Mitigating Label Flipping Attacks in Malicious URL Detectors Using Ensemble Trees

Ehsan Nowroozi[1], Nada Jadalla[1], Samaneh Ghelichkhani[1], and Alireza Jolfaei[1]

[1]Affiliation not available

December 22, 2023

Mitigating Label Flipping Attacks in Malicious URL Detectors Using Ensemble Trees

Ehsan Nowroozi, *Senior Member, IEEE*, Nada Jadalla, *Member, IEEE*, Samaneh Ghelichkhani, *Member, IEEE*, Alireza Jolfaei, *Senior Member, IEEE*

***Abstract*—Malicious URLs provide adversarial opportunities across various industries, including transportation, healthcare, energy, and banking which could be detrimental to business operations. Consequently, the detection of these URLs is of crucial importance however, current Machine Learning (ML) models are susceptible to backdoor attacks. These attacks involve manipulating a small percentage of training data labels, such as Label Flipping (LF), which changes benign labels to malicious ones and vice versa. This manipulation results in misclassification and leads to incorrect model behavior. Therefore, integrating defense mechanisms into the architecture of ML models becomes an imperative consideration to fortify against potential attacks.**

The focus of this study is on backdoor attacks in the context of URL detection using ensemble trees. By illuminating the motivations behind such attacks, highlighting the roles of attackers, and emphasizing the critical importance of effective defense strategies, this paper contributes to the ongoing efforts to fortify ML models against adversarial threats within the ML domain in network security. We propose an innovative alarm system that detects the presence of poisoned labels and a defense mechanism designed to uncover the original class labels with the aim of mitigating backdoor attacks on ensemble tree classifiers. We conducted a case study using the Alexa and Phishing Site URL datasets and showed that LF attacks can be addressed using our proposed defense mechanism. Our experimental results prove that the LF attack achieved an Attack Success Rate (ASR) between 50-65% within 2-5%, and the innovative defense method successfully detected poisoned labels with an accuracy of up to 100%.

***Index Terms*—Adversarial machine learning, backdoor attack, corrupted training sets, cybersecurity, poisoning attack.**

I. Introduction

WHILE URLs (Uniform Resource Locators) play a crucial role in web browsing and the overall functioning of the World Wide Web, they have also been used as gateways to adversely impact and detrimentally affect users and businesses. To address this issue, extensive research has been conducted using ML systems, such as the Random Forest (RF) models [1]. However, these models are susceptible to backdoor attacks due to their vulnerability to biased or manipulated training data. RF is a widely adopted ensemble learning technique used across various domains throughout recent years due to its ability to handle complex datasets, feature interactions, and noisy data however, it can be vulnerable to adversarial attacks e.g. poisoning attacks. Malicious URLs can be poisoned in different ways to fool detection systems such as RF to evade detection and appear legitimate. One approach is through flipping the labels of the URLs from malicious to benign and benign to malicious with the aim to fool the system leading to malicious content for gaining unauthorized access or disrupting the systems. You can find several papers on the detection of malicious URLs in addition to different software deployed to block malicious URLs such as Symantec Endpoint Protection [2], McAfee [3], Cisco Umbrella [4]. However, the focus is more on the detection of malicious URLs, which is carried out mostly through the usage of common blacklists that include both benign and malicious URLs, [5] but not on the implementation of a URL defense strategy. Introducing a defense strategy is important and must be implemented in all ML systems since blacklisting cannot detect unknown or new malicious URLs and attackers are capable of modifying the URL's characteristics and tweaking their approach to avoid detection leading to a successful injection of poisoned data into the training dataset.

Attacks targeting ML systems are classified as either Poisoning (Causative) Attacks or Evasion (Exploratory) Attacks, based on the phase during which the attack is launched [6], [7]. Poisoning attacks are launched during the training phase to manipulate, disrupt, or impact the ML system such as Clean Label Attacks, LF attacks, and Backdoor attacks [8]. While evasion attacks are applied during the testing phase with the aim to produce adversary-selected outputs without tampering with the ML model such as Confidence Score Attacks, Gradient Attacks, and Hard Label Attacks [9]. Furthermore, attacks can be classified into targeted attacks, where the attacker changes the behavior of the model on particular instances, or into untargeted attacks, where the attacker aims to impact a model's performance with random instances or scenarios [10]. Several research studies have employed different strategies to implement poisoning attacks including LF attacks through different threat models and usually worst case scenarios are studied, where the attacker is capable of poisoning the training dataset directly aiming to impact the model's behavior. For instance, paper [11] designed a data poisoning attack on an RF-based ML model with the aim of decreasing the model's classification accuracy, while paper [12] proposed poisoning attacks by using label modification functions on ML and Deep Learning (DL) models, along with varying poison rates, to

E. Nowroozi is with the Centre for Secure Information Technologies (ECIT), Queen's University Belfast (QUB), United Kingdom (e-mail: e.nowroozi@qub.ac.uk, ehsan.nowroozi65@gmail.com) - Corresponding Author.

N. Jadalla is with the Bahcesehir University (BAU), Master in Cybersecurity, Istanbul, Turkey. (e-mail: nada.jadalla@bahcesehir.edu.tr)

S. Ghelichkhani is with the University of Leeds, Faculty of Engineering and Physical Sciences Master (Computing), Master in Advanced Computer Science, United Kingdom. (e-mail: samanehghelichkhani@gmail.com)

A. Jolfaei is with the College of Science and Engineering at Flinders University, Adelaide, Australia. (e-mail: alireza.jolfaei@flinders.edu.au)

measure performance degradation.

Over the years, researchers in the ML field mainly focused on listing out existing detection mechanisms [13] and how to enhance them [14], since targeted cyber-attacks are changing regularly with further enhanced attack strategies. Paper [15] conducted a comprehensive survey to identify and fill the gaps in protecting ML systems within different organizations. This survey indicates that organizations are under threat from poisoning attacks more than other attacks however, the right security tools are not being used to protect their systems despite the importance of AI security to the operation of their business [15]. The usage of ML systems is rapidly growing in the software industry however, organizations seem to have a lack of knowledge on securing their ML systems. Top Organizations like Google and Microsoft have called for initiatives to secure ML systems [16], [17], and several studies have been published on applying poisoning attacks and building defense mechanisms through different approaches such as K-LID [10], PEFL [18], and so forth.

The purpose behind our study is to investigate, highlight, and mitigate the vulnerabilities of RF classifiers introduced by LF attacks using corrupted training sets within the domain of ensemble tree-based URL classification and prove the efficiency of the proposed defense mechanism. Through comprehensive analysis of LF attack strategies and their impact on ensemble tree models, we developed an innovative defense method, which is the first mechanism able to identify malicious URLs from benign URLs by detecting manipulated labels, identifying their true label, and ultimately improving the robustness of RF-based malicious URL detection systems. By bridging the gap between backdoor attacks and ensemble tree classifiers, our work contributes to the field of ML security, advancing our understanding of effective countermeasures to defend against highly developed threats in real-world cybersecurity scenarios. Our simulation Python is publicly available at Github [19].

A. Contributions

The contributions of our paper are briefly listed in the following points:

- We initially ran the RF model on six clean datasets [1] to observe the model's performance and detect any future behavioral changes. We obtained 99.87% training accuracy for Dataset 6 and 100% training accuracy for the other five datasets.
- We apply LF attacks on six RF models by applying different poison rates (2-5%) on training datasets and analyzed the LF impact on the model's behavior. The results show that the RF model was not able to detect the attack and the attack successfully fooled the RF model with an ASR higher than 50%. For Dataset 5 [1], we obtained 57.14% ASR with a 4% poison rate and 61.09% ASR with a 5% poison rate.
- We evaluate the effectiveness of applying the *K*-NN method with the aim of detecting poisoned labels and predicting their true label. Then ran the RF model with the recovered datasets and observed an increase in training accuracy. For instance, training accuracy increased from 96.08% to 99.89% for Dataset 5 recovered from 4% poisoned data, and training accuracy increased from 95.19% to 100% for Dataset 5 recovered from 5% poisoned data.

B. Organization

This paper consists of five sections. Section I presents the introduction, which covers the main contributions, motivations, and rationale of our research. Our primary objective is to propose a detection and defense method against adversarial attacks, in specific, LF attacks that target RF classifiers. The remaining sections are organized as follows: Section II presents the related work within our domain. Section III explains the applied methodology in our study including the used datasets, selected classifier, applied LF-based attack, and proposed defense method against such attacks. Section IV presents the results of running clean datasets, poisoned datasets, and recovered datasets. Finally, Section V concludes our research and discusses future work.

II. Related Work

Due to the increase in digitization and high usage of technologies, ML models are susceptible to poisoning attacks causing it to be an emerging research topic over the recent years [20]. For instance, paper [10] designed a white-box attack on a Human Activity Recognition (HAR) by randomly altering the labels from the training dataset using different algorithms including RF, Multi-layer Perceptron (MLP), XGboost, and Decision Tree (DT). This attack has minimized the loss function on true data however, the system may notice some changes in data which limits the attacker's capability to further cause larger damage to the model. While paper [21] applied both untargeted and targeted LF attacks on five different ML models including DT, and RF models. Both attacks have caused a degradation in the accuracy of all ML classifiers and increased the misclassification rate however, the attacker must access the dataset during the training phase to be able to perform the attack. Another study proposed poisoning attacks against IoT fake packet classifiers by using label modification functions on RF-based ML models using different poison rates to illustrate the degradation in the model's performance [12]. In paper [22], a random LF attack is designed on different ML models, including the RF model, with the aim to cause an impact on the accuracy of the models. A minor drop in accuracy of RF was observed within 0-12.5% poisoning rate yet a further increase in poison rate has caused a higher degradation in accuracy rate which means that the model has detected the poisoned labels.

The importance of developing a model-agnostic defense has been today's research topic to detect and defend against poisoning attacks targeting different ML models. Paper [23] introduced several Command & Control (C&C) detection systems to detect sophisticated attacks and reduce false positive rates, however, as network size increases, it is a challenging task to store all network traffic yet, it is an essential requirement of most C&C detection methods to store traffic data. While in paper [24], the aim is to detect poisoned

TABLE I
COMPARISON BETWEEN PREVIOUS WORKS AND OUR'S

Ref.	ML Model	Utilized Datasets	Advantages	Disadvantages
[23]	Different classifiers	10 real-world datasets	- Detects sophisticated attacks - Reduces false positive	- Requires traffic storage
[24]	Linear Classifier	MNIST [25], Spambase [26] and BreastCancer [27]	- Effective Against Label Flipping Attacks - Applicability in Various Scenarios	- Sensitivity to parameters - Scalability
[28]	Linear Classifier	MNIST [25], Spambase [26] and BreastCancer [27]	- Detects attack points and Outlier elimination	- Computationally intensive and requires outlier estimation
[29]	Ensemble Trees	UCI ML Repository [30] & KEEL-dataset Repository [31]	- Achieves high detection accuracy and handles multiple attack types	- Does not locate attacked points and requires untainted data for training
[10]	Ensemble Trees	HAR dataset [32]	- Recovers poisoned data, and increases accuracy	- Limited effectiveness
[33]	RF	Musk2 [34] and Android malware	- Acheived high accuracy under other perturbations and scalable approach	- Ensemble size must be considered and dependent on adversary knowledge
[35]	AdaBoost	Spambase [26], Breast-w [36], Kr-vs-kp [37], and so forth	- Combines weak classifiers into a strong classifier	- Sensitive to noisy & abnormal data
[38]	AdaBoost	MNIST [25]	- Applicable across various ML models & detect flipped labels during the training process	- Learning problem & label flipping budget constraint
[39]	CNN	Drebin [40], Contagio [41], and Genome [42]	-Addresses IoT malware detection - limit to other platforms & defense methodology might not generalize to unseen malware samples.	- Proposing innovative method, called Silhouette clustering-based attack
Our's	RF	URL Datasets in Table III	- High detection accuracy of poisoned labels (See TableVII), and considering attack and defense in Dataset URLs for the first time in computer networks (see TableII and III)	- Future work: Difficult to apply feature positioning attack with compared to label poisoning, since the datasets consist of numerical and lexical features.

labels through applying the K-NN approach and mitigate the impact of LF attacks through label sanitization, however, it assumes a large number of benign samples are available for sanitization, which may reduce classifier accuracy. Another paper suggested to use of an outlier detection-based scheme to identify attack points targeting linear classifiers [28]. However, the application of this scheme can be challenging in large high-dimensional datasets. On the other hand, paper [29] presents how the detection of causative attacks can enhance learning robustness through a two-step secure classification model that uses data complexity measures, yet, these measures calculate the difficulty of classification through computing the geometrical characteristics of data. Furthermore, paper [10] discusses the usage of K-NN defense scheme to predict the true label and recover most of the data leading to a sharp increase in the model's accuracy. Paper [33] proposes a hash-based ensemble approach with the aim of increasing the robustness of RF models taking into account that the size of the ensemble is essential to avoid a drop in a model's accuracy due to the usage of a large ensemble considered to be oversized. Paper [39] presents a Label-based Semi-supervised Defense (LSD) that works by finding the poisoned samples and a Clustering-based Semi-supervised Defense (CSD) that uses clustering techniques to recover true labels. LSD and CSD increase the model's accuracy rate, however, they are considered to have poor performance with reference to other defense methods in terms of speed. In paper [35], a boosting ensemble method for two classifications is used through training a number of weak classifiers using the same training dataset for classifiers to merge into a much stronger classifier. However, noisy and abnormal data can easily impact the model due to its high sensitivity to such data. Paper [38] applies the Regularized Synthetic Reduced Nearest Neighbor (RSRNN) defense method by checking if any of the samples are above the confidence range, then this sample will be considered malicious. RSRNN was able to achieve the smallest test error while also detecting a high portion of malicious samples, however, the problem in learning an SRNN model is similar to that of a K-means problem which is identifying the most relevant K value of samples that will provide the lowest error rate. Table I summarizes different detection and defense methods to protect ML models from LF attacks.

III. METHODOLOGY

In this section, we address the considered datasets, possible LF, and defense methodology.

A. Datasets

This paper used a dataset of 3,980,870 benign and malicious URLs [1], gathered from twelve different datasets. Then six datasets are formed by merging the six benign URL datasets and the six malicious URL datasets in Table II.

TABLE II
THE SELECTED BENIGN AND MALICIOUS DATASETS [1]

No.	Benign Datasets	Malicious Datasets
1	Pristine Alexa	Phishing Site URL
2	Pristine Crowdflower	Phishtank
3	Pristine DMOZ	Malicious data URL
4	Benign set URL	ISCX-URL-2016
5	Non-malicious URL	Phishstrom
6	Pristine ISCX	Malicious set URL

Table III shows the six merged datasets including both benign and malicious URLs labeled as '0' and '1', respectively. To further analyze the datasets, a statistical study identified that malicious URLs within the selected six datasets present a wider range of variability in their length compared to URLs that are considered benign in [1]. However, in our poisoning scenario, we are targeting the labels of both benign and malicious URLs.

TABLE III
THE COMBINED DATASETS WERE OBTAINED FROM [1]

Dataset Name	Combined Datasets
Dataset 1	Pristine Alexa and Phishing Site URL
Dataset 2	Pristine Crowdflower and malicious Phishtank
Dataset 3	Pristine DMOZ and Malicious data URL
Dataset 4	Benign set URL and malicious ISCX
Dataset 5	Non-malicious URL and malicious Phishstrom
Dataset 6	Pristine ISCX and Malicious set URL

In order to achieve accurate results, it is necessary to adjust and restructure the dataset to align with the input format needed for our dependencies. During the preprocessing phase, we apply the interquartile range technique to rescale the data. All duplicates and unprocessed cells containing values are eliminated, along with excluding repetitive hostname URLs. Subsequently, the collections of URLs are mixed up. We extract samples from these datasets for more in-depth analysis. Accordingly, datasets are ready to be used with the ML e.g. RF.

B. Classifier

In this study, RF model is the selected classifier which is a type of supervised learning based on using several decision trees to confirm a single output. The foundation of RF lies in the idea that while each individual tree might make accurate predictions, it is highly likely that some trees will excessively tailor themselves to specific data points, leading to overfitting. To counter this issue, the approach involves combining the outcomes of numerous trees, each excelling and overfitting in distinct ways, which effectively mitigates overfitting. This reduction in overfitting is achieved without compromising the predictive capability of the trees. The key parameters that can be adjusted include n estimators, max features, and pre-pruning settings like max depth. For n estimators, a larger value is recommended to ensure better performance. With an increased number of trees being averaged, the ensemble becomes more resilient by curbing overfitting. However, it is important to note that employing more trees also demands greater memory and training time. The mathematical representation of an RF can be expressed as follows:

Let D represent the original training dataset, and T denote the number of decision trees in the ensemble. For each tree t, a random subset D_t of the training data, D is drawn, typically by bootstrapping (sampling with replacement). Additionally, a random subset of features F_t is selected for each node of the tree, which is a subset of the total features F available in the dataset. The decision tree t is then constructed using the data D_t and features F_t based on a specified criterion, such as Gini impurity or information gain.

During prediction, each tree t in the ensemble independently classifies or predicts a target variable. For classification tasks, the class, y_c, is determined by a majority vote among the trees (mode of class predictions) and can be represented mathematically as:

$$y_c = \text{Mode}(y_t) \text{ for } t = 1, 2, \ldots, T \tag{1}$$

while for regression tasks, the predicted value ,y_r, is obtained by averaging the predictions of all trees and can be represented mathematically as:

$$y_r = \frac{1}{T} \sum_{t=1}^{T} y_t \tag{2}$$

The aggregation of predictions from multiple randomized decision trees helps minimize the overfitting issue and enhances the performance of the model. RF is a powerful and widely used ML due to its robustness and effectiveness, making it applicable in various domains, including classification, regression, and feature importance analysis.

Since our six datasets are a combination of numerical and lexical features, therefore, only decision trees have the ability to handle multiclass classification. Accordingly, RF is selected to be used in our study.

C. LF Based Attack

A random LF attack is applied to the six datasets with the aim of manipulating them by altering both benign and malicious labels randomly. Random LF attacks are a form of adversarial attack, where the attacker seeks to exploit vulnerabilities in the ML by manipulating the input data.

Black box, ***gray box***, *and* ***white box*** attacks are the three main types of attacks of adversarial attacks. In a *black box* attack, the attacker has no access to the system and this is usually considered the worst-case scenario in terms of system accessibility. In a *white box* attack, the attacker has full access to the classifier including its weights and parameters. In this paper, we applied a *gray box* attack, where the attacker has limited access to the classifier during a training phase. The threat model of such attacks is related to the attacker's goal, the attacker's knowledge, and the the attacker's capability to impact the training dataset [43].

a) ***Attacker's goal****:* clarifies the kind of security violation and what error the attacker aims to apply. In this paper, the goal is to manipulate the training dataset used by the RF model and fool the model by concealing the poisoned labels.

b) ***Attacker's knowledge****:* the level of knowledge is determined based on the attacker's access to the feature space, the target classifier, the model parameters, and the training dataset. The attacker in this paper has only access to the training dataset thus, the attacker's knowledge is considered as limited knowledge.

*c) **Attacker's capability**:* This is defined as the degree of control that the adversary possesses over both the training and testing data, as stated in [1].

In this paper, we split the six datasets into 79% for training set and 21% for testing set. Then, applied random LF attack by altering both benign and malicious labels randomly by considering five scenarios: 2%, 3%, 4%, and 5% poisoning rate.

Our attack method is presented in the following Algorithm. As an input, this algorithm takes the RF, original dataset D, and a number of poisoned samples based on the poisoning rate.

- **Input:** Original Dataset $\mathcal{D}$, Poisoned Samples X.
- **Output:** Poisoned Dataset $\mathcal{D}'$, Accuracy Rate.
- For each sample x_i in X:
  - Select X' as random rows from $\mathcal{D}$.
  - Apply LF (a function or operation) on each x_i in X'.
  - Update the poisoned dataset $\mathcal{D}'$ with x_i.
  - Train a model on the poisoned dataset $\mathcal{D}'$.
  - Measure and record the accuracy of the model.

Based on the selected poison rate, X rows are selected randomly from D to flip their labels. Let $D = \{(x_i, y_i)\}_{i=1}^{N}$ represent the original training dataset, where x_i denotes the input data instances, y_i denotes their corresponding true labels, and N is the total number of data points. In a random LF attack, an attacker randomly alters a subset of the true labels as y_i' which is the poisoned label and stored into a poisoned dataset, typically denoted as $D' = \{(x_i, y_i')\}_{i=1}^{M}$, where M represents the number of instances that label manipulated. The manipulation involves changing randomly y_i to y_i', causing the model to learn the potentially erroneous labels.

LF attack is now applied by flipping the label randomly on each X selected row in D. Labels denoted as "0" are flipped to "1" and treated as malicious samples in the training, and labels denoted as "1" are flipped to "0" and treated as benign samples. After executing the random LF process, the flipped labels are stored into D'. The RF is trained using D' then the accuracy of both testing and training is recorded. The attack is triggered by testing the stored model with the labels of the entire flipped testing dataset, which measures the accuracy of the testing. The accuracy rate reported in this scenario reflects the ASR of labels that alter from an adversary of the intended attack approach.

D. Proposed Defense Strategy

The detection and defense method introduced in this paper aims to detect and sanitize the dataset from LF attacks. It works by taking original datasets, D, and untrusted datasets, D', as input. Here we used the K-NN method similar to the approach applied in [24] to detect flipped labels. This method trains a model using the inputs to predict the true label of a given URL sample and checks if it matches the current label of the URL sample. Our defense strategy is built as follows:

*a) **Choose the Best K Value**:* The selection of K is crucial to avoid overfitting or underfitting the model. The value of K determines the number of labels that need to be checked and helps in predicting the value of the tested label.

- **Input:** Original Dataset $\mathcal{D}$, Untrusted Dataset $\mathcal{D}'$, $\mathcal{X}$ values of K.
- **Output:** K.
- For each K in $\mathcal{X}$:
  - For each i:
    - Calculate the distance d as the Euclidean distance between points in datasets $\mathcal{D}$ and $\mathcal{D}'$:

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

    - Determine the mode of the K nearest labels:

$$\text{Mode} = \arg\max_i \left(\sum_{j=1}^{K} I(x_j = i) \right)$$

    - If Mode equals 1, set x_p to 1; otherwise, set x_p to 0.
  - If $L_m = 0$, select K.

This Algorithm determines the best value of K within X set of K values and selects K with respect to the minimum error rate. This algorithm takes an original dataset D, untrusted (poisoned) dataset D', and X set of K values as inputs. In our study, we considered the following X set of K values: $X = \{1, 3, 5, 7, 9, \ldots, 33, 35, 37, \text{ and } 39\}$. For each K value within X set, the distance d of labels in D from the tested label in D' is computed and the mode of K nearest labels is found to confirm the label value. The most relevant K value is selected when a number of mismatch labels, L_m, in D equals to zero to minimize the error rate.

*b) **Defense Method**:* In this study, we considered K-NN as our defense method to mitigate the impact of random LF attack. The K-NN algorithm can be expressed as follows:

Given a dataset D with data points represented as vectors in a d-dimensional feature space, let x_i and y_i be the target data points for which we want to make a prediction or classification and n is the total number of labels in the dataset. The K-NN algorithm estimates distances between x_i and all other data points in D, typically using metrics like the Euclidean distance or Manhattan distance defined as the following:

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3}$$

Our detection and defense method is presented in the following Algorithm. This algorithm takes an original dataset D, an untrusted dataset D', and a value of K as inputs.

- **Input:** Parameter K, Original Dataset D, Untrusted Dataset D'.
- **Output:** True labels x_i, Alarm.
- For each index i:
  - Calculate the distance d as the Euclidean distance between points in datasets D and D':

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

  - Based on d, pick K nearest labels from D.

– Determine the mode of the K nearest labels:

$$\text{Mode} = \arg\max_i \left(\sum_{j=1}^{K} I(x_j = i) \right)$$

– If Mode equals 1, set x_p to 1; otherwise, set x_p to 0.
– If x_p is not equal to x_i:
  * Raise an alarm indicating a potential LF Attack.
  * Update x_i to x_p.

After selecting the value of K using Algorithm **??**, this method measures the distance d of each label y_i in the original dataset with reference to the input sample x_i and then selects the K nearest samples. The algorithm finds the mode of K nearest labels and checks if mode equals 1 then the predicted label, x_p is equal to 1 otherwise if mode equals 0 then, x_p equals 0. Afterward, x_p is compared with the value of x_i and if x_p is not equal to x_i, the system detects the LF attack, activates an alarm to notify the user and recovers the true label by storing x_p into x_i.

The predicted value of the label is identified based on the mode of the K nearest labels. To find the mode, sort the numbers from lowest to highest and observe which label is the most frequently appearing. For example, if the mode of K nearest labels equals to "1" then the predicted label will equal to "1" and if the mode equals to "0", then the predicted label will equal to "0". Finally, the model checks if the predicted label matches the value of the label in the untrusted dataset D'. In case a mismatch is found, it flips the label to recover the true label and clean the dataset from LF attack. This process will be repeated until all labels in the dataset are checked and if any poisoned label is detected, the label is corrected and an alarm is activated to inform the user about the LF attack. Algorithm **??** describes the steps of the applied defense mechanism.

IV. Results and Discussion

In our experiment, we utilized the six different datasets described in Table III [1]. Each dataset consists of 1000 URLs which involves both benign or malicious samples. Benign URLs have a label of "0" while malicious URLs have a label of "1". Different poison rates of LF attack are applied to each dataset through flipping both benign and malicious labels. We ran the original dataset D on RF without any attack. The confusion matrix utilized in our investigation comprised four distinct values, namely: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), which facilitated the visualization of the RF performance. Based on the confusion matrix, several performance metrics are computed by utilizing specific formulas to provide insights into the efficacy of the RF. The following set of metrics was used to calculate accuracy:

A True Positive Rate (TPR), refers to positive URL samples classified as a true class.

$$TPR = \frac{TP}{TP + FN} \tag{4}$$

A True Negative Rate (TNR), refers to positive URL samples that yield to negative test results.

$$TNR = \frac{TN}{TN + FP} \tag{5}$$

A False Positive Rate (FPR), refers to all negative URLs that RF classifier generates it as positive samples.

$$FPR = \frac{FP}{FP + TN} \tag{6}$$

A False Negative Rate (FNR), refers to all positive URLs samples that RF classifier generates it as negative samples.

$$FNR = \frac{FN}{FN + TP} \tag{7}$$

Accuracy is determined by the total number of examples that can be predicted with high reliability, relative to all examples in the six dataset.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

Beside computing the accuracy for each dataset, we asses the performance of the adversary and RF by computing ASR. ASR is calculated by measuring the model's accuracy when tested on a poisoned dataset. A detailed report of the obtained results for each scenario is presented in this section to evaluate both the model's performance and the impact of the utilized attack in addition to the effectiveness of our proposed defense method against random LF attacks.

A. Run RF with Original dataset

To establish a baseline mode for evaluating the attack and defense approach, we initially trained the RF model on original datasets to observe the accuracy behavior when no attack is applied on all six datasets in Table III. The obtained training and testing accuracies on RF using original datasets are reported in Table IV. We can clearly observe from the results when no attack applied to RF model, no drop in accuracy for five datasets except Dataset 6 observed a minor drop that caused the accuracy rate to equal 99.87% however this drop is considered negligible in ML systems. Taking into account that the acceptable range of accuracy drop can vary significantly based on the application, domain, and potential consequences of errors.

TABLE IV
Accuracy of RF model on the six datasets without attack

Dataset	Tr. Accuracy	Te. Accuracy
Dataset 1 ... 5	100.00%	100.00%
Dataset 6	99.87%	100.00%

Considering the original dataset on RF models can be used as a reliable reference point for both the attack and defense scenarios to evaluate model performance. Our aim is to propose a defense mechanism against random LF attacks that can detect manipulated URL labels and enhance the robustness of the RF model.

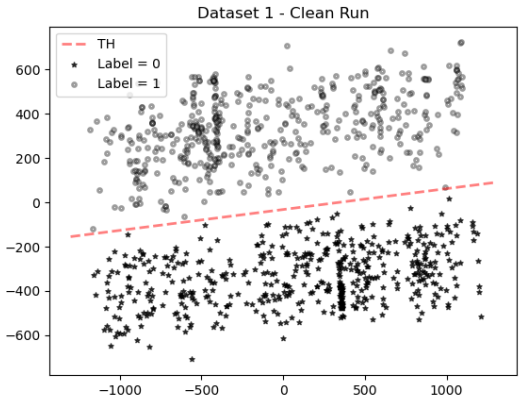


Fig. 1. Distribution of Benign and Malicious labels from original Dataset 1

As shown in Figure 1, Dataset 1 stands as an example of a clean dataset where benign URLs are labeled as "0", and malicious URLs are labeled as "1", both are separate from the decision margin. The red threshold acts as a decision margin that splits the benign samples from the malicious samples and helps to detect any manipulated label.

B. Apply LF attack on RF

By considering Algorithm **??**, we poisoned each dataset with 2% up to 5% causing 16 to 40 URL labels to be poisoned. The results of our random LF attack are reported in Table V.

Based on the results, we can understand that the utilized attack works since the attack barely had a 2% effect on the RF model when 2% of the data is poisoned. Even with introducing a minor increase in the poison rate (from 2% to 5%), the RF model's accuracy drop was restricted to no more than 5% and the impact would not be considered significant since it did not influence the model's accuracy crucially. This approach maintained the attack's covert nature while permitting poisoned backdoor samples to be integrated into the training dataset. The success of the attack was determined by calculating the ASR using all of the testing samples, which revealed the model's classification error rate. Normal testing accuracy measures the correct classification, while ASR measures the classification error. The ASR values surpassed 40%, indicating that this method was successful in bypassing detection by the URL detector and served as a ghost attack, remaining unnoticed in the framework, in line with the study's threshold. For example in Table V, Dataset 1 obtained 98.35% training accuracy and 57.62% ASR with 2% poisoning rate while Dataset 2 obtained 97.97% training accuracy and 58.1% ASR with 2% poisoning rate. Both results present the attacker's success in manipulating Dataset 1 and Dataset 2 which led to fooling the RF model with only 2% poisoned data.

TABLE V
ACCURACY OF RF MODEL ON THE SIX DATASETS WITH RANDOM LF ATTACK

Dataset	Poison%	Poisoned Count	Tr. Accuracy	ASR
Dataset 1	2%	16	98.35%	57.62%
	3%	24	97.09%	50.47%
	4%	32	95.95%	61.09%
	5%	40	97.34%	55.23%
Dataset 2	2%	16	97.97%	58.1%
	3%	24	96.96%	55.23%
	4%	32	96.08%	50.47%
	5%	40	95.18%	64.76%
Dataset 3	2%	16	98.23%	50.48%
	3%	24	97.22%	53.33%
	4%	32	96.2%	56.19%
	5%	40	95.19%	55.23%
Dataset 4	2%	16	97.97%	55.24%
	3%	24	97.47%	54.28%
	4%	32	96.33%	58.09%
	5%	40	95.19%	56.19%
Dataset 5	2%	16	97.97%	62.38%
	3%	24	96.96%	53.33%
	4%	32	96.08%	57.14%
	5%	40	95.19%	61.09%
Dataset 6	2%	16	98.10%	59.05%
	3%	24	96.96%	58.09%
	4%	32	96.08%	52.38%
	5%	40	95.32%	56.19%

To get deeper insights, we generated plots of the poisoned datasets to visualize the manipulated labels. A threshold in red that acts as a decision margin is introduced to differentiate between benign URLs labeled as "0" and malicious URLs labeled as "1". Figure 2 displays Dataset 1 after randomly flipping 5% of the labels, applied on both benign and malicious URLs. The 40 manipulated samples from the 5% poisoned data are observed in this figure as they are inaccurately positioned with reference to the decision margin. We crossed in red the 40 samples in Figure 2 for clear observation.

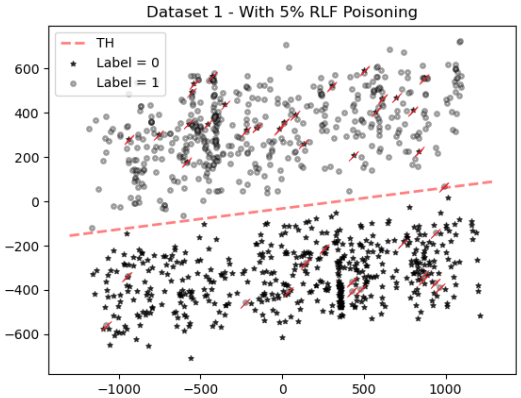


Fig. 2. Illustration of 5% samples poisoned in Dataset 1

C. Apply Defense on RF

We analyzed the effectiveness of our defense mechanism against random LF attacks on RF classifiers. This defense mechanism is based on the K-NN approach that predicts true labels by identifying the K nearest labels from the original

dataset D to each label from the untrusted dataset D'. Initially, we started with identifying the best value of K for each dataset to ensure a minimum error rate in our results using original datasets. Table VI presents the best value of K for each dataset obtained using Algorithm **??** and we can observe that K value can vary for different datasets. For instance, K is set to "33" for Dataset 1 while for Dataset 2, K is set to "5".

TABLE VI
THE BEST VALUE OF K FOR ALL SIX DATASETS

Original Dataset	Value of K
Original Dataset 1	33
Original Dataset 2	5
Original Dataset 3	3
Original Dataset 4	3
Original Dataset 5	3
Original Dataset 6	3

Using the K values, we applied our defense mechanism with the aim to detect the poisoned labels, activate an alarm for each poisoned label, and recovering the value of the poisoned label. After applying the defense mechanism on the entire dataset, this will provide a full recovered dataset from the applied random LF attack.

TABLE VII
ACCURACY OF RF MODEL ON THE SIX DATASETS WITH K-NN DEFENSE AGAINST RANDOM LF ATTACK

Dataset	Poison%	Tr. Accuracy	Det. Poisoned labels
Dataset 1	2%	99.87%	17
	3%	99.87%	23
	4%	100.00%	31
	5%	100.00%	40
Dataset 2	2%	99.87%	18
	3%	100.00%	24
	4%	100.00%	34
	5%	99.87%	40
Dataset 3	2%	100.00%	18
	3%	100.00%	24
	4%	100.00%	31
	5%	100.00%	40
Dataset 4	2%	100.00%	16
	3%	100.00%	20
	4%	100.00%	30
	5%	100.00%	38
Dataset 5	2%	100.00%	18
	3%	100.00%	26
	4%	99.89%	34
	5%	100.00%	42
Dataset 6	2%	100.00%	16
	3%	100.00%	24
	4%	100.00%	32
	5%	99.87%	38

As per the results in Table VII, our mechanism successfully recovered the majority of the poisoned labels, and the training accuracy has increased. For instance, with 5% poison rate, Dataset 3 training accuracy increased after applying the defense from 95.19% to 100% by recovering 40 detected poisoned labels while Dataset 2 training accuracy increased from 95.18% to 99.87% accuracy by recovering 40 detected poisoned labels. This proves the effectiveness of our mechanism in detecting poisoned labels and correcting them to their true label. Taking into account that achieving 100% accuracy in most cases may sound unrealistic, however, this is due to repeatedly training the RF model on the same dataset.

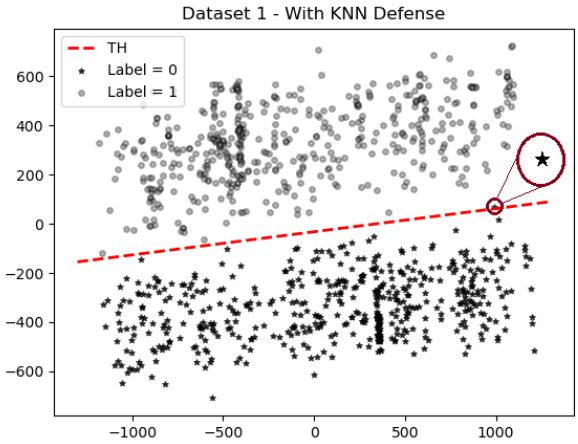


Fig. 3. Recovered Dataset 1 from 5% poisoned samples

For the final analysis, we also generated plots for the recovered datasets obtained after applying the defense mechanism. By comparing the poisoned dataset plot and the recovered dataset plot, we can clearly visualize the effectiveness of our approach. Figure 3 displays Dataset 1, which was initially manipulated by 5% of its labels, that is recovered by our defense mechanism, and all labels are split correctly with reference to the red decision margin. If we compared Figure3 with the original dataset in Figure1, we can observe a slight difference between the two figures where a label on the decision margin is denoted as “0” in Figure 3 however the same label is denoted as “1” in Figure 1.

D. Discussion

We initially started our experiment by running the RF model on original datasets to observe the behavior of the model and 100% training accuracy rate is obtained for five datasets. We set a reference point to measure the impact of the random LF attack and the efficiency of our proposed defense method. Our findings underscored the possibility of an attacker manipulating datasets on RF models through label poisoning, as evidenced by manipulation, a minor degradation is observed in model accuracy between 95-98% is usually negligible in ML systems in addition to achieving ASR in the range of 50.47- 64.76%. Finally, the application of the K-NN defense method demonstrated a way to mitigate the impact of label poisoning, with notable improvements in the model’s robustness through detecting poisoned labels and achieving an accuracy rate in the range of 99.87-100% with the recovered datasets. Taking into account, the vital role of having an original dataset to implement our defense method.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, we shed light on several crucial aspects of random LF attacks against malicious URL detectors and

possible defense strategies through the results of our analysis. Initially, a LF attack is applied through random flipping of benign and malicious labels at a small poisoning rate of 2-5%. According to the results of our study, corrupting the samples within the training dataset during the training phase and achieving ASR between 50-65% within 2-5% poisoning rate makes this attack a more convenient option for an attacker who aims to impact the performance of a model. It is important to note that the LF attack succeeded in fooling the detector and playing the role of a ghost. Accordingly, the implementation of a defense model is crucial to overcome the impact of LF attacks and recover the poisoned datasets. Our defense method has successfully detected the poisoned labels, predicted true labels, and improved the accuracy of the model up to 100%.

Our study has the potential to function as a valuable tool for appraising the robustness of RF models and improving their performance against malicious URLs. The focus of future research will counter other types of backdoor attacks in URL detectors that use the RF classifier to implement possible attack recognition and defense mechanisms. The main objective of these mechanisms will be to identify the existence of backdoors within the training dataset by analyzing the distribution of features in the samples. Further investigations can play a pivotal role in advancing not only our study but also the wider domain of ML security.

REFERENCES

[1] E. Nowroozi, Abhishek, M. Mohammadi, and M. Conti, "An adversarial attack analysis on malicious advertisement url detection framework," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2022.

[2] "Symantec endpoint protection," https://techdocs.broadcom.com/us/en/symantec-security-software/endpoint-security-and-management/endpoint-protection/all/symantec-endpoint-protection-client-for-windows-help/enabling-network-traffic-redirection.html, note = Accessed: 2023-10-04.

[3] "Mcafee," https://www.mcafee.com/en-us/safe-browser/mcafee-webadvisor.html, note = Accessed: 2023-10-04.

[4] "Cisco umbrella," https://umbrella.cisco.com/trends-threats/malware-detection-and-protection, note = Accessed: 2023-10-04.

[5] M. A. Sankaran, S. Mathiyazhagan, M. Dharmaraj *et al.*, "Detection of malicious urls using machine learning techniques," *Int. J. of Aquatic Science*, vol. 12, no. 3, pp. 1980–1989, 2021.

[6] I. M. Ahmed and M. Y. Kashmoola, "Threats on machine learning technique by data poisoning attack: A survey," in *Advances in Cyber Security: Third International Conference, ACeS 2021, Penang, Malaysia, August 24–25, 2021, Revised Selected Papers 3*. Springer, 2021, pp. 586–600.

[7] E. Nowroozi, M. Mohammadi, E. Savaş, Y. Mekdad, and M. Conti, "Employing deep ensemble learning for improving the security of computer networks against adversarial attacks," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2023.

[8] J. Lin, L. Dang, M. Rahouti, and K. Xiong, "Ml attack models: Adversarial attacks and data poisoning attacks," *arXiv preprint arXiv:2112.02797*, 2021.

[9] I. Moisejevs, "Evasion attacks on machine learning (or "adversarial examples")," *Towards Data Science, accessed July*, vol. 21, 2019.

[10] A. R. Shahid, A. Imteaj, P. Y. Wu, D. A. Igoche, and T. Alam, "Label flipping data poisoning attack against wearable human activity recognition system," in *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2022, pp. 908–914.

[11] J. Y. Chang and E. G. Im, "Data poisoning attack on random forest classification model," *Proc. of SMA 2020*, 2020.

[12] C. Dunn, N. Moustafa, and B. Turnbull, "Robustness evaluations of sustainable machine learning models against data poisoning attacks in the internet of things," *Sustainability*, vol. 12, no. 16, p. 6434, 2020.

[13] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 3–14.

[14] J. Gilmer, R. P. Adams, I. Goodfellow, D. Andersen, and G. E. Dahl, "Motivating the rules of the game for adversarial example research. arxiv 2018," *arXiv preprint arXiv:1807.06732*, 1807.

[15] R. S. S. Kumar, M. Nyström, J. Lambert, A. Marshall, M. Goertzel, A. Comissoneru, M. Swann, and S. Xia, "Adversarial machine learning-industry perspectives," in *2020 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2020, pp. 69–75.

[16] "Goodgle advertisement: Responsible ai practices," https://ai.google/responsibility/responsible-ai-practices/, accessed: 2023-08-22.

[17] "Microsoft advertisement: Securing the future of artificial intelligence and machine learning at microsoft," https://learn.microsoft.com/en-us/security/engineering/securing-artificial-intelligence-machine-learning, accessed: 2023-08-22.

[18] X. Liu, H. Li, G. Xu, Z. Chen, X. Huang, and R. Lu, "Privacy-enhanced federated learning against poisoning adversaries," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4574–4588, 2021.

[19] E. Nowroozi, "Ensemble trees: A defense mechanism against label flipping in malicious url detection," https://github.com/ehsannowroozi/LF_Attack_Defense_URL, 2023.

[20] M. Goldblum, D. Tsipras, C. Xie, X. Chen, A. Schwarzschild, D. Song, A. Mądry, B. Li, and T. Goldstein, "Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 1563–1580, 2022.

[21] M. A. Ramirez, S. Yoon, E. Damiani, H. A. Hamadi, C. A. Ardagna, N. Bena, Y.-J. Byon, T.-Y. Kim, C.-S. Cho, and C. Y. Yeun, "New data poison attacks on machine learning classifiers for mobile exfiltration," *arXiv preprint arXiv:2210.11592*, 2022.

[22] F. A. Yerlikaya and Ş. Bahtiyar, "Data poisoning attacks against machine learning algorithms," *Expert Systems with Applications*, vol. 208, p. 118101, 2022.

[23] J. Gardiner and S. Nagaraja, "On the security of machine learning in malware c&c detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 49, no. 3, pp. 1–39, 2016.

[24] A. Paudice, L. Muñoz-González, and E. C. Lupu, "Label sanitization against label flipping poisoning attacks," in *ECML PKDD 2018 Workshops: Nemesis 2018, UrbReas 2018, SoGood 2018, IWAISe 2018, and Green Data Mining 2018, Dublin, Ireland, September 10-14, 2018, Proceedings 18*. Springer, 2019, pp. 5–15.

[25] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[26] R. E. F. G. Hopkins, Mark and J. Suermondt, "Spambase," UCI Machine Learning Repository, 1999, DOI: https://doi.org/10.24432/C53G6X.

[27] M. O. S. N. Wolberg, William and W. Street, "Breast Cancer Wisconsin (Diagnostic)," UCI Machine Learning Repository, 1995, DOI: https://doi.org/10.24432/C5DW2B.

[28] A. Paudice, L. Muñoz-González, A. Gyorgy, and E. C. Lupu, "Detection of adversarial training examples in poisoning attacks through anomaly detection," *arXiv preprint arXiv:1802.03041*, 2018.

[29] P. P. Chan, Z. He, X. Hu, E. C. Tsang, D. S. Yeung, and W. W. Ng, "Causative label flip attack detection with data complexity measures," *International Journal of Machine Learning and Cybernetics*, vol. 12, pp. 103–116, 2021.

[30] "Uci machine learning repository," http://archive.ics.uci.edu/, note = Accessed: 2023-09-26.

[31] J. Alcal-Fdez, A. Fernndez, J. Luengo, J. Derrac, S. Garca, L. Snchez, and F. Herrera, "Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2-3, pp. 255–287, 2011.

[32] A. D. G. A. O. L. Reyes-Ortiz, Jorge and X. Parra, "Human Activity Recognition Using Smartphones," UCI Machine Learning Repository, 2012, DOI: https://doi.org/10.24432/C54S4K.

[33] M. Anisetti, C. A. Ardagna, A. Balestrucci, N. Bena, E. Damiani, and C. Y. Yeun, "On the robustness of ensemble-based machine learning against data poisoning," *arXiv preprint arXiv:2209.14013*, 2022.

[34] D. Chapman and A. Jain, "Musk (Version 2)," UCI Machine Learning Repository, 1994, DOI: https://doi.org/10.24432/C51608.

[35] N. Cheng, H. Zhang, and Z. Li, "Data sanitization against label flipping attacks using adaboost-based semi-supervised learning technology," *Soft Computing*, vol. 25, no. 23, pp. 14 573–14 581, 2021.

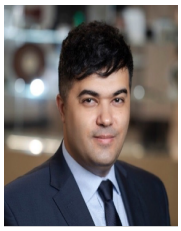
[36] W. Wolberg, "Breast Cancer Wisconsin (Original)," UCI Machine Learning Repository, 1992, DOI: https://doi.org/10.24432/C5HP4Z.

[37] A. Shapiro, "Chess (King-Rook vs. King-Pawn)," UCI Machine Learning Repository, 1989, DOI: https://doi.org/10.24432/C5DK5C.
[38] P. Tavallali, V. Behzadan, A. Alizadeh, A. Ranganath, and M. Singhal, "Adversarial label-poisoning attacks and defense for general multi-class models based on synthetic reduced nearest neighbor," in *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 3717–3722.
[39] R. Taheri, R. Javidan, M. Shojafar, Z. Pooranian, A. Miri, and M. Conti, "On defending against label flipping attacks on malware detection systems," *Neural Computing and Applications*, vol. 32, pp. 14 781–14 800, 2020.
[40] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, and C. Siemens, "Drebin: Effective and explainable detection of android malware in your pocket." in *Ndss*, vol. 14, 2014, pp. 23–26.
[41] "Contagio dataset," http://contagiominidump.blogspot.com/, note = Accessed: 2023-09-26.
[42] Y. Zhou and X. Jiang, "Dissecting android malware: Characterization and evolution," in *2012 IEEE symposium on security and privacy*. IEEE, 2012, pp. 95–109.
[43] C. Wang, J. Chen, Y. Yang, X. Ma, and J. Liu, "Poisoning attacks and countermeasures in intelligent networks: Status quo and prospects," *Digital Communications and Networks*, vol. 8, no. 2, pp. 225–234, 2022.

Ehsan Nowroozi is a leading cybersecurity researcher and Senior IEEE Member/ACM Professional Member with expertise in adversarial machine learning and multimedia forensics. He currently serves as a Research Fellow at Queen's University Belfast's Centre for Secure Information Technologies, where he identifies and addresses vulnerabilities in AI systems. After earning his Ph.D. in Cybersecurity from the University of Siena in Italy, Dr. Nowroozi expanded his knowledge through postdoctoral fellowships at multiple prestigious institutions, such as Siena and Padua Universities in Italy and Sabanci University in Turkey. He also served as an Assistant Professor at Bahçeşehir University in Istanbul, Turkey, 2022-2023. His main research interest is in artificial intelligence for cybersecurity.

Nada Jadalla received her Master's from Bahcesehir University. She received a bachelor's degree in electrical/telecommunication engineering from Ajman University, Ajman, UAE in 2017. Her main research interest is in the area of Machine Learning, Artificial Intelligence and Cybersecurity. She worked in the technology and regulatory field within the telecommunication industry and she is also a student member of IEEE institution.

Samaneh Ghelichkhani received a Master's degree in Advanced Computer Science from the University of Leeds, United Kingdom, and in Information Technology Engineering (Networking branch) from Islamic Azad University. Furthermore received a Bachelor's degree in Information Technology Engineering from Islamic Azad University, Iran. Her main research interest is in artificial intelligence including machine and deep learning, and networks.



Alireza Jolfaei is an Associate Professor in Cybersecurity and Networking at the College of Science and Engineering at Flinders University. He is a Senior Member of the IEEE and a Distinguished Speaker of the ACM on the topic of Cybersecurity. He has previously been a faculty member with Macquarie University, and Federation University in Australia, and Temple University in the USA. He received a Ph.D. degree in Applied Cryptography from Griffith University, Gold Coast, Australia. His main research interest is in Cyber-Physical Systems Security, where he investigates the hidden interdependencies in industrial communication protocols and aims to provide fundamentally new methods for security-aware modeling, analysis, and design of safety-critical cyber-physical systems in the presence of cyber-adversaries. He has been a chief investigator of several internal and external grants. He successfully supervised eight HDR students to completion. He received the prestigious IEEE Australian Council award for his research paper published in the IEEE Transactions on Information Forensics and Security.