balanced so that there were 13,725 samples of both malicious and benign data points. In order to generate a representative testing dataset and comply with relevant work [42,43], where the benign samples outnumber the malicious ones, a random sample of 40% of the malicious packets was selected. Subsequently, the final distribution of class labels in the testing dataset was 3560 malicious and 8989 benign data points.

Previous works which have used a very small sample of this power system dataset to support their classification experiments have shown that the ensemble classifier which combines both the Adaboost and JRipper models was found to be the best performing [44]. Conversely, the classifiers with the highest performances were Random Forest and Weka’s implementation of the J48 decision tree method with no pruning respectively (see Table 3).

5. Adversarial machine learning

To reiterate, AML aims to automatically introduce perturbations to the unseen data points to confuse the pre-trained model. The following sections introduce the types of AML attacks, as well as the methods used to automatically generate adversarial samples.

Table 3
Weighted average results following cross-validation (P = Precision, R = Recall, F = F1-score)

Classifier	P	R	F	Time (s)
Random Forest	0.94	0.94	0.94	25.21
J48	0.87	0.87	0.87	19.80

5.1. Adversarial attack types

Depending on the phase and aspect of the machine learning model that is being targeted, AML attacks can be described in terms of four primary vectors: [13,45]:

- The **Influence** of an attack's affects the classifier's decision. Attacks can be further categorised as causative attacks, which occur during the learning phase (poison attacks), or exploratory attacks, which target the trained model during the testing phases (evasion attacks).
- **Security Violations** affect either the integrity of the model when the adversarial samples cause misclassifications, or when the high rate of misclassifications causes the model to become unusable.
- **Specificity** refers to targeted attacks, where the adversarial samples aim to target a specific target value, or indiscriminate attacks, where the samples do not target a specific target value.
- **Privacy** refers to attacks where the adversary's goal is to extract information from the classifier.

Papernot et al. [46] further categorise adversarial attacks based on:

- Their **complexity**. The consequences of such attacks can range from slightly reducing the confidence of a model's predictions to causing it to misclassify all unseen data points.
- The **knowledge** an adversary may have. A *white box* attack refers to when an attacker has useful knowledge related to the learning model, such as its architecture, the network's traffic it reads, and the features used to support its training. It is considered as being a *black box* attack when an adversary has no information about the internal workings of the target model.

5.2. Attacker model

In this work, we consider an insider threat attacker who has admin access privileges to the local plant communication network systems (chief network engineer). Insider threats are one of the most underestimated but rather critical threats for ICSs [47]. More specifically, as insiders reside behind the enterprise-level security defence mechanisms and often have privileged access to the network, detecting and preventing insider threats is a complex and challenging problem [48].

According to the German Federal Office for Information Security [49], insider threats include those with potentially privileged access to IT components, services, installations, documents, or any other critical information about the infrastructure and its components. In particular, the following groups are considered as insider threats:

- A person with direct physical access to control systems (e.g. operators, engineers).
- A person with privileged rights (e.g. administrators).
- People with indirect access (e.g. even to the office network or administration buildings).
- External service providers (e.g. maintenance or software development), suppliers, etc.

Such adversary can deploy a range of attacks such as:

- Theft/modification of sensitive information (data leakage) through access to file servers, historians, data storage media. The main motive for this includes industrial espionage and whistleblowing.
- Social Engineering to prepare follow-up attacks. This can occur by determining weak employees, understanding the industrial processes, mapping the IT infrastructure and more.
- Sabotage the company. This is mainly motivated by political or economic interests. This may include manipulating control components or implanting malware or spyware.

In the power system scenario discussed herein and given the capabilities of the attacker as discussed above, it is assumed that the adversary interested in utilising AML would have access to both the dataset and its features. Additionally, given the position of the adversary (chief network engineer), it is assumed that they know the features that the IDS is utilising for the classification; however, they do not know the exact algorithm configuration of the detector. This is due to the obscurity of the exact product specifications that often accompanies enterprise software. The goal of the attacker is, therefore, to identify how to bypass the IDS to (a) cause further damage in the future by deploying further attacks or (b) give this information to competitors so that they can harm the organisation. It is also important to highlight that it is assumed that there are no measures in place to protect the leaked information and the ICS from AML attacks, such as [50,51]. Due to the nature of the knowledge obtained by the adversary, such an AML attack is classified as being a grey box attack.

5.3. Adversarial sample generation methods

There are various methods by which adversarial samples can be generated. Such methods vary in complexity, the speed of their generation, and their performance. An unsophisticated approach towards crafting such samples is to manually perturb the input data points. However, manual perturbations of large datasets are tedious to generate and may be less accurate. More sophisticated approaches include automatically analysing and identifying features which best discriminate between target values. Such features are discretely perturbed so that they reflect similar values to those which represent target values other than their own. Two of the most popular 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. [52] and Papernot et al. [46] respectively.

Both methods rely on the methodology, that when adding small perturbations (δ) to the original sample (X), the resulting sample (X*) can exhibit adversarial characteristics (X* = X + δ) [11] in that X* is now classified differently by the targeted model. Moreover, both methods are also usually applied by using a pre-trained MLP as the underlying model for the adversarial sample generation.

The FGSM method aims to target each of the features of the input data by adding a specified amount of perturbation. The perturbation noise is computed by the gradient of the cost function J with respect to the input data. Let θ represent the model parameters, x are the inputs to the model, y are the labels associated with the input data, ϵ is a value which represents the extent of the noise to be applied, and $J(\theta, x, y)$ is the cost function used to train the targeted neural network.

$$x^* = x + \epsilon\,\mathrm{sign}\left(\nabla_x J(\theta, x, y)\right)$$

On the other hand, the JSMA method generates perturbations using saliency maps and it requires three steps [11]. Initially, the Jacobian of the overall neural network function F in respect to the input X is calculated:

$$J_F = \frac{\delta F(X)}{\delta X}$$

Secondly, the Jacobian is used in order to calculate a Saliency map. A saliency map identifies which features of the input data are the

Table 4
An example of how features are perturbed using JSMA.

Dataset	R1-PA1:VH	R1-PM1:V	R1-PM4:I	R3-PM6:I	R3-PA8:VH
Original test data	0.7645	0.8710	0.1756	0.0261	0.5027
$\theta = 0.1, \gamma = 0.1$	*0.7650*	0.8710	0.1756	0.0261	*0.5030*
$\theta = 0.5, \gamma = 0.5$	*0.7650*	0.8710	0.1756	0.0261	*0.5030*
$\theta = 0.9, \gamma = 0.9$	*1.0000*	*0.8770*	0.1756	0.0261	*0.5070*

most relevant to the model decision being one class or another. These features, if altered, are most likely affect the classification of the target values. More specifically, an initial percentage of features (θ) is chosen to be perturbed by a (γ) amount of noise. Thirdly, the model establishes whether the added noise has caused the targeted model to misclassify or not. If the noise has not affected the model's performance, another set of features is selected and a new iteration occurs until a saliency map appears which can be used to generate an adversarial sample. For the adversarial sample generation herein, we utilised the JSMA algorithm as described in [46].

Given that the JSMA method may take a few iterations to generate adversarial samples, the FGSM is computationally faster [46]. More specifically its time complexity is O(N). However, as opposed to FGSM which alters each feature, JSMA is a more complex and elaborate approach which represents more realistic attacks as it progressively alters a small percentage of features at a time. The complexity of JSMA heavily depends on the number of input features. The larger the feature space, the more iterations it requires to establish whether the approach is successful in generating adversarial samples which affect a model's performance. Nevertheless, this approach allows for more realistic and finer-grained AML attacks, as adversaries can define both the percentage of features to perturb and the amount of perturbation to include when generating the adversarial samples.

This work presents the use of JSMA in a grey box attack, in which the attacker has no knowledge of the target model but has access to the full dataset and knowledge of features. Despite not knowing the target model, we can approximate samples that will cause the target model to misclassify using another model due to the transferability of adversarial samples across machine learning models [53].

In this case, the adversarial samples used in the experiments herein were generated using the JSMA method. A pre-trained MLP was used as the underlying model for the generation. The code implementation used to create the adversarial data was based on the CleverHans project [46]. To illustrate, Table 4 shows the transformation of the features of a malicious data point using the JSMA method using different variants of θ and γ. Such examples demonstrate that the higher the value of θ, the more intense the perturbation of the feature value. This is shown for the R1-PA1:VH feature, where its original value increases from 0.7645 to 0.7650 when $\theta = 0.1$ and 0.5, and to 1.000 when $\theta = 0.9$. Similarly, the higher the value of γ, the more features are perturbed. This is shown in the R1-PM1:V feature, where its value is perturbed only when $\gamma = 0.9$. The attacker's ultimate goal is to find the minimum value of θ and γ to decrease the performance of the classifier, without significantly modifying feature values.

6. Evaluating supervised models on adversarial samples

Both the trained Random Forest and J48 models presented in Section 4.3 were first evaluated against the original testing dataset. The F1-scores achieved by both classifiers were 0.61 and 0.60 respectively. The confusion matrix in Table 5 shows how the predicted classes for each data point in the original testing dataset compare against the actual ones. In comparison to the Random Forest model, J48 demonstrated a higher percentage of correct predictions, thus less often misclassifying the data points.

Table 5
Confusion matrices for the original test set (Benign = 0, Malicious = 1)

		Predicted				Predicted	
		0	**1**			**0**	**1**
Actual	**0**	5556	3433	Actual	**0**	5253	3736
	1	1666	1930		**1**	1583	2013
	Random Forest				J48		

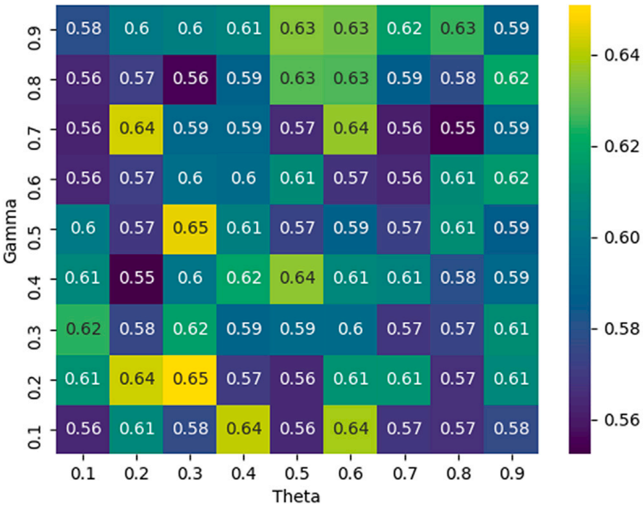

Fig. 4. Random Forest classification performance (F1-score) on adversarial samples generated using JSMA.

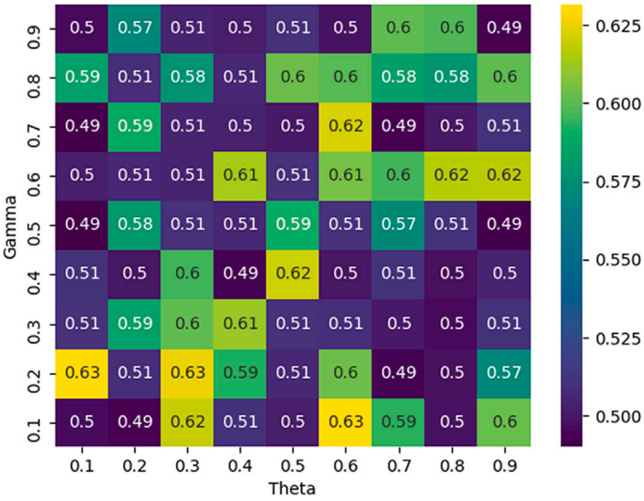

Fig. 5. J48 classification performance (F1-score) on adversarial samples generated using JSMA.

To explore how different combinations of the JSMA parameters affect the performance of the trained classifiers, adversarial samples were generated from all malicious data points present in the testing data by using a range of combinations of θ and γ. The adversarial samples were joined with the benign testing data points and subsequently presented to the trained models. Figs. 4 and 5 report the overall weighted-averaged Recall for all adversarial combinations of JSMA's θ and γ parameters.

In comparison to Random Forest, the J48 model achieved a decrease in Recall across the majority of the θ and γ parameters. This may indicate that J48 may be more sensitive, subsequently misclassifying malicious data points as benign. However, when $\theta = 0.1$, $\gamma = 0.2$, $\theta = 0.3$, $\gamma = 0.2$ and $\theta = 0.6$, $\gamma = 0.1$, the model achieves a higher F1-score of 0.63 (an increase of 3 percentage points). This may indicate that the

Table 6
Confusion matrices after applying Random Forest to adversarial testing samples (Benign = 0, Malicious = 1)

		Predicted				Predicted	
		0	1			0	1
Actual	0	5253	3736	Actual	0	5253	3736
	1	2390	1206		1	1390	2206
	θ = 0.2, γ = 0.4				θ = 0.5, γ = 0.9		

Table 7
Confusion matrices after applying J48 to adversarial testing samples (Benign = 0, Malicious = 1)

		Predicted				Predicted	
		0	1			0	1
Actual	0	5556	3433	Actual	0	5556	3433
	1	2612	984		1	1141	2455
	θ = 0.1, γ = 0.5				θ = 0.6, γ = 0.1		

generation of some adversarial samples has made such data points more distinct in discriminating between the target values.

Conversely, the classification performance of the Random Forest model achieved an increase in F1-score in many of θ and γ combinations. This may indicate that Random Forest may be a more robust classifier in discriminating between malicious and benign data points correctly. However, when $\theta = 0.2$, $\gamma = 0.4$, and $\theta = 0.8$, $\gamma = 0.7$, the model's classification performance decreases by 6 percentage points (F1-score = 0.55). Based on the dataset used in the experiments presented in this paper, such combinations would be the optimal parameter an adversary would use to successfully reduce the accuracy of a machine learning-based IDS, subsequently diverting malicious data points.

These findings demonstrate the importance of parameter tuning in applying JSMA for generating adversarial examples. The JSMA model is likely to be more robust under white/grey box conditions as it was designed but these results indicate that with careful parameter tuning, this approach can be adapted to work under black-box conditions. Although the F1-score increases in some instances, the attacker is primarily interested in their malicious data points being classified as benign, such that an increase in Recall is not necessarily undesirable from the attacker's perspective.

The confusion matrices in Tables 7 and 6 provide a better insight into the performance of the classifiers across the experiments. In comparison to the original classification distributions in Table 5, both classifiers demonstrate a significant increase in false positives. That is, data points with an actual target value of malicious have been misclassified as being benign.

7. Defending adversarial machine learning

A few methods of defending AML attacks have been proposed in the literature. Two of the most popular techniques include adversarial training and adversarial sample detection. The former has been explored in the field of visual computing, where Goodfellow et al. [54] demonstrated that re-training the neural network on a dataset containing both the original and adversarial samples significantly improves its efficiency against adversarial samples. The latter technique involves the implementation of mechanisms that are capable of detecting the presence of such samples using direct classification, neural network uncertainty, or input processing [20]. However, these detection mechanisms have been found to be weak in defending AML [20,55].

Subsequently, in this paper, the robustness of supervised machine learning classifiers against AML is further evaluated using adversarial training. In this case, to avoid bias and by drawing inspiration from the 10-fold cross-validation method [56], 10 random samples of 10% of the adversarial data points in the testing dataset which significantly decreased the model's performance (Random Forest: $\theta = 0.2$, $\gamma = 0.4$ and J48: $\theta = 0.1$, $\gamma = 0.5$) were included in the original training dataset. Subsequently, the average F1-score across the 10 models was calculated and is reported in Figs. 6 and 7. The adversarial datasets produced using the selected θ and γ combinations were omitted from the evaluations as they were not comparable and are thus represented as black boxes.

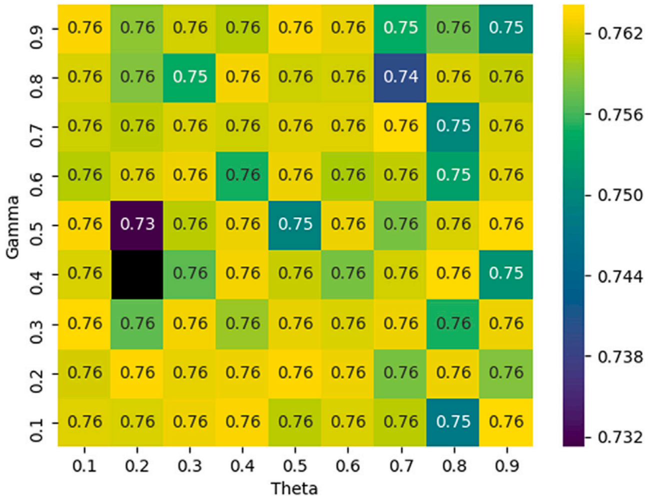

Fig. 6. Random Forest classification performance (F1-score) following adversarial training ($\theta = 0.2$, $\gamma = 0.4$).

The experiments described in Sections 4.3 and 6 were repeated by retraining the models with the newly generated training data and applying such models on all unseen adversarial samples. Both the Random Forest and J48 models achieved average cross-validation F1-scores of 0.94 and 0.89 respectively.

Figs. 6 and 7 report the overall weighted-averaged F1-scores for all adversarial combinations of JSMA's θ and γ parameters following adversarial training. The results demonstrated that for both classifiers, including adversarial samples in the training data increased the classification performances for several adversarial combinations of JSMA. For example, when $\theta = 0.1$, $\gamma = 0.5$ and $\theta = 0.9$, $\gamma = 0.9$, Random Forest and J48 achieved F1-scores of 0.76 and 0.80 respectively, an increase of 11 and 17 percentage points in comparison to the highest classification performances reported in Figs. 4 and 5.

The classification performances demonstrated by the Random Forest model achieves a greater overall increase in comparison to the J48 model. That is, for Random Forest, the classification performance for all combinations were improved. Whereas for J48, only around 30% of the classification performances increased significantly. This may imply that Random Forest is a more robust model towards classifying adversarial samples of all combinations of JSMA's θ and γ parameters. This is intuitive given Strauss et al.'s [7] demonstration that ensemble machine learning algorithms are more robust against adversarial techniques and Random Forests are ensembles of decision trees (such as J48).

8. Conclusion

Due to their effectiveness and flexibility, machine learning-based IDSs are now recognised as fundamental tools for detecting cyber attacks in ICS systems. Nevertheless, such systems are vulnerable to attacks that may severely undermine or mislead their capabilities, commonly known as AML. Such attacks may have severe consequences in ICS infrastructures, as adversaries could potentially modify malicious data points to bypass the IDS, causing delayed attack detection and

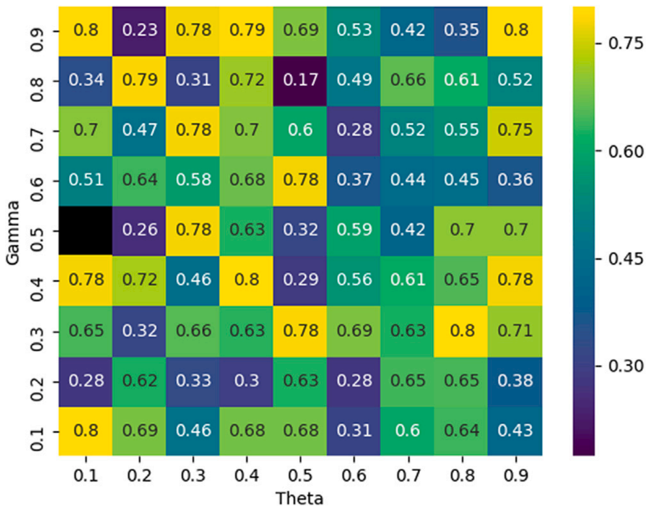

Fig. 7. J48 classification performance (F1-score) following adversarial training (θ = 0.1, γ = 0.5).

extensive damages. Thus, it is evident that understanding the applicability of these attacks in ICS systems is necessary to develop more robust machine learning-based IDSs.

This paper explores how adversarial learning can be used to target supervised models by generating adversarial samples and exploring classification behaviours. To support the experiments presented herein, an authentic power system dataset was used to train and test widely used supervised machine learning classifiers. Moreover, this work considers a realistic attacker model and assumptions. The testing data was presented to a JSMA to generate adversarial samples with a range of combinations that affect the amount of noise and the number of features to perturb. Such samples were evaluated against two of the best performing classifiers, Random Forest and J48. Overall, the classification performance for both models decreased by 6 and 11 percentage points when adversarial samples were present.

The analysis also includes the exploration of how such samples can support the robustness of supervised models using adversarial training. A random sample of 10% of the generated adversarial data points were included in the original training dataset. The models were retrained and applied on all unseen adversarial samples. Overall, the classification performance of the Random Forest model reported an increase across all JSMA parameters in comparison to the J48 model. This demonstrates that Random Forest is a more robust model towards classifying adversarial samples of all combinations of JSMA parameters on the given dataset.

9. Future work

Although the experiments described in this paper have demonstrated that adversarial samples can successfully be generated using JSMA and affect the classification performance of state-of-the-art supervised models, it is important to note that there are several other methods of generating such samples to consider (e.g. Iterative Gradient Sign, Carlini Wagner, Generative Adversarial Networks). In this case, as part of future work, this study can be extended further to include different models as a source for generating adversarial samples. Moreover, AML should be further investigated against other models such as LSTMs.

Finally, the robustness of the supervised models was demonstrated using adversarial training. It is also important to note that this method may not always be sufficient as it is difficult to anticipate all possible types of adversarial machine learning attacks against a given system. Therefore, there is a need to investigate other possible defence mechanisms.

CRediT authorship contribution statement

Eirini Anthi: Conceptualisation, Methodology, Validation, Formal analysis, Investigation, Writing - original draft, Writing - review & editing. **Lowri Williams:** Methodology, Validation, Formal analysis, Software, Writing - original draft, Writing - review & editing. **Matilda Rhode:** Methodology, Validation, Software. **Pete Burnap:** Conceptualisation, Resources, Supervision. **Adam Wedgbury:** Resources, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This project was part-funded by: Airbus Endeavr, grant "SCADA Cybersecurity Lifecycle 2"; and the Engineering and Physical Sciences Research Council (EPSRC), grant "New Industrial Systems: Chatty Factories", REF EP/R021031/1.

References

[1] Kravchik M, Shabtai A. Detecting cyber attacks in industrial control systems using convolutional neural networks. In: 2018 workshop on cyber-physical systems security and privacy. ACM; 2018, p. 72–83.
[2] Ashibani Y, Mahmoud QH. Cyber physical systems security: Analysis, challenges and solutions. Comput Secur 2017;68:81–97.
[3] Erba A, Taormina R, Galelli S, Pogliani M, Carminati M, Zanero S, et al. Real-time evasion attacks with physical constraints on deep learning-based anomaly detectors in industrial control systems. 2019, arXiv preprint arXiv:1907.07487.
[4] Beaver JM, Borges-Hink RC, Buckner MA. An evaluation of machine learning methods to detect malicious scada communications. In: 2013 12th international conference on machine learning and applications, vol. 2. IEEE; 2013, p. 54–9.
[5] Bigham J, Gamez D, Lu N. Safeguarding SCADA systems with anomaly detection. In: International workshop on mathematical methods, models, and architectures for computer network security. Springer; 2003, p. 171–82.
[6] Teixeira MA, Salman T, Zolanvari M, Jain R, Meskin N, Samaka M. SCADA system testbed for cybersecurity research using machine learning approach. Future Internet 2018;10(8):76.
[7] Strauss T, Hanselmann M, Junginger A, Ulmer H. Ensemble methods as a defense to adversarial perturbations against deep neural networks. 2017, arXiv preprint arXiv:1709.03423.
[8] Goh J, Adepu S, Tan M, Lee ZS. Anomaly detection in cyber physical systems using recurrent neural networks. In: 2017 IEEE 18th international symposium on high assurance systems engineering. IEEE; 2017, p. 140–5.
[9] Maglaras LA, Jiang J. Intrusion detection in SCADA systems using machine learning techniques. In: 2014 science and information conference. IEEE; 2014, p. 626–31.
[10] Linda O, Vollmer T, Manic M. Neural network based intrusion detection system for critical infrastructures. In: 2009 International joint conference on neural networks. IEEE; 2009, p. 1827–34.
[11] Rigaki M. Adversarial deep learning against intrusion detection classifiers. 2017.
[12] Biggio B, Fumera G, Roli F. Multiple classifier systems under attack. In: International workshop on multiple classifier systems. Springer; 2010, p. 74–83.
[13] Huang L, Joseph AD, Nelson B, Rubinstein BI, Tygar JD. Adversarial machine learning. In: Proceedings of the 4th ACM workshop on security and artificial intelligence. ACM; 2011, p. 43–58.
[14] Nelson B, Barreno M, Chi FJ, Joseph AD, Rubinstein BI, Saini U, et al. Exploiting machine learning to subvert your spam filter. LEET 2008;8:1–9.
[15] Zhou Y, Kantarcioglu M, Thuraisingham B, Xi B. Adversarial support vector machine learning. In: 18th ACM SIGKDD international conference on knowledge discovery and data mining. 2012. p. 1059–67.
[16] 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.
[17] Pierazzi F, Pendlebury F, Cortellazzi J, Cavallaro L. Intriguing properties of adversarial ML attacks in the problem space. In: 2020 IEEE symposium on security and privacy. 2020. p. 1332–49.
[18] Hu W, Tan Y. Generating adversarial malware examples for black-box attacks based on gan. 2017, arXiv preprint arXiv:1702.05983.

[19] Apruzzese G, Colajanni M, Marchetti M. Evaluating the effectiveness of adversarial attacks against botnet detectors. In: 2019 IEEE 18th international symposium on network computing and applications. 2019. p. 1–8.

[20] Zizzo G, Hankin C, Maffeis S, Jones K. Adversarial machine learning beyond the image domain. In: 2019 56th ACM/IEEE conference on design automation. IEEE; 2019, p. 1–4.

[21] Yaghoubi S, Fainekos G. Gray-box adversarial testing for control systems with machine learning components. In: 22nd ACM international conference on hybrid systems: Computation and control. 2019. p. 179–84.

[22] Wang D, Wang X, Zhang Y, Jin L. Detection of power grid disturbances and cyber-attacks based on machine learning. J. Inf. Secur. Appl. 2019;46:42–52.

[23] Hoxha E, Vidal Seguí Y, Pozo Montero F. Supervised classification with SCADA data for condition monitoring of wind turbines. In: 9th ECCOMAS thematic conference on smart structures and materials. 2019. p. 263–73.

[24] Gao J, Gan L, Buschendorf F, Zhang L, Liu H, Li P, et al. LSTM for SCADA intrusion detection. In: 2019 IEEE pacific rim conference on communications, computers and signal processing. IEEE; 2019, p. 1–5.

[25] Anton SD, Kanoor S, Fraunholz D, Schotten HD. Evaluation of machine learning-based anomaly detection algorithms on an industrial modbus/tcp data set. In: 13th international conference on availability, reliability and security. 2018. p. 1–9.

[26] Robles-Durazno A, Moradpoor N, McWhinnie J, Russell G. A supervised energy monitoring-based machine learning approach for anomaly detection in a clean water supply system. In: 2018 international conference on cyber security and protection of digital services. IEEE; 2018, p. 1–8.

[27] Perez RL, Adamsky F, Soua R, Engel T. Machine learning for reliable network attack detection in scada systems. In: 2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering. IEEE; 2018, p. 633–8.

[28] Frazão I, Abreu PH, Cruz T, Araújo H, Simões P. Denial of service attacks: detecting the frailties of machine learning algorithms in the classification process. In: International conference on critical information infrastructures security. Springer; 2018, p. 230–5.

[29] Lahza H, Radke K, Foo E. Applying domain-specific knowledge to construct features for detecting distributed denial-of-service attacks on the GOOSE and MMS protocols. Int J Crit Infrastruct Prot 2018;20:48–67.

[30] Abdallah I, Dertimanis V, Mylonas H, Tatsis K, Chatzi E, Dervilis N, et al. Fault diagnosis of wind turbine structures using decision tree learning algorithms with big data. Saf Reliab–Safe Soc Chang World 2018;3053–61.

[31] Ullah I, Mahmoud QH. A hybrid model for anomaly-based intrusion detection in SCADA networks. In: 2017 IEEE international conference on big data. IEEE; 2017, p. 2160–7.

[32] Qu H, Qin J, Liu W, Chen H. Instruction detection in SCADA/Modbus network based on machine learning. In: International conference on machine learning and intelligent communications. Springer; 2017, p. 437–54.

[33] Yeckle J, Abdelwahed S. An evaluation of selection method in the classification of scada datasets based on the characteristics of the data and priority of performance. In: International conference on compute and data analysis. 2017. p. 98–103.

[34] Siddavatam IA, Satish S, Mahesh W, Kazi F. An ensemble learning for anomaly identification in SCADA system. In: 2017 7th international conference on power systems. IEEE; 2017, p. 457–62.

[35] Feng C, Li T, Chana D. Multi-level anomaly detection in industrial control systems via package signatures and lstm networks. In: 2017 47th annual IEEE/IFIP international conference on dependable systems and networks. IEEE; 2017, p. 261–72.

[36] Morris TH, Thornton Z, Turnipseed I. Industrial control system simulation and data logging for intrusion detection system research. In: 7th annual southeastern cyber security summit. 2015. p. 3–4.

[37] Werling JR. Behavioral profiling of SCADA network traffic using machine learning algorithms. Tech. rep., Air Force Institute Of Technology; 2014.

[38] Pan S, Morris T, Adhikari U. Classification of disturbances and cyber-attacks in power systems using heterogeneous time-synchronized data. IEEE Trans Ind Inf 2015;11(3):650–62.

[39] Powersystem_dataset_readme.pdf. 2020, [Accessed 18 March 2020].

[40] Wolpert DH. The supervised learning no-free-lunch theorems. In: Soft computing and industry. Springer; 2002, p. 25–42.

[41] Weka 3 - data mining with open source machine learning software in java. 2018, https://www.cs.waikato.ac.nz/ml/weka/. [Accessed 03 June 2018].

[42] Stevanovic M, Pedersen JM. An efficient flow-based botnet detection using supervised machine learning. In: 2014 international conference on computing, networking and communications. IEEE; 2014, p. 797–801.

[43] Kirubavathi G, Anitha R. Botnet detection via mining of traffic flow characteristics. Comput Electr Eng 2016;50:91–101.

[44] Hink RCB, Beaver JM, Buckner MA, Morris T, Adhikari U, Pan S. Machine learning for power system disturbance and cyber-attack discrimination. In: 2014 7th international symposium on resilient control systems. IEEE; 2014, p. 1–8.

[45] Barreno M, Nelson B, Sears R, Joseph AD, Tygar JD. Can machine learning be secure?. In: Proceedings of the 2006 ACM symposium on information, computer and communications security. ACM; 2006, p. 16–25.

[46] 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. IEEE; 2016, p. 372–87.

[47] Gollmann D. From insider threats to business processes that are secure-by-design. In: INCoS. Citeseer; 2011, p. 627.

[48] Liu L, De Vel O, Han Q, Zhang J, Xiang Y. Detecting and preventing cyber insider threats: A survey. IEEE Commun Surv Tutor 2018;20(2):1397–417.

[49] Industrial control system security - insider threat. 2020, https://www.allianz-fuer-cybersicherheit.de. [Accessed 02 September 2020].

[50] Apruzzese G, Colajanni M, Ferretti L, Marchetti M. Addressing adversarial attacks against security systems based on machine learning. In: 2019 11th international conference on cyber conflict, vol. 900. IEEE; 2019, p. 1–18.

[51] Martins N, Cruz JM, Cruz T, Abreu PH. Adversarial machine learning applied to intrusion and malware scenarios: a systematic review. IEEE Access 2020;8:35403–19.

[52] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. In: Advances in neural information processing systems. 2014, p. 2672–80.

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

[54] Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. 2014, arXiv preprint arXiv:1412.6572.

[55] Athalye A, Carlini N, Wagner D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In: International conference on machine learning. 2018. p. 274–83.

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