A Trustworthy Learning Technique for Securing Industrial Internet of Things Systems

Osama Maher[1], and Elena Sitnikova[2]
[1]Faculty of Engineering, Fayoum University, Egypt
[2] University of New South Wales, Canberra, ACT 2600, Australia
Emails: eng.osamamaher21@gmail.com, e.sitnikova@adfa.edu.au

Abstract

Since the Industrial Internet of Things (IIoT) networks comprise heterogeneous manufacturing and technological devices and services, discovering advanced cyber threats is an arduous and risk-prone process. Cyber-attack detection techniques have been recently emerged to understand the process of obtaining knowledge about cyber threats to collect evidence. These techniques have broadly employed for identifying malicious events of cyber threats to protect organizations' assets. The main limitation of these systems is that they are not able to discover and interpret new attack activities. This paper proposes a new adversarial deep learning for discovering adversarial attacks in IIoT networks. Evaluation of correlation reduction has been used as a means of feature selection for reducing the impact of data poisoning attacks on the subsequent deep learning techniques. Feed Forward Deep Neural Networks have been developed using across various parameter permutations, at differing rates of data poisoning, to develop a robust deep learning architecture. The results of the proposed technique have been compared with previously developed deep learning models, proving the increased robustness of the new deep learning architectures across the ToN_IoT datasets.

Keywords: Adversarial deep learning; adversarial attacks; Industrial Internet of Things

1. Introduction

As the proliferation of Internet of Things-based Maritime Transportation Systems (IoT-MTS) continues to characterise the modern technological environment, billions of IoT devices are connecting to networks each year [1]. These devices cumulatively generate vast amounts of data at the edge of networks, which must be validated to ensure the integrity of the broader network and the security of all connected systems [2]. Intrusion Detection Systems that analyse these vast amounts of data commonly use machine and deep learning techniques as a foundation [3] to defend against cyber attackers seeking to infiltrate secure systems. These deep learning systems are susceptible to adversarial techniques, specifically data poisoning in the context of this research.

Cyber-attacks are categorised through various approaches. In this paper, the focus will be on their uses and their effects on the underlying systems. When referring to the uses of attacks, terms used will be congruent with the CIA triad. The CIA triad is the foundation of information security [4] and stands for *Confidentiality*, *Integrity* and *Availability*. *Confidentiality* refers to the safeguard of confidential information within a network and is ensured by network security protocols, authentication services and data encryption. *Integrity* within a system ensures the correctness and completeness of the information within the system. It also refers to integrity within online services, ensuring users are communicating with the desired recipient. Integrity is ensured by firewalls and intrusion detection systems. *Availability* concerns the guarantee for system users, that information and services will be accessible when needed and in a timely manner. Availability is ensured through network security and authentication.

DOI: 10.5281/zenodo.5226915
Received: January 22, 2021 Accepted: June 19, 2021

Shallow and deep learning models are not robust to data poisoning attacks in their base state. As such, with the increasing threat of adversarial machine learning [5], and specifically data poisoning attacks, it is imperative that robust architectures are developed to increase the overall reliability of machine learning models. The focus is to develop a Feed Forward Deep Learning architecture that is more robust than those seen in [5], [6]. Through the implementation of effective feature selection and parameter evaluation, two deep learning models are developed (one for each benchmark dataset), that show increased performance under the effects of label manipulation. This paper explores the impacts of correlation reduction as a means of feature selection, and the influence of parameter values to the robustness of Feed Forward Deep Learning models to data poisoning attacks. The models produced throughout this process are analysed and compared to determine an optimal correlation rate at each level of poisoning, across the two benchmark datasets. This is then compared to the base deep learning models described that were produced to demonstrate the increased robustness at all rates of data poisoning. The key contributions of this paper can be summarised as follows:

- Evaluation of correlation reduction as a means of feature selection, to reduce the impact of data poisoning attacks on the subsequent deep learning models.
- Evaluation and comparison of Feed Forward Deep Neural Networks across various parameter permutations, at differing rates of data poisoning, to develop a robust deep learning architecture.
- Comparison of new robust models with previously developed base models, proving the increased robustness of the new deep learning architectures across both benchmark ToN_IoT datasets.

A. Background Related Work

Machine Learning algorithms gain insights and make decisions from large-scale datasets. As these algorithms are becoming increasingly popular in business, military and other real-world use-cases, the requirement for the security of these systems and privacy of their data become proportionately more significant. Due to the weaknesses of machine learning systems, the field of Adversarial Machine Learning (AML) emerged to study the vulnerabilities that machine learning systems face in an adversarial environment [5]. Integrity attacks allow âadversarial inputsâ to be classified as normal data and cause the filter to pass through a false negative.

This taxonomy allows broad categorisation of attacks in the adversarial environment, however, more detail is required to describe individual attacks accurately. For example, Kurakin et al. [7] published work titled *'Adversarial Machine Learning at Scale'* in 2016 detailing the effectiveness of a variety of existing methods for the development of adversarial examples and their application in adversarial training. Adversarial training is the process of injecting small volumes of adversarial examples into the training data to increase the robustness of the model in an adversarial environment [8]. Carlini and Wagner [9] published *'Towards Evaluating the Robustness of Neural Networks'* in 2017, in which they detail the vulnerabilities of neural networks to adversarial examples. As they explain, *âThe existence of adversarial examples limits the areas in which deep learning can be applied [9].â* This limitation affects the trustworthiness of machine learning and deep neural networks and their application in security systems. To continue the development of robust models, Carlini and Wagner developed three attacks to be used to evaluate the efficacy of future models [9].

a) Attacks against Machine Learning Models:

Within cybersecurity systems, machine learning algorithms are considered to be the weakest link [10] due to a) their nature of constant evolution; and b) sophisticated adversaries manipulating their behaviours to exploit the models. These models can be exploited using various methods, namely evasion attacks and poisoning attacks. ML models applied to real-world systems must deal with dynamically changing data patterns and a wide variety of inputs, to the extent that it is highly improbable that every situation is captured in the training data [11]. Areas where an ML model is vulnerable, are known as âadversarial regionsâ [12], which are near the decision boundary. These areas indicate that training data has not covered the entire feature space, which allows adversaries to apply trial and error or reverse engineering to find âadversarial samplesâ. These samples can be used to fool the model and enable the data inputs to be misclassified, thus compromising a classifier and the integrity of its system. These attacks are exploratory and known as evasion attacks [13].

Zamani et al. [14] defined intrusion Detection as *âThe process of dynamically monitoring events occurring in a computer system or network, analysing them for signs of possible incidents and often interdicting the unauthorised*

DOI: 10.5281/zenodo.5226915
Received: January 22, 2021 Accepted: June 19, 2021

access.â This process of monitoring events is achieved using two detection methods: anomaly detection and signature-based detection [15]. Signature-based detection, also known as misuse detection, identifies attacks through their patterns and/or signatures [16]. When the system classifies a known malicious signature, it generates an alert for the network administrator, or the threat is automatically blocked. Anomaly detection systems utilise machine learning classifiers to analyse system events and network traffic to build ânormalâ network profiles. When new incoming traffic reaches the IDS, it is compared to the previously learned profile, identifying events or traffic that are anomalous [16].

II. DATA POISONING ATTACKS

Data poisoning attacks are known as a training-time attacks, where an adversary targets the ML model during the training phase, limiting the modelâs effectiveness through manipulation of the training data [5]. Non-stationary machine learning models that are trained on user-provided data are vulnerable to data poisoning attacks, whereby malicious parties inject false training data to corrupt the learning process [17], subsequently degrading the accuracy of the classifier and the integrity of the overall system.

A. Data Poisoning Techniques

When attacking a model using data poisoning, adversaries have three broad strategies they can use, dependant on their level of access to the data and learning process. First, label modification, also known as label flipping [5], is a strategy where an adversary is constricted and cannot inject their own inputs but can modify the labels of the existing training datasets [18]. Finally, data modification attacks can occur when the adversary has no access to the learning algorithm but has full access to training data [19]; the data can be poisoned by modifying the features before the dataset is used to train the model.

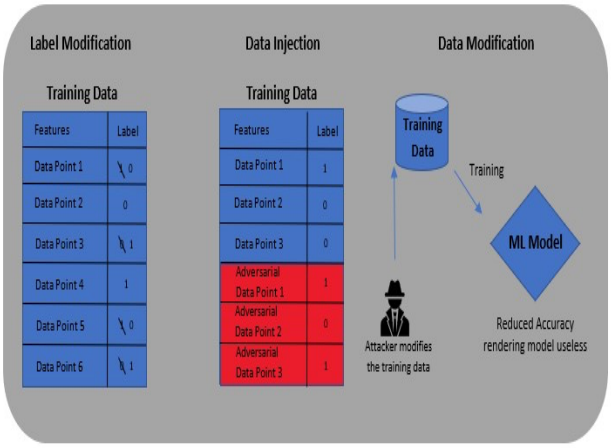

Fig. 1. Data Poisoning Methods

Alternatively, if it is known that the defender is training their model iteratively, over an extended period, a boiling frog attack may be implemented [5]. This iterative retraining of the model provides the attacker with the opportunity to poison new training datasets repetitively. By poisoning a small amount of data each time, the impact of any single data injection is minimal; however, over multiple training periods, the impact becomes significant. They proposed a network-wide anomaly detection method, known as *'Antidote'*, that is less sensitive to outliers and can reject poisonous training data.

B. Data Poisoning Countermeasures

There are two defences against data poisoning attacks outlined in the literature [20]: *Outlier Sanitisation* and *Reject on Negative Impact (RONI)*. *Outlier Sanitisation* defences [17] identify data points that are significantly different from historical training data and reject them. The challenge with Outlier Sanitisation is to optimise the threshold, considering how far outside the âlocal groupâ a data point must be, in order to be considered an outlier. If the threshold is too far, the model will be susceptible to poisoning; whereas if the threshold is too close, the model may be under-fitted and

DOI: 10.5281/zenodo.5226915
Received: January 22, 2021 Accepted: June 19, 2021

ineffective. Determination of which training points are malicious begins with a control classifier trained on a base set of inputs; then new training data are added to train a second classifier. The two classifiers are tested, and the accuracy of the latter is compared against the former. If the new data significantly negatively impact the accuracy of the second classifier, the data are rejected from the training set. *RONI* is far less susceptible to the boiling frog attack but is more complex to configure and operate.

III. PROPOSED METHODOLOGY

This section discusses the methodology used to develop a trustworthy Deep Learning model that is inherently more robust against data poisoning attacks. The proposed method is designed to improve Deep Learning model robustness by optimising training datasets and iteratively improving the model parameters. This methodology includes a description of the feature selection and models design processes used to determine the most effective architecture for the Feed Forward Deep Neural Network (FFDNN). As shown in Figure 2, the Deep Learning models are trained on the various poisoned datasets using four different correlation analysis rates.

Once trained, the evaluation metrics are compared and analysed to determine the maximum correlation rate for the next phase of testing. The FFDNN will be trained using this feature selection process and a variety of model parameter permutations (Cost Function, Hidden Layer Size, Epochs) to evaluate the best Feed Forward Deep Learning architecture in this simulated adversarial IoT environment. Once the optimal architecture is determined for the datasets of the ToN_IoT, they will be compared to their respective base models, to ensure the new models are more robust to the poisoned training data.

A. Feature Selection

Feature Selection is a data pre-processing strategy designed to reduce high-dimensional datasets with the goal of reducing computational requirements and increasing the performance of the output model [21], [22]. Additionally, by reducing the dimensionality of the dataset, superfluous features that do not improve the model performance are removed.

1) Correlation Analysis

Correlation refers to the association or relationship between two quantitative variables in a given dataset [23]. Through correlation analysis, highly correlated features can be determined, and removed from the dataset prior to training. The output of correlation analysis is the Correlation Coefficient *r* which ranges from -1 to +1.

There are three possible types of correlation within any given dataset [24], as shown in Figure 3:

- Positive Correlation: an *r* value between 0 and +1 indicates a positive correlation between the two variables. This means that an increase in the first variable A will lead to an increase in the correlated variable B.
- Negative Correlation: an *r* value between -1 and 0 indicates a negative correlation between the two variables. This means that an increase in the first variable A will lead to a decrease in the correlated variable B.
- No Correlation: an *r* value of 0 indicates no relationship between the two variables A and B.

Each correlation coefficient exists on a spectrum, such that an *r* value closer to 1 or -1 represents a strong positive or negative correlation respectively between the two variables. A dataset with two or more strongly correlated features may be affected by a problem called *Multicollinearity* [25].

Correlation Reduction Inputs: Training Dataset (*D*), Maximum correlation rate (*max*) Output: Filtered Training Dataset Steps: for each feature pair in *D* Calculate the correlation coefficient *r*

if *r* > *max*: Remove feature with the greater mean absolute correlation end if end for return Filtered Training Dataset

B. Feature Relevance

Following the feature selection process, the reduced datasets allowed for further investigation into the importance of each remaining feature to class predictions. The research field of explainable AI seeks to increase the transparency often lacked by Machine, and particularly Deep Learning models [26]. Through the reduction of training datasets, the relevance of each individual feature to the Feed Forward Deep Neural Networks is easier to observe, further increasing the transparency of the model [27]. Once the optimal correlation rate is determined, the ten most relevant features from

DOI: 10.5281/zenodo.5226915
Received: January 22, 2021 Accepted: June 19, 2021

each benchmark dataset were extracted and examined, to determine the impact of each feature to model performance and prediction accuracy.

C. Feed Forward Deep Neural Network Design

This section details the architectural design aspects tested during the experiments for this phase of research. The parameters of the Feed Forward Deep Learning models that were tested and evaluated are the cost function, hidden layer size and training epochs.Cost functions evaluate the difference between the outputs of the model and the correct labels of the input data [28]. The objective of this function is to minimise the cost such that the optimal weights and biases are determined for the model, leading to increased accuracy. Perfect models will have a cost of 0 [28]. To determine the optimal cost function to be used in the Deep Learning architecture, the Quadratic and Cross Entropy cost functions will be tested and evaluated:

- Quadratic Cost: Also known as Mean Squared Error, the quadratic cost function measures the average of the squared difference between the predicted value p and the actual label y [28]. Due to this squaring, the quadratic cost function heavily penalises wildly incorrect predictions regardless of direction. The quadratic cost C is calculated as:

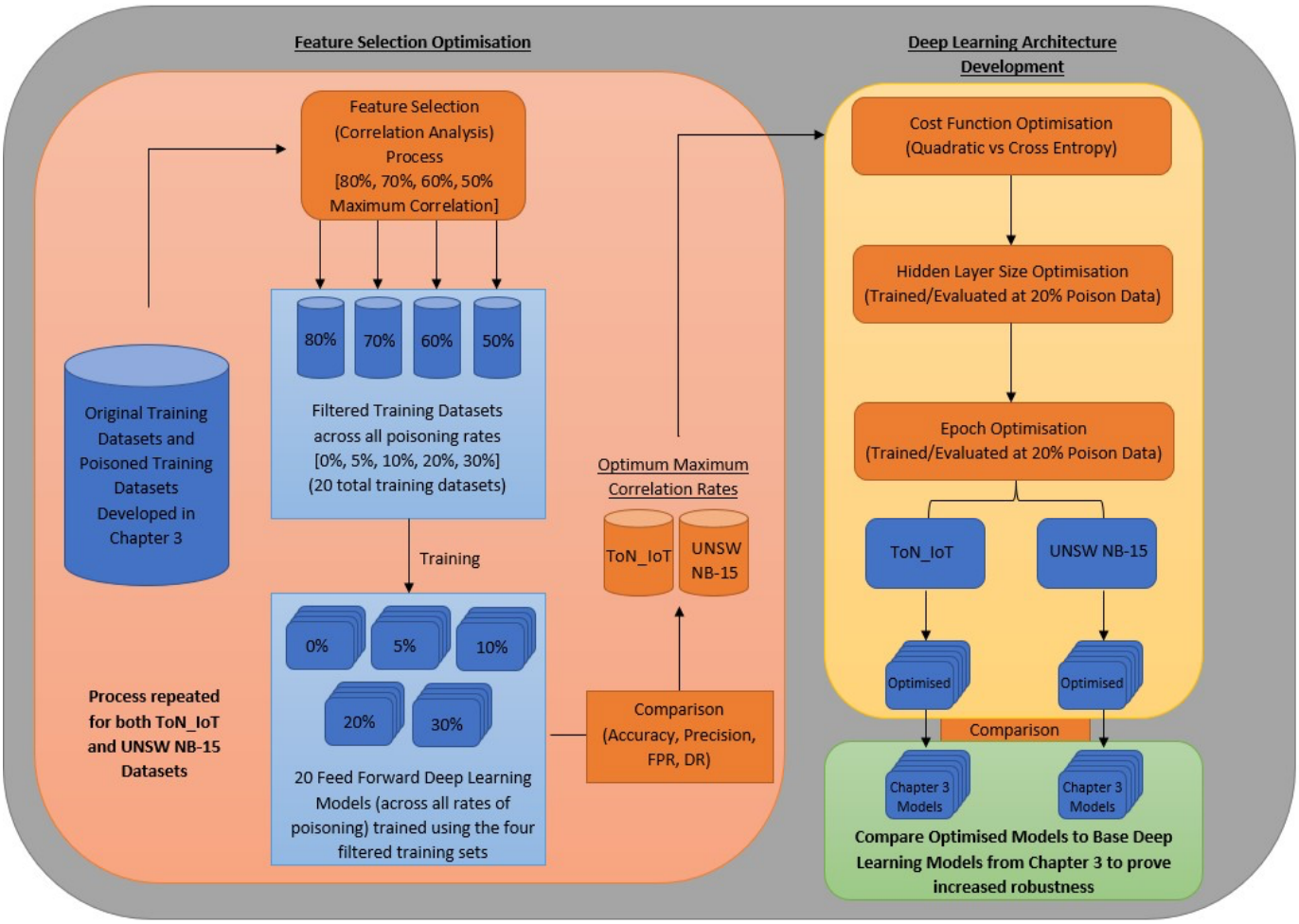

Fig. 2. Proposed robust Deep Learning architecture development methodology against data poisoning attacks in IoT networks

DOI: 10.5281/zenodo.5226915
Received: January 22, 2021 Accepted: June 19, 2021

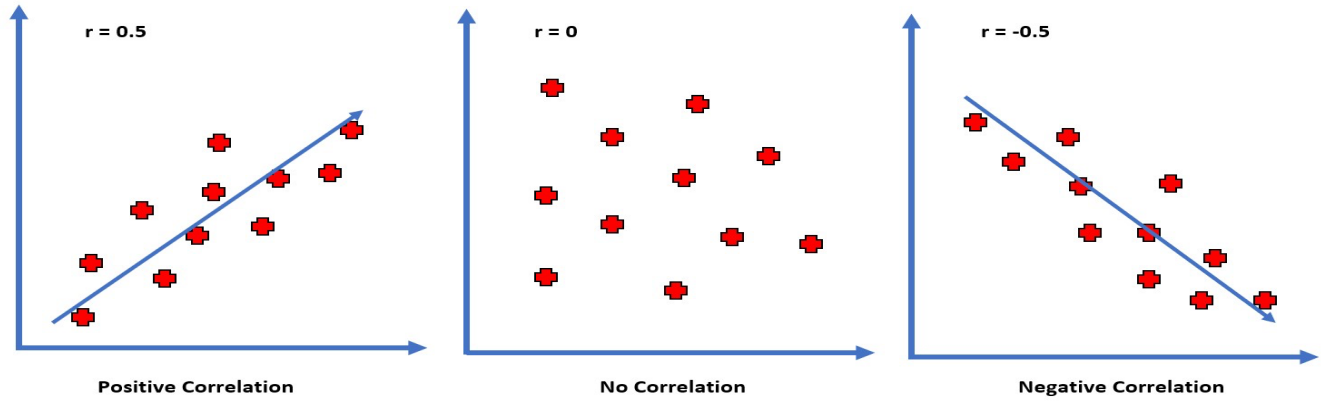

Fig. 3. Correlation Analysis

$$C = \frac{\sum_{j=1}^{n}(p_j - y_j)^2}{n} \quad (1)$$

- Cross Entropy: Also known as 'log loss', the cross entropy cost function measures the model's performance when the classification is between 0 and 1. As the output of the log function increases when the predicted value p diverges from the actual label y, the further the predicted value is from its actual label, the higher the cost [28]. Due to this, the cross entropy cost function heavily punishes predictions that are confident and wrong. For binary classification where the number of output classes is two, cross entropy C is calculated as:

$$C = -\sum_{j=1}^{n}(y_j log\,(p_j) + (1 - y_j)\,log\,(1 - p_j)) \quad (2)$$

1) Hidden Layers

The second parameter to be tested is the size of the hidden layers within the model. A hidden layer sits in between the input and output layers in the network and is challenging to observe, hence the name 'hidden layer' [28]. For further explanation of the intricacies of hidden layers, *Feed Forward Deep Neural Network*. The original models had hidden layers of size (15,15); whereas the new models will be trained with hidden layers of size (5,5), (8,8), (15,15), (25,25), (50,50), (75,75) and (100,100) with their accuracy, precision, FPR and DR evaluated after each training session to determine which hidden layer size, or range of sizes best suits an optimal FFDNN model. The variation of hidden layer sizes allows for the observation of performance changes at sizes both above and below those of the base models, to determine the size that optimises model performance across the ToN_IoT datasets.

2) Epochs

The final parameter to be tested is that of the training epochs. An epoch is the period taken to pass the entire dataset through the network *once* [28], [29]. A model requires the dataset to be passed through multiple times in order to adjust the weights and biases to best fit the data [29]. As the number of epochs increases, the weights and biases are slowly adjusted until an optimised set is found. When increasing the number of training epochs, the issues of over and under fitting must be addressed. A graphical representation of these phenomena can be seen in Figure 4.

- Overfitting: Overfitting occurs when the model learns the training data so well, and in such fine detail that it negatively affects the model's ability to predict new data [28]. The weights and biases within the FFDNN are not applicable to new, unseen testing sets, and as such, the model consequently performs poorly. Overfitting is characterised by an increase in training accuracy, with a concurrent decrease in validation accuracy, as the model performs well on existing data but cannot accurately categorise new data [30]. In order to avoid overfitting during training, k-fold cross-validation was used to evaluate the model's performance over the entire dataset.
- Underfitting: Underfitting occurs when the model is not adequately trained, such that it cannot accurately predict new data, and commonly performs poorly on training data [31]. An underfitted model requires more training epochs, and potentially a more extensive training dataset to ensure the FFDNN's weights and biases are optimised during training.

DOI: 10.5281/zenodo.5226915
Received: January 22, 2021 Accepted: June 19, 2021

The new models were trained using 10, 25, 50, 100, 200 and 250 epochs in order to analyse the impact increasing epoch ranges has on a poisoned FFDNN model, and determine the optimal number of epochs or range of epochs for the Deep Learning model within this adversarial environment.

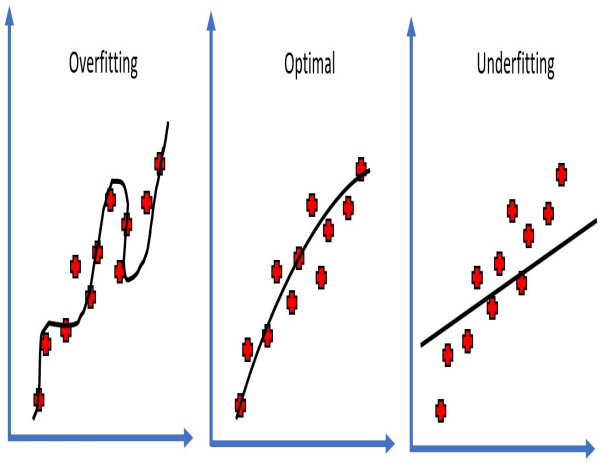

Fig. 4. Overfitting vs Underfitting

IV. EXPERIMENTAL DESIGN

A. Experimental Setting B. Training Datasets

The training datasets used throughout this research is the ToN_IoT Network Data [32], both of which contain a wide variety of contemporary normal and abnormal network observations, to allow for practical training of all models in order to mimic a deep learning model deployed as part of a more extensive network security application. Both these datasets contain a known ground truth.

C. Evaluation Metrics

The models that are explored in this paper were evaluated based on their performance in terms of accuracy, precision, false positive rate and detection rate. These metrics are calculated from the four classifications of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN).

V. FEATURE SELECTION RESULTS

A. 0% Data Poisoning 1) ToN_IoT Dataset

Using the ToN_IoT training datasets, greater than 99% accuracy was observed for each maximum correlation rate except 50%. As expected, with no poisoning in the dataset, all models (except the model trained using the 50% maximum correlation dataset) displayed accuracy of 99%+, FPR of <1% and DR of 98%+. As shown in Figure 5, the model trained using the 50% correlation rate dataset underperformed the three other models in both accuracy and detection rate without poisoned data present. This indicates that this training dataset has had too many features removed, leading to an ineffective, underfitted deep learning model. Overall, the highest performing training dataset, with 0% poisoning present, was the 70% maximum correlation, closely followed by 80% correlation.

B. 10% Data Poisoning

When trained using 10% poisoned data, the Feed Forward Deep Learning models displayed a relatively linear change in the evaluation metrics when compared to the 5% poisoned models. Once again, the models trained on the dataset with

DOI: 10.5281/zenodo.5226915
Received: January 22, 2021 Accepted: June 19, 2021

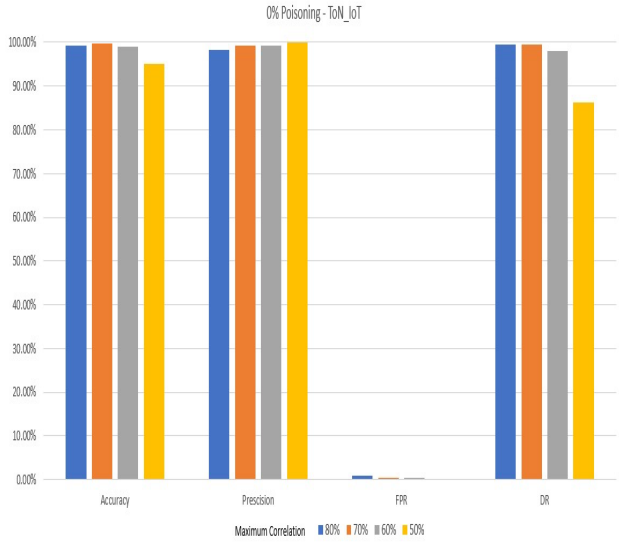

Fig. 5. ToN_IoT Deep Learning Model Comparison
(0% Poison Data Present)

50% maximum correlation showed increased deterioration, however, this trend continued across all poisoning rates. Furthermore, the model trained using the 70% maximum correlation rate underperformed slightly, specifically in precision and FPR, which is surprising due to the relatively consistent results across the four other poisoning rates. This model was re-trained, yielding the same result, and indicating that at 10% poisoning, 70% maximum correlation offers an ineffective dataset for deep learning using the base architecture.

As seen in Figure 6, the 60% and 80% maximum correlation rate datasets both produced higher performing models, with the 60% maximum correlation rate model outperforming with an accuracy of 89.29%, precision of 88.69%, FPR of 6.39% and DR of 82.22%. At only 10% poisoning, the models exhibited significant losses in classifier integrity, with 17.78% of attack vectors remaining undetected for the highest performing model. This indicated that the feature selection process was reducing dataset dimensionality and superfluous features, but was not leading to greater model integrity.

C. 20% Data Poisoning

When trained using 20% poisoned data, the Feed Forward Deep Learning models continued to display a linear change in the evaluation metrics, with accuracy between 78.27% to 79.07%, precision of 75.82% to 78.10%, FPR of 15.10% to 13.09% and DR of 67.73% to 68.67% for the 60%, 70% and 80% maximum correlation rate datasets. The 50% maximum correlation rate dataset continued to display reduced performance across all metrics, with an accuracy of 75.31%, precision of 69.29%, FPR of 21.71% and DR of 71%.

As shown in Figure 7, the 50% maximum correlation model, whilst underperforming throughout previous tests, produced the highest detection rate at this level of data poisoning. This increase, combined with raised FPR and reduced accuracy was not an indication of higher performance, but rather, of the rapid deterioration of the model's ability to classify new data correctly. In order to compensate for this, the model classified

DOI: 10.5281/zenodo.5226915
Received: January 22, 2021 Accepted: June 19, 2021

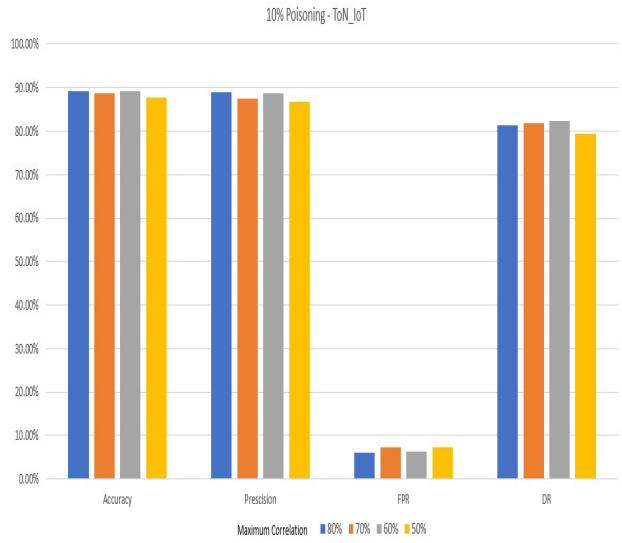

Fig. 6. ToN_IoT Deep Learning Model Comparison
(10% Poison Data Present)

more test data than expected as attack vectors, resulting in the increased FPR and DR.

Overall, at a poisoning rate of 20%, the highest performing model was trained using a maximum correlation rate of 80%. This was despite not having the highest DR at 67.73%, compared to 68.67%, shown by the 70% correlation rate model. The model's other three metrics were significantly higher, indicating increased robustness at this high rate of poisoning.

D. 30% Data Poisoning

When trained using the 30% poisoned data, each of the Feed Forward Deep Learning models showed significant deterioration across all evaluation metrics, as borne out by the robustness evaluations. The highest performing model, trained using the 60% maximum correlation rate dataset, significantly outperformed the other three models with an accuracy of

64.82%, precision of 59.18%, FPR of 34.49% and DR of 63.95%. This model displayed a lower DR than the other three, which all showed apparent signs of deterioration, with drastically increased FPRs and DRs coupled with significantly lower accuracy.

Both models trained using 50% and 80% maximum correlation rate datasets showed signs of complete model corruption, with FPRs and DRs of more than 99%. This indicates that the models are entirely unable to make predictions with any accuracy, and as such, have classified all validation test data as attack vectors, rendering the model worthless. Overall, the 60% and 70% maximum correlation datasets produced models that showed signs of robustness at this increased rate of data poisoning. The models trained using the other two datasets were unusable at this rate. The model trained using 60% correlated data is clearly the highest performing model at this increased rate of data poisoning, as seen in Figure 8.

E. Feature Relevance Results

Processing both datasets using the optimal correlation rates
(70% for both the ToN_IoT & UNSW NB-15 datasets) allows

DOI: 10.5281/zenodo.5226915
Received: January 22, 2021 Accepted: June 19, 2021

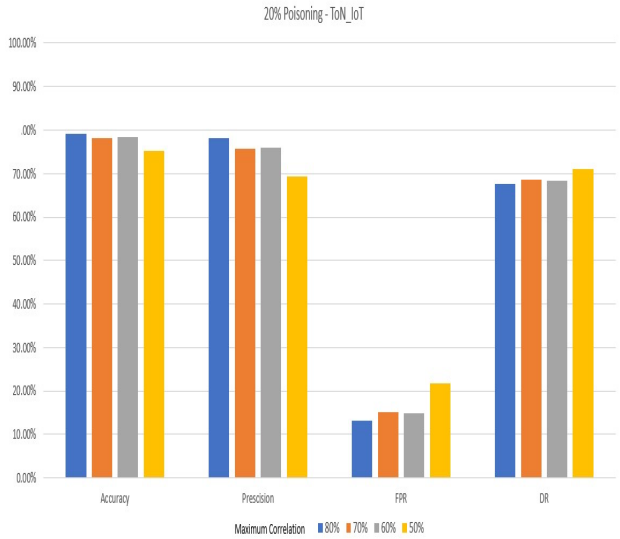

Fig. 7. ToN_IoT Deep Learning Model Comparison(20% Poison Data Present)

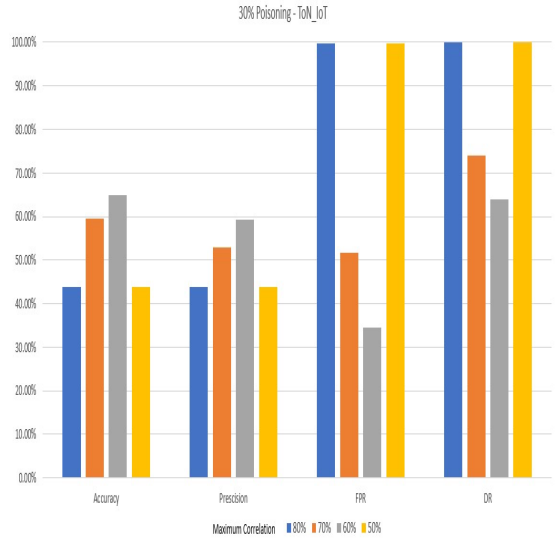

Fig. 8. ToN_IoT Deep Learning Model Comparison
(30% Poison Data Present)

for the derivation of the most important features for model development. After the correlation reduction process has removed superfluous features, the ten most important features during training for Feed Forward Deep Learning models were determined, and are shown below; Table I for the ToN_IoT dataset and Table II for the UNSW NB-15 Dataset. The descriptions for each of the features are derived from the base dataset documentation [33], [34].

The features detailed above were selected as having the highest contribution to the identification of correct class labels. As expected, for the ToN_IoT dataset, the most relevant features pertained to information contained in the packet header, such as the source and destination IP addresses and destination port. Throughout the learning process, the models have associated attack traffic with attempts to connect from malicious sources and to vulnerable or rarely used ports. The most relevant features from the UNSW NB-15 dataset pertained to *statistic-based traffic features* [35] such as *ct_dst_src_ltm* and *ct_srv_src*; and *basic features* such as the mean packet size and connection duration. Each of these features correlates to inconnectivity between host systems. Increased interconnectivity is characteristic of cyber attacks, such as DDoS and botnet attacks.

DOI: 10.5281/zenodo.5226915
Received: January 22, 2021 Accepted: June 19, 2021

VI. ARCHITECTURAL DEVELOPMENT RESULTS

A. Cost Functions

For 5% and 10% Data Poisoning, both cost functions produced models with relatively equal metrics, with the Cross Entropy cost function outperforming the Quadratic cost function marginally. When trained using 20% poisoned data, the model produced using the Quadratic cost function begins to significantly outperform the Cross Entropy model, displaying an FPR of 12.20% in comparison with that of the Cross Entropy model, at 16.12%, whilst also having a 1.8% higher accuracy and 3.5% higher precision, illustrated in Figure 9. The Cross Entropy model outperforms the Quadratic model in Detection Rate, by 1.3%; However, when coupled with the significantly increased FPR, shown in Figure 10, this model was showing early signs of deterioration, in Section V.

When trained using 30% poisoned data, differences between the cost functions are heightened, with the FPR of the Cross Entropy model increasing to 34.29%, 8.7% higher than that of the Quadratic model, which sits at 25.56%. At this rate of poisoning, the Quadratic model is superior, with 2.7%

TABLE I FEATURE IMPORTANCE - TON_IOT

Feature Importance - ToN_IoT	
Feature	Description
src_ip	Source IP address
dst_port	Destination port
conn_state	Connection state (i.e. S0, S1, REJ) sourced from the *Bro* logs
dns_qtype	Value which specifies the DNS query types
dst_ip	Destination IP addresses
http_method	HTTP request methods (i.e. GET, POST or HEAD)
weird_notice	Indicates if the violation/anomaly was turned into a notice
http_trans_depth	Pipelined depth into the HTTP connection
http_status_code	Status code returned by the HTTP server
dns_rcode	Response code values in the DNS response

TABLE II

FEATURE IMPORTANCE - UNSW NB-15

Feature Importance - UNSW NB-15	
Feature	Description
dmean	Mean of the row packet size transmitted by the destination IP address
ct_dst_src_ltm	No. of connections of the same source and destination address in the last 100 connections
dur	Total record duration
is_sm_ips_ports	Binary variable, equals (1) if the source and destination IP addresses equal and port numbers equal
ct_srv_src	No. of connections that contain the same service and source address in 100 connections
ct_state_ttl	No. for each state according to specific range of values for source/destination time to live.
sttl	Source to destination time to live value
sloss	Source packets re-transmitted or dropped

DOI: 10.5281/zenodo.5226915
Received: January 22, 2021 Accepted: June 19, 2021

dload	Destination bits per second
dpkts	Destination to source packet count

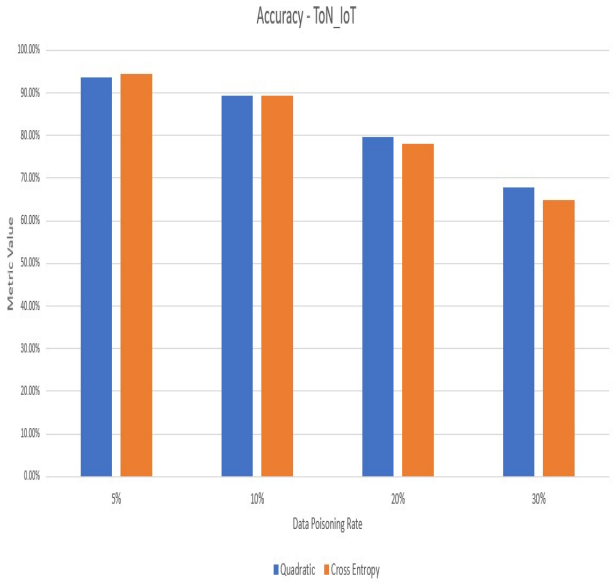

Fig. 9. Cost Function Accuracy - ToN_IoT higher accuracy and 5% higher precision rates. As illustrated in Figure 11, the model shows a 4.92% lower Detection Rate, whilst this is, however, due to the Cross Entropy model deteriorating, as indicated when evaluated along with its increased FPR. For lower rates of poisoning, between 5% and 10%, the

Cross Entropy cost function produced a slightly more robust

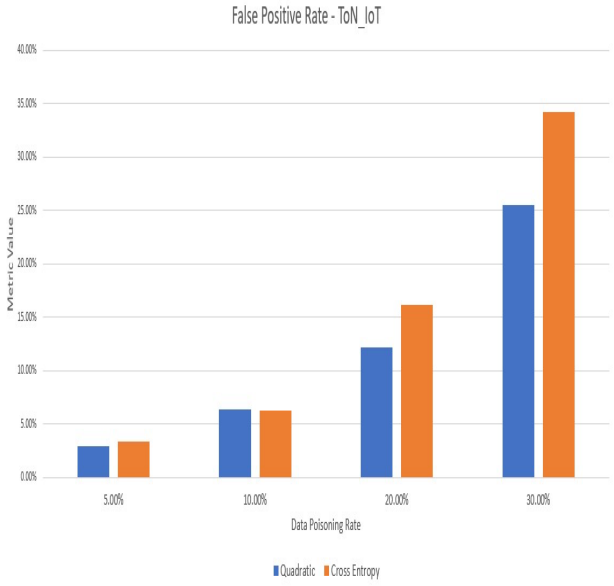

Fig. 10. Cost Function False Positive Rate - ToN_IoT

DOI: 10.5281/zenodo.5226915
Received: January 22, 2021 Accepted: June 19, 2021

FFDNN model. However, at higher rates, as observed in the cases of more than 20% poisoning, the Quadratic cost function produced a significantly more robust model, indicating that it is the optimal cost function in this simulated adversarial environment.

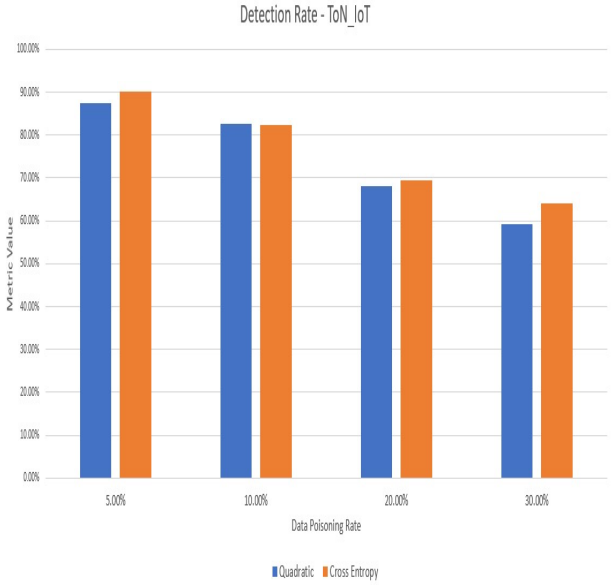

Fig. 11. Cost Function Detection Rate - ToN_IoT

1) ToN_IoT Dataset

As shown in Table III, the hidden layer size variation yielded a slight increase to the performance of the models trained using ToN_IoT data. This was maximised at a hidden layer size of (75,75), which is evidently the highest performing parameter value. This model displayed an accuracy of 79.00%, precision of 77.70%, FPR of 13.48% and DR of 68.11%. This model exhibited one of the lowest DRs out of the seven models trained. However, the reduced FPR indicates robustness in comparison with other models. This multi-variable analysis produces a better model overall, instead of solely using the model with the highest DR.

TABLE III

HIDDEN LAYER SIZE COMPARISON - TON_IOT (TRAINED AT 20% POISONED DATA)

Hidden Layer Size - ToN_IoT				
	Accuracy	Precision	FPR	DR
(5,5)	78.35%	75.95%	15.01%	68.73%
(8,8)	78.54%	76.41%	14.62%	68.63%
(15,15)	78.70%	76.81%	14.27%	68.52%
(25,25)	78.62%	76.91%	14.10%	68.08%
(50,50)	78.42%	76.20%	14.76%	68.53%
(75,75)	79.00%	77.70%	13.48%	68.11%
(100,100)	78.82%	77.42%	13.67%	67.94%

2) ToN_IoT Dataset

As shown in Table IV, the epoch variation led to significant increases in the performance of the models trained using ToN_IoT data. This was maximised at 100 epochs, which evidently produced the highest performing model. At 100

DOI: 10.5281/zenodo.5226915
Received: January 22, 2021 Accepted: June 19, 2021

epochs, the model's accuracy and DR increased substantially, to 80.51% and 69.54% respectively, slightly higher than the observations from the base Deep Learning model. Additionally, the FPR for the 100 epoch model was reduced to 11.92%, 0.34% lower than the base model. Whilst the difference between this and the base model is small, the models trained using the ToN_IoT data were inherently more robust to the data poisoning.

TABLE IV EPOCH COMPARISON - TON_IOT (TRAINED AT 20% POISONED DATA)

Epoch Comparison - ToN_IoT				
	Accuracy	Precision	FPR	DR
10	78.89%	77.39%	13.75%	68.23%
25	79.34%	78.70%	12.65%	67.73%
50	79.34%	78.81%	12.53%	67.54%
100	80.51%	80.10%	11.92%	69.54%
200	79.07%	77.51%	13.75%	68.66%
250	79.00%	76.81%	14.49%	69.57%

B. Feature Selection Optimisation

Models trained using the ToN_IoT data showed varying results, with the optimal correlation rate differing at each rate of data poisoning. A maximum correlation of 80% produced the best models at 5% and 20% poisoning; whilst 60% maximum correlation produced the best models at 10% and 30% poisoning. The 70% maximum correlation dataset produced the best model when no poisoning was present. As a result of this variation, analysis of correlation rate performance across all poisoning rates was explored; and 70% maximum correlation proved to provide a consistent training dataset; and as such, was implemented throughout the architecture development process. As seen throughout the Feature Selection results, the models trained using the UNSW NB-15 data performed more consistently than the ToN_IoT models, with four of five poisoning rates producing optimal models when trained using a 70% maximum correlation rate during the feature selection process. For the UNSW NB-15 models, 70% is the optimum correlation rate, and these pre-processed datasets were therefore implemented in the Architecture Development process.

Across both datasets, 50% maximum correlation proved to be too low to produce an adequate dataset for Deep Learning. Models trained using these datasets showed deterioration before any poisoning was present. This decline was most evident in the ToN_IoT models at 0% poisoning, with the 50% maximum correlation model displaying a 4% lower accuracy and 13% lower DR than the next highest performing model. This deterioration was due to the feature selection process removing too many relevant features, leading to a dataset that was ineffective for training. At all rates of data poisoning, models trained using the 50% maximum correlation rate data yielded drastically lower accuracy and higher FPRs, indicating that these models were significantly underfitted in comparison to alternative models at all rates of poisoning.

The advantages of this feature selection process are the reduced dimensionality of the datasets, leading to reduced multicollinearity and overall processing time, in comparison with the base models. This reduction in training time allowed for faster model development using CPU resources, producing models of similar performance to the base models, despite removing up to twenty-two features from the training dataset (ToN_IoT at 60% maximum correlation). The dimensionality reduction allowed for a greater understanding of the features that are most important to development of these deep learning models. The apparent disadvantage of this feature selection process is the lack of any notable performance increase to the deep learning models. The models produced from this research, detailed throughout Section V performed similarly to the deep learning models. This indicates that this feature selection process can be further optimised for Feed Forward Deep Neural Network development.

DOI: 10.5281/zenodo.5226915
Received: January 22, 2021 Accepted: June 19, 2021

VII. CONCLUSION

This paper analysed and discussed the impacts of data poisoning attacks on Feed-Forward Deep Neural Network models and proposed an architecture that has shown to improve model performance in the presence of data poisoning attacks. The feature selection process facilitated the removal of superfluous features, decreasing data dimensionality and impact of multicollinearity. The two resultant architectures produced models that displayed increased performance when compared to the base models. The optimised ToN_IoT model displayed slightly increased metrics up to 20% data poisoning, but performed worse when trained using sanitised data. In future, it is recommended that DL models be trained using these architectures with data poisoning defences implemented to examine their combined effectiveness. These defences will allow for additional increases to model robustness, further improving their suitability for use in online cyber defence systems.

REFERENCES

[1] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, 2018.

[2] Y. Zhao, J. Chen, J. Zhang, D. Wu, M. Blumenstein, and S. Yu, "Detecting and mitigating poisoning attacks in federated learning using generative adversarial networks," *Concurrency and Computation: Practice and Experience*, p. e5906, 2020.

[3] E. Alhajjar, P. Maxwell, and N. D. Bastian, "Adversarial machine learning in network intrusion detection systems," *arXiv preprint arXiv:2004.11898*, 2020.

[4] A. Agarwal and A. Agarwal, "The security risks associated with cloud computing," *International Journal of Computer Applications in Engineering Sciences*, vol. 1, pp. 257–259, 2011.

[5] Y. Vorobeychik, *Adversarial machine learning*. [San Rafael, California]: San Rafael, California : Morgan & Claypool, 2018.

[6] S. Leminen, M. Westerlund, M. Rajahonka, and R. Siuruainen, "Towards iot ecosystems and business models," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7469, 2012, pp. 15–26.

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

[8] B. S. Vivek, K. R. Mopuri, and R. V. Babu, "Gray-box adversarial training," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[9] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE symposium on security and privacy (sp)*. IEEE, 2017, Conference Proceedings, pp. 39–57.

[10] V. Duddu, "A survey of adversarial machine learning in cyber warfare," *Defence Science Journal*, vol. 68, no. 4, 2018.

[11] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, 2006, Conference Proceedings, pp. 16–25.

[12] P. McDaniel, N. Papernot, and Z. B. Celik, "Machine learning in adversarial settings," *IEEE Security & Privacy*, vol. 14, no. 3, pp. 68–72, 2016.

[13] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, 2011, Conference Proceedings, pp. 43–58.

[14] M. Zamani and M. Movahedi, "Machine learning techniques for intrusion detection," *arXiv preprint arXiv:1312.2177*, 2013.

[15] M. F. Elrawy, A. I. Awad, and H. F. Hamed, "Intrusion detection systems for iot-based smart environments: a survey," *Journal of Cloud Computing*, vol. 7, no. 1, p. 21, 2018.

[16] B. Zarpelao, R. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in internet of things," *J. Netw. Comput. Appl.*, vol. 84, pp. 25–37, 2017.

[17] J. Steinhardt, P. W. W. Koh, and P. S. Liang, "Certified defenses for data poisoning attacks," in *Advances in neural information processing systems*, 2017, Conference Proceedings, pp. 3517–3529.

DOI: 10.5281/zenodo.5226915
Received: January 22, 2021 Accepted: June 19, 2021

[18] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *arXiv preprint arXiv:1206.6389*, 2012.

[19] B. Biggio, K. Rieck, D. Ariu, C. Wressnegger, I. Corona, G. Giacinto, and F. Roli, "Poisoning behavioral malware clustering," in *Proceedings of the 2014 workshop on artificial intelligent and security workshop*, 2014, Conference Proceedings, pp. 27–36.

[20] B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. Rubinstein, U. Saini, C. Sutton, J. Tygar, and K. Xia, *Misleading learners: Co-opting your spam filter*. Springer, 2009, pp. 17–51.

[21] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 1–45, 2017.

[22] M. A. Al-Garadi, A. Mohamed, A. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for internet of things (iot) security," *IEEE Communications Surveys & Tutorials*, 2020.

[23] N. Gogtay and U. Thatte, "Principles of correlation analysis," *Journal of the Association of Physicians of India*, vol. 65, no. 3, pp. 78–81, 2017. [24] R. Rivera, C. M. PÃ©rez, M. N. MartÃnez, and E. H. SuÃ¡rez, "Correlation analysis," in *Applications of Regression Models in Epidemiology*. Hoboken, NJ, USA: John Wiley & Sons, Inc, 2017, pp. 87–96.

[25] C. Obite, N. Olewuezi, G. Ugwuanyim, and D. Bartholomew, "Multicollinearity effect in regression analysis: A feed forward artificial neural network approach," *Asian Journal of Probability and Statistics*, pp. 22– 33, 2020.

[26] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52138– 52160, 2018.

[27] D. Janzing, L. Minorics, and P. Blöbaum, "Feature relevance quantification in explainable ai: A causal problem," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2907–2916.

[28] A. C. Ian Goodfellow, Yoshua Bengio, *Deep Learning*. MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[29] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[31] W. M. Van der Aalst, V. Rubin, H. Verbeek, B. F. van Dongen, E. Kindler, and C. W. Günther, "Process mining: a two-step approach to balance between underfitting and overfitting," *Software & Systems Modeling*, vol. 9, no. 1, p. 87, 2010.

[32] N. Moustafa, "Ton_iot datasets," https://cloudstor.aarnet.edu.au/plus/s/ ds5zW91vdgjEj9i, 2019.

[33] N. Moustafa, G. Creech, and J. Slay, *Anomaly detection system using beta mixture models and outlier detection*. Springer, 2018, pp. 125–135. [34] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "Ton_iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020.

[35] H. G. Kayacik, A. N. Zincir-Heywood, and M. I. Heywood, "Selecting features for intrusion detection: A feature relevance analysis on kdd 99 intrusion detection datasets," in *Proceedings of the third annual conference on privacy, security and trust*, vol. 94. Citeseer, 2005, pp. 1723–1722.

DOI: 10.5281/zenodo.5226915
Received: January 22, 2021 Accepted: June 19, 2021