Resisting Deep Learning Models Against Adversarial Attack Transferability via Feature Randomization

Ehsan Nowroozi, Senior Member, IEEE, Mohammadreza Mohammadi, Member, IEEE, Pargol Golmohammadi, Member, IEEE, Yassine Mekdad, Member, IEEE, Mauro Conti, Fellow, IEEE, and Selcuk Uluagac, Senior Member, IEEE

Abstract—In the past decades, the rise of artificial intelligence has given us the capabilities to solve the most challenging problems in our day-to-day lives, such as cancer prediction and autonomous navigation. However, these applications might not be reliable if not secured against adversarial attacks. In addition, recent works demonstrated that some adversarial examples are transferable across different models. Therefore, it is crucial to avoid such transferability via robust models that resist adversarial manipulations. In this paper, we propose a feature randomization-based approach that resists eight adversarial attacks targeting deep learning models in the testing phase. Our novel approach consists of changing the training strategy in the target network classifier and selecting random feature samples. We consider the attacker with a Limited-Knowledge and Semi-Knowledge conditions to undertake the most prevalent types of adversarial attacks. We evaluate the robustness of our approach using the well-known UNSW-NB15 datasets that include realistic and synthetic attacks. Afterward, we demonstrate that our strategy outperforms the existing state-of-the-art approach, such as the Most Powerful Attack, which consists of fine-tuning the network model against specific adversarial attacks. Further, we demonstrate the practicality of our approach using the VIPRPrint dataset through a comprehensive set of experiments. Finally, our experimental results show that our methodology can secure the target network and resists adversarial attack transferability by over 60%.

Index Terms—Adversarial attacks, adversarial learning, adversarial machine learning, convolutional neural network, cybersecurity, machine and deep learning, network security.

I. INTRODUCTION

The recent studies on the vulnerabilities of machine and deep learning have attracted researchers’ attention [1]. This new field is known as Adversarial Machine Learning and has been widely investigated [2], [3]. In particular, artificial neural networks that are frequently used in deep learning such as Convolutional Neural Networks (CNNs), including image classification, computer vision, network security, and natural language processing, have raised significant concerns about the vulnerability of these models to adversarial attacks. To that end, we distinguish two different attack strategies in machine learning techniques: poisoning attacks and exploratory evasion attacks. In the poisoning attack, the adversary is aware of the training examples, and the attack is performed throughout the training process, while in the exploratory evasion attack, the adversary compromises the network during the testing phase.

To the best of our knowledge, most machine and deep learning techniques are intrinsically vulnerable to various types of adversarial attacks [4], [5]. Moreover, the adversary can transfer some of these attacks from one network (a.k.a., Source Network (SN)) to another one (a.k.a., Target Network (TN)). As a result, it is necessary to consider preventing this behavior since the transferability property between the networks presents a challenging security issue [6], [7]. Therefore, it is crucial to consider from a research point of view preventing such transferability and thus improving the TN’s security against adversarial attacks.

In this paper, we are motivated to provide a new model to secure the TN and avoid adversarial transferability between the SN and the TN. More specifically, we demonstrate that our model can significantly improve TN’s security. Our main contribution consists of strengthening the TN against exploratory evasion attacks. In particular, we modify the TN’s classifier by considering a Feature Randomization (FR) strategy to extract random features from the SN’s flattened layer. In our work, we experimentally demonstrate the transferability of eight adversarial attacks (with different parameters) from the SN to the TN. Then, we improve the TN’s security via an FR technique by considering two adversarial settings: the case where the adversary has Limited Knowledge (LK) about the TN and the case where the adversary has Semi Knowledge (SK) about the TN. Moreover, we show the performance of our strategy against the MPA approach. For the replication of our experiments, we made our implementation code publicly available [8].

Summary of Contributions: The major novelty and contributions of our work are summarized as follows:

- We propose a novel FR approach that improves the target network’s security against eight adversarial attacks in both
LK and SK conditions. We focus on evasion attacks in the testing phase that try to decrease the false positive rate (i.e., detecting attacked samples as pristine).

- We demonstrate that adversarial attacks may be transferred across the SN and the TN, emphasizing the importance of developing robust models against this property. Our results show that for both shallow and deep networks, most of the considered attacks have an Attack Success Rate (ASR) of more than 95%.

- We design and construct a Support Vector Machine (SVM) in the TN based on the FR strategy and test its robustness under different adversarial settings. By analyzing a wide range of random feature vectors, we find that the feature vectors of sizes 30, 50, and 200 achieve a promising secure model.

- We evaluate the effectiveness of the FR strategy against different adversarial attacks, including the I-FGSM, the FGSM, the BIM, the PGD, the L-BFGS, the JSMA, the DeepFool, and the C&W attack. The experimental results of our study show that the FR methodology is more efficient than the MPA approach, and can defeat adversarial transferability over 60%.

- We validate our approach using the VIPPrint [9] dataset through a comprehensive set of experiments, and show its effectiveness and robustness against different adversarial attacks.

Organization: We discuss the related works in Section II. Section III presents the problem scope and threat model. In Section IV, we describe our proposed approach to improve the security of the TN. In Section V, we report our adversarial attacks with their corresponding attack parameters. We propose the defense mechanisms in Sections VI, VII, and VIII discusses the outcomes of our approach. In Section VII, we demonstrate the practicality of our approach via a case study. Finally, we conclude the paper and present future work in Section IX.

II. RELATED WORK

Several studies in the literature proposed adversarial attacks and countermeasures for deep learning-based networks [10], [11], [12]. In this section, we overview prior works on countermeasures techniques in deep learning models that leverage the randomization strategy. Then, we outline the differences from existing works.

Recent works proposed adding randomization layers to improve the robustness of CNN-based models. In [13], the authors presented a two-layer defense-based neural network; the first layer is a random nullification layer that consists of randomly deleting some features from the input to minimize the adversarial perturbations, while the second layer is an autoencoder-based reconstructor that rebuilds the input features and performs the classification tasks. The numerical results against the FGSM, BIM, JSMA, DeepFool, and C&W attacks show high robustness with an accuracy of up to 80%. Similarly, the authors in [14] proposed a randomization at inference time technique to defeat iterative adversarial attacks. This technique adds two randomization layers at the beginning of the classification neural network. The first layer consists of a random resizing layer, which resizes the input samples randomly. The second layer performs random padding operations for the samples with zeros before its transformation to the CNN model. The experiments performed with such a randomization strategy demonstrate its robustness against the FGSM, the DeepFool, and the C&W attacks. However, additional computations are required by adding the random resizing and random padding layers. Another approach is developed by Taran et al. [15], the authors employ a randomized diversifying mechanism to protect neural networks against various attacks in classification. Such a strategy is implemented in a multi-channel architecture and utilizes a shared secret key between the training and testing stages. The experimental results regarding the randomization diversification mechanism show the robustness against the C&W attack with different parameters. The authors in [16] presented the Randomized Adversarial-Image Input Detection (RAID) for Neural Networks, which relies on building a secondary classifier capable of detecting malicious input based on neuron activation values. The number of monitoring neurons is randomly selected and labeled as activation fingerprints. The authors evaluated the effectiveness of RAID against six adversarial attacks: the PGD, the FGSM, the BIM, the DeepFool, the C&W, and the JSMA attack. The experimental results demonstrate a good detection accuracy of up to 90% for different attacks.

Differences From Existing Works: Different from existing studies, our approach improves the security of the TN and avoids the transferability property by considering eight adversarial attacks. Moreover, our feature randomization strategy relies on changing the TN’s classifier into a Support Vector Machine. Consequently, we decrease the attacker’s knowledge by changing the architecture of the target’s network. In Table I, we show the difference between our work and existing works.

III. PROBLEM SCOPE AND THREAT MODEL

The transferability property is satisfied if the adversarial samples that compromise the SN can be used to fool the TN. In this case, we provide the assumptions regarding the attacker’s capabilities. According to the Deep Learning literature, the attacker can fool CNN-based models in three different settings: white-box, gray-box, and black-box attacks. For each of these settings, the adversary has information about the TN with different levels of knowledge. These settings are classified into three categories [18]: Limited Knowledge (LK), Semi Knowledge (SK), and Perfect Knowledge (PK).

- LK: Commonly known as black-box attacks, and indeed regarded where the attacker cannot access the hyper model parameters, which is a more complex scenario in digital forensics; as a result, the attacker conducts multiple queries to obtain the internal details of the model.

- SK: In this scenario, the adversary has partial information about the victim’s network and performs his attacks under gray-box settings.
TABLE I
DIFFERENCE BETWEEN OUR WORK AND EXISTING WORKS

| Ref. | App Domain | Attacks | Datasets | Type | Case |
|------|------------|---------|----------|------|------|
| [13] | Computer vision | FGSM, JSMA, BIM, DeepFool, and CW | MNIST -Fashion-MNIST | Minimize the effect of adversaries for the trained network. High robustness against adversarial and higher performance on normal samples. | Low performance in CW which is close to 76% |
| [14] | Computer vision | FGSM, DeepFool, and CW | ImageNet | Adversarial examples rarely manage for iterative attacks. High accuracy on clean examples. | Low performance for the network EfficientNetV2 and ResNetV2, (i.e. random, brightness, and brightness++) |
| [15] | Computer vision | CW | MNIST -Fashion-MNIST -CIFAR-10 | Reduce back gradient propagation. | Failed gradient sparses and non-gradient based attacks. |
| [16] | Computer vision | PGD, FGSM, BIM, DeepFool, CW, and JSMA | MNIST -CIFAR-10 | 90% accuracy with the strongest attacks (CW, DP, and excellent detection versus weaker adversaries (e.g., PGD, BIM, and FGSM). | Need to run the tool through its tests with additional threat models. |
| Our work | Computer networks | I-FGSM, FGSM, BIM, PGD, L-BFGS, JSMA, DeepFool, and CW | -UNSW-NB 15 | Improve the security in computer networks domain for Lk and SK scenarios. Avoid a transferability issue [27]. | Need to investigate: Attack transferability in poisoning attacks, and backdoor attacks. |

layer of the CNN network to decrease the attacker’s knowledge and feed them to the SVM to classify them.

B. Threat Model

In real-world applications, the attacker has very limited knowledge of the TN (referred to as the black-box setting). For this reason, we consider our experiments under black-box settings. In this case, with the limited capabilities of the adversary that cannot have access to the TN (e.g., parameters of the model, network architecture), we assume that the adversary has an Lk and can potentially increase his access to the victim’s network to get into an SK scenario. More specifically, the attacker builds the adversarial examples on the SN, which is trained using a different or similar dataset from the TN. Then, the adversary launches different adversarial attacks to fool the TN. In this study, we test and develop a generalized approach for black-box attacks against DL models that take advantage of adversarial example transferability.

IV. METHODOLOGY

In this section, we introduce our novel method to improve TN’s security by including a feature randomization strategy to mitigate adversarial attacks. In what follows, we describe the considered datasets, the learning models, and their parameters, as well as the shallow (N1) and deep network (N2) architectures.

A. Proposed Approach

In our study, we performed eight adversarial attacks with different parameters in black-box settings without accessibility to the model, namely: The JSMA [1], the PGD [19], the L-BFGS [20], the I-FGSM [21], the FGSM [22], the DeepFool [23], the BIM [21], and the C&W attacks [24]. To decrease the adversary’s knowledge, we propose an FR strategy by changing the TN’s classifier into a Support Vector Machine (SVM), as depicted in Fig. 2. For the SN, we consider a Convolutional Neural Network (CNN), while for the target network, we consider the SVM classifier that receives random features from the flattened layer of the CNN network to decrease the attacker’s knowledge and feed them to the SVM to classify them.
we flatten the output of the final layer of CNN in order to get a one-dimensional output. Then, we utilize this output as input for the randomization procedure, and we specify a random amount of it that will be given as input to the SVM for classification.

B. Network Architecture and Learning Models

Due to the wide usability of the networks proposed by Bayer et al. [25] and Barni et al. [26] for many applications such as multimedia forensics, computer vision, cybersecurity, and, in particular, network security [17], we believe that these networks are suitable for our investigations and consider them for constructing the shallow ($N_1$) and deep ($N_2$) network. To demonstrate the applicability of our approach on both $N_1$ and $N_2$, all the processes have been done separately, and these two networks are not connected. In what follows, we describe the $N_1$ and $N_2$ respectively. It is worth mentioning that $N_1$ and $N_2$ are SN and should not be considered as TN.

1) Shallow Network ($N_1$): For the shallow network $N_1$, we consider the architecture presented in [25]. It consists of 3 convolutional layers, namely constrained convolutional layers. This architecture can adaptively learn manipulation detection features directly from the data with high accuracy. Moreover, it outperforms existing image manipulation detection techniques [27], especially when considering real large-scale training datasets. Therefore, we can perfectly use this model as a forensic detector for different image manipulation.

2) Deep Network ($N_2$): For the $N_2$, we consider the architecture proposed in [26]. This network relies on eight convolutional layers and can be seen as a patch-based CNN. This network can detect contrast-adjusted images with good performance in the presence of JPEG post-processing operations. Additionally, it achieves high accuracy under different Quality Factors (QFs).

3) Description of the Datasets: In our study, we consider the UNSW-NB15 [28] dataset generated by the IXIA PerfectStorm application to generate a combination of realistic modern routine operations and synthetic existing attack characteristics. The raw network packets of this dataset are obtained via the tcpdump operations and synthetic existing attack characteristics. The raw application to generate a combination of realistic modern routine

| # of Conv. Layers | $N_1$ | $N_2$ |
|-------------------|-------|-------|
| # of Epochs       | 20    | 10    |
| # of Train Batch  | 64    | 16    |
| # of Validation Batch | 100  | 16    |
| # of Test Batch   | 100   | 100   |
| Optimizer         | Adam, LR=1e-06 | Adam, LR=1e-04 |
| Validation Accuracy | 95.99% | 96.43% |
| Test Accuracy     | 95.86% | 95.42% |

V. Adversarial Attack Strategies

In this section, we report on the experiments regarding the adversarial attacks when $N_1$ and $N_2$ are considered as SN. Then, we present our experimental results for the $N_1$ and $N_2$ networks when they are considered TN.

A. Attack Parameters

In general, we define two types of adversarial attacks against deep learning CNN models: targeted and untargeted attacks. The untargeted attacks enable the trained network to misclassify the input regardless of the output label. In contrast, the targeted attacks aim to deceive a deep learning model by encouraging the model to produce a specific target label for the adversarial sample. For a binary classification task (i.e., which is the case of our study), and since we consider two classes (pristine/manipulate), the targeted and untargeted attacks are similar [29]. Therefore, to apply the attacks on $N_1$ and $N_2$, we perform eight adversarial attacks with different parameters. Each parameter has a different role for different attacks and can influence the performance of the attacks in terms of fooling the DL models. The considered attacks are the most popular adversarial attacks in deep learning:
TABLE III
ATTACK RESULTS ON THE N1 WHEN CONSIDERED AS TN

| Attack Type | PSNR  | L1 dist | Max. dist | ASR  |
|-------------|-------|---------|-----------|------|
| I-FGSM, ϵ = 0.1 | 36.4225 | 3.0011  | 6.0792    | 1.00 |
| FGSM, ϵ = 0.1 | 9.4111  | 80.1985 | 143.4194  | 0.96 |
| JSMA, ϵ = 0.05 | 39.9512 | 0.6003  | 17.85     | 0.72 |
| BIM, ϵ = 0.01 | 18.3842 | 30.0737 | 45.9516   | 0.97 |
| LBFGS, ϵ = 1e-5 | 46.3229 | 0.7998  | 9.4839    | 0.99 |

DeeperPool, default parameter | 45.0145 | 0.897   | 11.8975   | 0.38 |
| PGD, ϵ = 0.05, step size = 0.3, Binary search = true | 18.4940 | 26.4033 | 39.6142   | 0.99 |
| C&W, conf = 0 | 46.2162  | 0.7342  | 10.2935   | 0.99 |
| C&W, conf = 100 | 45.2334  | 0.8051  | 11.4638   | 0.97 |

TABLE IV
ATTACK RESULTS ON THE N2 WHEN CONSIDERED AS TN

| Attack Type | PSNR  | L1 dist | Max. dist | ASR  |
|-------------|-------|---------|-----------|------|
| I-FGSM, ϵ = 0.1 | 37.4549 | 2.6206  | 5.9534    | 1.00 |
| FGSM, ϵ = 0.1 | 28.1658 | 23.1140 | 41.0907   | 0.69 |
| JSMA, ϵ = 0.01 | 54.2103 | 0.04665 | 13.3155   | 0.96 |
| BIM, ϵ = 0.01 | 43.8285 | 1.8659  | 2.5848    | 0.99 |
| LBFGS, ϵ = 1e-5 | 61.0327 | 0.0760  | 3.0495    | 1.00 |
| DeepPool, default parameter | 59.1931 | 0.1198  | 4.5238    | 0.58 |
| PGD, ϵ = 0.05, step size = 0.3, Binary search = true | 44.3092 | 2.0654  | 2.9510    | 0.98 |
| C&W, conf = 0 | 61.3460  | 0.0460  | 4.6834    | 0.96 |
| C&W, conf = 100 | 59.9532  | 0.0650  | 4.8708    | 1.00 |

The I-FGSM, the FGSM, the BIM, the PGD, the L-BFGS, the JSMA, the DeepPool, and the C&W attack. To produce the attack samples, we selected 500 samples randomly from a malicious test dataset; it is obvious that if we select samples from the training dataset, we cannot fool the models, as the networks and models that have considered these data can detect them easily. Afterward, we applied the abovementioned adversarial attacks on the selected samples using the Foolbox library [30]. Then, we fed these samples to N1 and N2 in order to measure the success rate of each attack. We mention that the 500 selected samples are different for N1 and N2, and were randomly selected.

B. Experimental Results

We consider several parameters during our experiments to evaluate our models. In particular, we compute the PSNR, L1 distortion, maximum absolute distortion, and Attack Success Rate (ASR) averages for each of the eight considered adversarial attacks. We define ASR as m/n where m is the number of attacked samples that successfully fooled the model, and n is the number of all the attacked samples. Then, we report the results in Tables III and IV for the N1 and N2 when considered as TN.

1) Experimental Results on the N1: In N1 with a test accuracy of 95.86%, we remark that the average PSNR is less than 46 dB and more than 80% of adversarial attacks succeeded with a high ASR. Given the number of convolutional layers in N1, the reported experimental results in Table III are expected. This could be explained due to the inherent vulnerability of machine learning models.

2) Experimental Results on the N2: In N2 with a test accuracy of 96.42%, we notice that the adversary can successfully fool the network, even when considering a high number of convolutional layers. The CNNs are generally vulnerable to adversarial attacks when considered as TN. However, these networks have high classification accuracy. The experimental results illustrated in Table IV demonstrate that most of the eight adversarial attacks have an ASR of more than 90% and the transferability property is satisfied (i.e., the attacker can transfer the samples from the SN to the TN). Therefore, it is crucial to address this problem by providing suitable defense mechanisms that are efficient against well-known adversarial attacks.

VI. ADVERSARIAL DEFENSES

In this section, we present and evaluate two different adversarial defense methods to improve TN’s security: the MPAs approach and the FR approach. The MPA approach is one of the adversarial defense mechanisms that has been recently considered to provide security for ML models. It consists of resisting the networks by fine-tuning the models [31]. On the other hand, the FR approach aims to decrease the adversary’s knowledge of the TN and place the adversary in LK or SK scenario by selecting various features for the classification task (i.e., selecting a random number of features from the full feature space of the flattening layer). In what follows, we describe each of these methods.

A. Most Powerful Attacks (MPAs)

The rationale behind the MPA approach is to secure the TN by resisting it against the Most Powerful Attacks (MPAs) (i.e., the attacks that significantly reduce the model’s accuracy). By leveraging MPA, with a high probability, we obtain a secure model against weaker attacks, thus, making the MPA approach efficient as it is not feasible to resist the detectors against all existing attacks [31]. The MPA approach consists of importing the attack samples into the training set that allows the decision margin to be refined. However, applying these new attacks to the detector is challenging. In [31], the authors proved that the choice of samples in the processing tools used for training is most effective in disabling the classifier’s performance. As seen in Fig. 3, the star samples are pristine, the circle samples are manipulated, and the solid line illustrates the decision margin before fine-tuning. The dots show the attacks that models were fine-tuned based on them. After fine-tuning, we observe that the decision margin completely changed (dotted line) and became closer to the pristine data; this new decision margin provides an extremely challenging situation for an attacker to cross the line as finding a gap between pristine data and decision margin is quite difficult. On the other hand, the experimental results in [4] demonstrated that adding MPAs samples to the training set enables a good performance in the presence of a wider variety of attacks and processing. Although this strategy considers the SVM classifier and is already used for cybersecurity in Multimedia Forensics, we apply this approach in the context of computer networks and DL models. This is quite a novel use
case of ML/DL models, which are usually considered due to their suitability and performance [17].

In [4], [31], the authors demonstrate most ML models, particularly deep learning networks, are inherently vulnerable and fragile against adversarial attacks. These vulnerabilities are critical in security-oriented applications given their negative impact on the performance of the models. In the MPA approach, we separately fine-tune $N_1$ and $N_2$ models with each attack sample. As we have eight types of attacks, we obtain eight newly tuned models for each of $N_1$ and $N_2$. Then, we provide as input to each fine-tuned model the attack samples of other attacks to predict their labels (pristine or manipulate). Afterward, we gather and save the corresponding ASR. With this method, we can determine which fine-tuned model has more resistance against all the other attacks. For instance, considering the TN network $N_1$ fine-tuned by an attack $A_1$ that is tested against an attack $A_2$ where the ASR is 10%; it can be said that the fine-tuned model with $A_1$ can detect 90% of $A_2$ attack samples and can be evaded with 10% of them.

1) Experimental Results for the MPA Approach: We report the score results of the MPA approach for $N_1$ and $N_2$ models in Tables V and VI, respectively. Note that all the other scores of the MPA approach are equal to 1 except the reported results. For $N_1$, the tested results of the tuned adversarial attacks achieve a high score value for the attacks: I-FGSM010, FGSM010, BIM100, PGD005, L-BFGS, JSMA001, DeepFool, CW0, and CW100. Regarding the fine-tuned PGD005 attack, we remark that the test results have low scores for most of the tested attacks. However, for the deep network $N_2$, all the fine-tuned attacks have a high score value when tested against the eight adversarial attacks. According to the results, approximately all of the tuned models of the attacks were secure against other attacks; however, for verifying the MPA method, we need to apply the eight attacks on the fine-tuned models to verify if they can fool again the networks.

2) Security Level Evaluation for the MPA: After testing each of the tuned models with other adversarial attacks, we observed that all of the considered attacks achieved good results. However, by applying each attack again on the tuned networks, we remark that the adversary can perfectly fool the tuned networks. Therefore, we can confirm that the MPA approach is not efficient in securing $N_1$ and $N_2$ against adversarial attacks. In this case, we consider in what follows another strategy based on the FR technique.

B. Features Randomization (FR)

In this approach, our goal is to provide a limited knowledge condition for the attacker by selecting random features vectors of size ($F < N$), where $F$ is the size of the random features vector and $N$ is the size of the full features vector extracted from the SN flatten layer. Then, we give the TN the selected random features vector as input to perform the classification task (pristine or manipulate). This method satisfies the Limited-Knowledge setting as the attacker cannot guess the selected random features to perform adversarial attacks to fool the TN. Even when the whole feature set is examined ($F = N$), the detector differs from the classification architecture of the SN. In this case, as the SN and TN are similar, the attacker does not know the architecture of the

| $N_1$ MPAs Score Results | Tested with |
|--------------------------|-------------|
| I-FGSM010 | 0.99 |
| FGSM010 | 0.99 |
| BIM100 | 0.99 |
| PGD005 | 0.98 |
| L-BFGS | 0.99 |
| JSMA001 | 0.99 |
| DeepFool | 0.98 |
| CW0 | 0.01 |
| CW100 | 0.004 |

| $N_2$ MPAs Score Results | Tested with |
|--------------------------|-------------|
| I-FGSM010 | 0.99 |
| PGD005 | 1 |

Fig. 3. Representation of the MPA approach for an adversary-aware classifier. The generation of adversarial samples (crosses) enables decreasing the region of the benign samples (stars), thus challenging the obfuscation of dot samples as the star ones.
TN. Moreover, the amount of attacker’s information regarding the SN is smaller than in the MPA approach. To evaluate the FR approach with different random feature vectors, we consider different feature space sizes to identify which size space would provide higher security for the TN.

1) Experimental Results for the FR Approach: To implement the FR approach, we defined random feature space sizes by \( F = \{5, 10, 30, 50, 200, 400, N\} \), where \( N \) represents the flat-layer’s full features size of \( N_1 \) and \( N_2 \), which are 1,728 and 3,200 respectively. In this study, we considered the SVM as TN to train it with random feature vectors. For this purpose, we randomly selected 120,000 samples for training (60,000 for pristine and 60,000 for manipulated), 10,000 samples for validation (5,000 for pristine and 5,000 for manipulated), and 20,000 samples for testing (10,000 for pristine and 10,000 for manipulated). Given the computation cost challenges, we note that the SVM is not trained with all 223,633 samples. Then, we fed the training, validation, and testing samples to SN and extracted full feature vectors of the samples from the flattened layer of the SN. Afterward, we selected random feature vectors from the full features vector 50 times for different sizes of \( F \) (i.e., at the end of this process, we had 50 different feature sets (training, validation, and test) for each \( f \in F \)). These feature sets are employed to train 50 SVMs for each \( f \in F \).

To better illustrate the FR approach, we assume \( f = 10 \). To provide data for SVMs training, we consider 120,000 samples of training, 10,000 samples of validation, and 20,000 samples of the test that we feed to \( N_1 \). Then, we extract the features of its flattened layer (full features vector). Afterward, as the full features vector size is 1,728 for \( N_1 \), we randomly select the features vector of size \( f = 10 \) from \( N = 1728 \) for each sample 50 different times, and which will be used to train 50 different SVMs. We repeat the same procedure for \( N_2 \).

When considering the SVMs as TN, we employed Radial Basis Function (RBF) kernel [32] which has two hyper-parameters \( C \) and \( \gamma \) to control the error of classification and to give curvature weight of the decision boundary, respectively. As the hyper-parameters should be set before training the model, for finding the best \( C \) and \( \gamma \) for each SVM, we performed a 5-fold cross-validation via grid search. Then, we used the found hyper-parameters to train the SVM models. Next, we evaluated the SVMs with the test set features that did not include attacked samples. We ran our experiments on a computer with an Intel(R) Core i7 - 10 and 11 generation CPU with 32 GB of RAM. In addition, we performed the training and testing processes of all the SVMs using the LiBSVM library package [33]. In Table VII, we report the average accuracies of 50 SVM models for each \( f \in F \).

To test the performance of the trained SVMs against attacked samples, we followed the same approach for getting the features from SN’s flattened layer (i.e., we gave each attack sample to SN and extracted the features from the flattening layer). Then, we randomly selected 50 different feature sets for each feature vector of size \( f \in F = \{5, 10, 30, 50, 200, 400, N\} \). We considered 500 samples for each attack, including I-FGSM010, FGSM010, BIM100, L-BFGS, JSMA001, JSMA, DeepFool, C&W0, and C&W100. For example, we assume \( f = 30 \) and select the FGSM010 attack. After applying the random feature selection procedure, we obtain 50 different feature vectors of size \( 500 \times 30 \) for the FGSM010 attack that is ready to test the SVM models. To test the SVMs, we considered two methods with different knowledge levels for attackers: mismatch index testing and match index testing. In the mismatch index testing, the adversary has a Semi-Knowledge, while in the match index testing, the adversary has Semi-Knowledge. In our study, we assume that TN’s model is secure if it can detect adversarial attacks with an accuracy of more than 60%.

Mis-match index testing: In this procedure, the attacker has an LK condition due to the absence of knowledge regarding the TN model, the parameters of the TN model, and the random indices used for classifying. The attacker knows only the feature size (i.e., \( f \in F \)). Therefore, we tested each SVM model with the 50 randomly selected feature vectors of each attack. Then, we summed the scores of these SVMs. As we have 50 SVMs, we calculated their average scores. In Algorithm 1, we present the mismatch index testing algorithm. In a mismatch index testing scenario, we considered \( N_1 \) as SN, the SVMs as TN, and we trained the SVMs by the randomly selected features vector. In Table VIII, we provide the numerical results of the mismatch index testing. According to the reported results, it can be clearly shown that we achieved good results for the random feature sizes of 30 and 50. Moreover, in the full feature size \( N = 1728 \), the adversary knows all the indexes, which is reasonable to obtain an insecure model. To that end, we state that using the random feature selection technique can increase the security model to some extent.

Similarly, we applied the mismatch index testing when considering \( N_2 \) as SN, and we report our numerical results in Table IX. The results show that we obtain good results for the random feature size of 200. Additionally, in the full feature case, the adversary knows all the indexes, which is also reasonable to have an insecure model. Therefore, by using the random feature

| Size of Random Feature Vectors | 5   | 10  | 30  | 50  | 200 | 400 | N  |
|-------------------------------|-----|-----|-----|-----|-----|-----|----|
| \( N_1 \)                     | 95.14 | 95.91 | 96.02 | 96.01 | 96.01 | 96.01 | 96.01 |
| \( N_2 \)                     | 93.75 | 95.83 | 95.90 | 96.13 | 96.13 | 96.16 | 96.18 |

| Size of Random Feature Vectors | 5   | 10  | 30  | 50  | 200 | 400 | N  |
|-------------------------------|-----|-----|-----|-----|-----|-----|----|
| I-FGSM010                    | 51.07 | 58.59 | 76.23 | 75.08 | 60.46 | 72.09 | 0.00 |
| BIM100                       | 45.34 | 51.54 | 61.18 | 64.01 | 57.67 | 68.51 | 0.00 |
| Pgd005                       | 45.43 | 51.43 | 62.01 | 65.37 | 58.5 | 69.29 | 0.00 |
| L-BFGFS                      | 50.78 | 58.37 | 76.24 | 75.4 | 60.3 | 72.15 | 0.00 |
| JSMA001                      | 50.51 | 57.88 | 75.47 | 74.63 | 60.45 | 72.00 | 0.00 |
| DeepFool                     | 49.56 | 56.74 | 72.54 | 72.66 | 60.14 | 71.82 | 0.00 |
| CW100                        | 50.82 | 58.34 | 75.98 | 75.18 | 60.31 | 72.11 | 0.00 |
| CW0                           | 50.76 | 58.31 | 76.05 | 75.29 | 60.31 | 72.11 | 0.00 |
| FGSM010                      | 51.07 | 58.59 | 76.23 | 75.08 | 60.46 | 72.09 | 0.00 |
selection, we can increase the security level of the TN to some extent.

Match Index Testing: In this scenario, the attacker has an SK condition. In other words, the TN classifier and its parameters are not available for the adversary. However, the information of selected indices and the feature vector size (i.e., \( f \in F \)) are accessible to the attacker. In this method, we tested each of the SVM models of feature size \( f \) with one random feature vector of each attack with the same index instead of 50 random feature vectors. In what follows, we provide the Algorithm 2 of match index testing.

In Tables X and XI, we report the numerical results in the match index testing of \( N_1 \) and \( N_2 \) when considered as SN models and SVM as TN, respectively. According to the presented results, we claim that if the adversary has an SK on the features, it will be challenging to fool the TN. In fact, for the match index testing of the \( N_1 \), choosing the SVM between 200 and 400 will provide a security level of the TN by more than 60% for all the considered adversarial attacks. For \( N_2 \), we notice that the SVM of size 200 enables a good security level for TN with more than 61% for all the adversarial attacks.

2) Security Level Evaluation for the FR Approach: According to the reported results, we observe that the mismatch index case is more robust than the match index case. This could be explained by the fact that each model was tested with 50 files, which is more likely to be secure against different attack features. Therefore, the FR approach’s experimental results demonstrate that the TN security level is improved compared to the MPAs approach.
VII. Case Study

To demonstrate the practicality of our approach in different domains of artificial intelligence, we conducted comprehensive experiments using the widely recognized VIPPrint dataset [9]. This dataset has been extensively utilized in the context of multimedia forensics and security, with a primary focus on facial recognition. Our selection of facial images was done due to several reasons. First, facial images are highly available, enabling us the opportunity to generate both digital and printed samples. Second, the impressive capabilities of Generative Adversarial Networks (GANs) in producing highly synthetic facial images [34] have made them almost indistinguishable from real samples. In this case, the features of the real and fake samples are very similar. Thus, making the detection task more challenging. For a structured evaluation in our case study, we divided our datasets into training, validation, and testing. Then, we trained two neural networks ($N_1$ and $N_2$) with the VIPPrint dataset, similar to the UNSW-15 datasets. Note that we did not modify the training hyperparameters such as the number of epochs, learning rate, and optimization strategy. This would enable a smooth comparison of our method in different domains. We achieved a testing accuracy of 90.68% and 92.10%, for $N_1$ and $N_2$, respectively. Regarding the validation accuracy, $N_1$ and $N_2$ models achieved 90.88% and 92.45%, respectively.

We observe a drop in validation and testing accuracy when compared to the UNSW dataset. This decrease is expected, given that the VIPPrint dataset is recognized for its challenging datasets in real-world contexts in multimedia forensics and security. As such, the validation accuracy of approximately 90% for $N_1$ and 92% for $N_2$ is an acceptable outcome from our perspective. Regarding the TN, with the use of SVM as the classifier, we followed a similar methodology. We followed a similar strategy for the target network using the SVM as the classifier. In this scenario, we considered an identical set of random feature sizes [5, 10, 30, 50, 200, 400, and N (full dimension)] for each SVM scenario and 50 iterations of SVM models trained for each SVM case. Table XII shows the average accuracy across 50 SVM models for each random feature case.

In the absence of adversarial attacks, when training the SVMs using different sizes of random feature vectors, as we increase the size of the random feature vectors from 5 to 400, the accuracy of both classifiers, $N_1$ and $N_2$, generally improves. This increase infers that higher-dimensional feature vectors contribute to better discrimination and classification performance. Moreover, when comparing $N_1$ and $N_2$, we remark that $N_2$ consistently exhibits higher accuracy than $N_1$ across all feature vector sizes. This indicates that $N_2$ is more effective than $N_1$ for identifying and classifying VIPPrint samples, given that $N_2$ is deeper than $N_1$ in terms of convolutional layers. In order to test the TN considering VIPPrint datasets, we considered match and mismatch indexing scenarios. Further, for $N_1$ and $N_2$ networks, we considered only three different adversarial attacks: I-FGSM010, JSMA001, and CW100. In Table XIII, we show the percentage of mismatch results for $N_1$ model in the presence of the considered adversarial attack strategies.

We remark from Table XIII, an increase of the percentage of the mismatch index as the size of random feature vectors increases. This increase explains that larger feature vector sizes (i.e., 50, 200, and 400) are more likely to result in higher robustness under different adversarial attacks. Similarly, we provide in Table XIV the percentage of mismatch index results for $N_2$ model for different adversarial attack scenarios.

Likewise in the previous table, we remark in Table XIV an increasing percentage of the mismatch index as the size of random feature vectors increases. Therefore, the perturbations are unlikely to happen when increasing the size of random feature vectors (i.e., 400). When comparing these results to the ones presented in Table XIII, we remark that the $N_2$ model exhibits a lower percentage of mismatch index values than the $N_1$ model, which indicates that the $N_1$ model is secure against adversarial attacks when the size of random feature vectors is 50, 200, and 400, while the $N_2$ model is secure only when the size of random feature vectors is 400. Thus, making it more challenging for the adversary to compromise the $N_2$ model. In the match index scenario when the adversary possesses the SK condition, we utilized Algorithm 2 for match index testing for feature randomization. Here, our goal is to assess the behavior of TN when the adversary lacks access to both the TN classifier and its associated parameters. Table XV presents the match index percentage for the $N_1$ model, where VIPPrint is the dataset under consideration.
In the same way, the percentage of the match index for $N_1$ model increases as the size of feature vectors (i.e., 50, 200, and 400). Therefore, the choice of feature vector size can impact the model’s ability to correctly classify VIPPrint samples under different adversarial attacks. Moreover, when comparing the percentage of the match index values of the $N_1$ model to the mismatch index values for the $N_1$ model, we can observe that the $N_1$ model in both scenarios can accurately classify VIPPrint samples for the feature vectors size of 50, 200, and 400, regardless from the adversarial perturbations. We provide in Table XVI the experimental results for the match index regarding the $N_2$ model.

According to the results reported in Table XVI, we remark that the feature vector size 400 has the highest percentage rate. Therefore, indicating the resilience of $N_2$ model against different adversarial attacks for the feature vector size of 400. In this case study, we experimentally demonstrated via the VIPPrint dataset in both match and mismatch index. More specifically, to obtain a secure model, we can construct the target network as SVM based on feature randomization strategy, and through specific network configuration of the feature vector size.

### VIII. Discussion

In this work, we applied two different defense strategies against eight adversarial attacks with different parameters under black-box settings, namely: The JSMA, the PGD attack, the L-BFGS attack, the I-FGSM attack, the FGS attack, the DeepFool attack, the BIM attack, and the C&W attack (with the strength parameters 0 and 100). The first strategy is based on the MPA (i.e., the source and target network architectures are similar), while the second defense mechanism is based on the features randomization technique (i.e., the source and target network architectures are different).

Given that prior works demonstrated the robustness of the MPA approach against adversarial manipulations, notably in the category of the adversary-aware detector. We imported the MPA approach to our study to evaluate its performance using the fine-tuning technique in the field of network security. However, we found that such an approach is inefficient, and the TN model cannot resist adversarial transferability. In this context, we proved that all eight considered adversarial attacks were transferred from the SN to the TN.

To that end, we developed a novel strategy based on the feature randomization technique. In this case, we investigated the potential of utilizing FR to increase the resilience of DL models against adversarial cases by limiting attack transferability. We applied our approach in a wide range of scenarios, demonstrating that the FR approach can significantly reduce the transferability of adversarial attacks, thereby enhancing the security of the DL models. Even though, in certain circumstances, we found that the mismatch in structure between the SN and the TN is sufficient to prevent adversarial transferability. Additionally, our investigations demonstrated that for a small size of random feature vectors, the complexity increases in the training data for the TN. This complexity could be explained by the high amount of random cases, which can decrease the adversary’s awareness of training data. Therefore, the TN can resist more adversarial attacks and prevent attack transferability. Interestingly, we decreased the attacker’s knowledge in the FR approach by changing the TN architecture and employing a TN model different from the SN model. Consequently, the attacker has limited knowledge of the TN model and its parameters. For instance, we considered the SVM as TN, which had a different architecture from SN models ($N_1$ and $N_2$). In this case, we decreased the adversary’s knowledge of the LK condition. Therefore, the attacker must perform deep searches to determine TN’s architecture and parameters. Moreover, we demonstrated that the FR approach is efficient in the SK scenario (i.e., the attacker has only information about the randomly selected indices for training the TN and their feature vector size). Accordingly, we analyzed the SK scenario through match index testing. Our numerical results demonstrated that the DL models are secure against attack transferability issues when the adversary is under SK conditions. Furthermore, we validated our approach by applying our strategy to the VIPPrint dataset. Our experimental results show the effectiveness of the FR strategy implemented through SVMs against different adversarial attacks, outperforming the MPA approach. Finally, we evaluated the computational cost of the attacks for 500 samples with the trained networks. As shown in Table XVII, we remark that the computational cost of the C&W attack is

| Attack Type  | $N_1$ | $N_2$ |
|--------------|-------|-------|
| L-FGSM010    | 1500  | 500   |
| FGSM010      | 300   | 500   |
| ISMA001      | 5000  | 3600  |
| BIM100       | 500   | 500   |
| L-BFGS       | 9000  | 6000  |
| DeepFool     | 500   | 500   |
| PGD005       | 3000  | 1800  |
| CW0          | 59000 | 45000 |
| CW100        | 60000 | 47000 |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
significantly higher than other adversarial attacks in the SN and TN. Further, we also evaluated the runtime for training 50 SVM based on the features extracted from $N_1$ and $N_2$ in the FR approach. As depicted in Table XVIII, the results show that the computational cost increases as the number of features increases.

IX. CONCLUSION AND FUTURE WORK

Over the past decades, the increasing applications of machine and deep learning have triggered the need to consider its robustness against adversarial attacks. In this context, the adversary can craft malicious samples and transfer the adversarial attacks from the SN to the TN. To avoid the transferability property, it is crucial to improve the target network’s security. In this paper, we investigated evasion attacks on ML/DL models applied in the testing phase. We leveraged the potential of utilizing the feature randomization technique to increase the resilience of DL models against adversarial samples, thus impeding attack transferability. Our experimental results in LK and SK conditions demonstrated that the FR approach could significantly reduce the transferability of adversarial attacks, thereby protecting the TNs from adversarial manipulations. We also showed that in some cases, the architectural difference between the SN and the TN is satisfactory to avoid adversarial transferability. Our future work will focus on poisoning attacks and their transferability. In particular, where an attacker may poison the training data by inserting precisely selected samples, ultimately threatening the entire learning process. The poisoning process can thus be viewed as malicious contamination of the training data. Further, we aim to design a defense mechanism against backdoor attacks for ML/DL models. These attacks look for high correlations in the training data without checking for causative variables.

REFERENCES

[1] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” in Proc. IEEE Eur. Symp. Secure Privacy, 2016, pp. 372–387.

[2] N. Akhtar and A. Mian, “Threat of adversarial attacks on deep learning in computer vision: A survey,” IEEE Access, vol. 6, pp. 14410–14430, 2018.

[3] D. Wang, R. Wang, L. Dong, Y. Zhang, X. Zhang, and Y. Gong, “Adversarial examples attack and countermeasure for speech recognition system: A survey,” Comput. Comput. Inf. Sci., vol. 1268, pp. 443–468, 2020.

[4] E. Nowroozi, A. Dehghantanha, R. M. Patrizi, and K.-K. R. Choo, “A survey of machine learning techniques in adversarial image forensics,” Comput. Secur., vol. 100, 2021, Art. no. 102092.

[5] E. Nowroozi, M. AbbsheshMohammadi, and M. Conti, “An adversarial attack analysis on malicious advertisement URL detection framework,” IEEE Trans. Netw. Service Manag., vol. 20, no. 2, pp. 1332–1344, Jun. 2023.

[6] Y. Wang, J. Liu, and X. Chang, “Assessing transferability of adversarial examples against malware detection classifiers,” in Proc. ACM Int. Conf. Comput. Comput. Front., 2019, pp. 211–214.

[7] M. Barni, K. Kallas, E. Nowroozi, and B. Tondi, “On the transferability of adversarial examples against CNN-based image forensics,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2019, pp. 8286–8290.

[8] E. Nowroozi, “Feature randomization for improving the security of deep net in network security,” 2022. [Online]. Available: https://github.com/ehsannowroozi/RDFS_SU

[9] A. Ferreira, E. Nowroozi, and M. Barni, “VIPPrint: Validating synthetic image detection and source linking methods on a large scale dataset of printed documents,” J. Imag., vol. no. 7, 2021, Art. no. 50.

[10] C. Oh, A. Xompero, and A. Cavallaro, “Chapter 15 – visual adversarial attacks and defenses,” in Advanced Methods and Deep Learning in Computer Vision, ser. Computer Vision and Pattern Recognition, E. Davies and M. A. Turk, Eds., San Francisco, CA, USA: Academic Press, 2022, pp. 511–543.

[11] Y. Guo, X. Han, Y. Zhang, W. Wang, and B. Wang, “Text adversarial attacks and defenses: Issues, taxonomy, and perspectives,” Secur. Commun. Netw., vol. 2022, 2022, Art. no. 6458488.

[12] V. W. Andelli, Y. Deldjoo, T. DiNoia, and F. A. Merra, Adversarial Recommender Systems: Attack, Defense, and Advances. Berlin, Germany: Springer, 2022, pp. 335–379.

[13] M. Ashrafamiriz, S. M. P. Dinakarrao, A. H. A. Zargari, M. Seo, F. Kordahi, and H. Homayoun, “RA2D: Randomization and reconstructor-based adversarial defense for deep neural networks,” in Proc. ACM/IEEE 2nd Workshop Mach. Learn. Learn. CAD, 2020, pp. 21–26.

[14] X. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, “Mitigating adversarial effects through randomization,” 2017, arXiv:1711.01991.

[15] O. Taran, S. Rezaeifar, T. Holotyak, and S. Volodymyrovskyi, “Defending against adversarial attacks by randomized diversification,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 11218–11225.

[16] H. F. Eniser, M. Christakis, and V. Wulzholz, “RAID: Randomized adversarial-input detection for neural networks,” 2020, arXiv:2002.02776.

[17] Z. Wei, J. Chen, Z. Wu, and Y.-Q. Jiang, “Boosting the transferability of video adversarial examples via temporal translation,” in Proc. AAAI Conf. Artif. Intell., 2022, pp. 2659–2667.

[18] E. Nowroozi, Y. Mekdad, M. H. Berenjestanaki, M. Conti, and A. E. Fergougui, “Demystifying the transferability of adversarial attacks in computer networks,” IEEE Trans. Netw. Service Manag., vol. 19, no. 3, pp. 3387–3400, Sep. 2022.

[19] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” 2017, arXiv:1706.06083.

[20] C. Szegedy et al., “Intriguing properties of neural networks,” in Proc. 2nd Int. Conf. Learn. Representations, 2013.

[21] A. Kurakin et al., “Adversarial examples in the physical world,” 2016.

[22] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” 2014, arXiv:1412.6572.

[23] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “DeepFool: A simple and accurate method to fool deep neural networks,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2574–2582.

[24] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in Proc. IEEE Symp. Secur. Privacy, 2017, pp. 39–57.

[25] B. Bayar and M. C. Stamm, “Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection,” IEEE Trans. Inf. Forensics Security, vol. 13, no. 11, pp. 2691–2706, Nov. 2018.

[26] M. Barni, A. Costanzo, E. Nowroozi, and B. Tondi, “CNN-based detection of generic contrast adjustment with JPEG post-processing,” in Proc. Int. Conf. Image Process., 2018, pp. 3803–3807.

[27] X. Qiu, H. Li, W. Luo, and J. Huang, “A universal image forensic strategy based on steganalytic model,” in Proc. 2nd ACM Workshop Info. Hiding Multimedia Secur., 2014, pp. 165–170.

[28] N. Moustafa and J. Slay, “UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),” in Proc. Mil. Commun. Inf. Syst. Conf., 2015, pp. 1–5.

[29] E. Nowroozi, M. Barni, and B. Tondi, “Machine learning techniques for image forensics in adversarial setting,” Ph.D. dissertation, 2020.

[30] J. Rauber, W. Brendel, and M. Bethge, “Foolbox V0.8.0: A Python toolbox to benchmark the robustness of machine learning models,” 2017, CoRR, vol. abs/1707.0.

[31] M. Barni, E. Nowroozi, and B. Tondi, “Higher-order, adversary-aware, double JPEG-detection via selected training on attacked samples,” in Proc. 25th Eur. Signal Process. Conf., 2017, pp. 281–285.

---

### Table XVIII

| Number of Features | 5     | 10    | 50    | 200   | 400   |
|--------------------|-------|-------|-------|-------|-------|
| $N_1$              | 8100  | 18000 | 45000 | 189000| 468000|
| $N_2$              | 3100  | 18000 | 45000 | 189000| 468000|

| Features for all SVM (1724 for $N_1$ and 3200 for $N_2$) | 22400 | 12600 |

---

Authorized use licensed under the terms of the applicable license agreement with IEEE. Restrictions apply.
Ehsan Nowroozi (Senior Member, IEEE) received the PhD degree from Siena University, Italy, in 2020. He is a research fellow with Queen’s University Belfast (QUB), the Centre Centre for Secure Information Technologies (CSIT) since 2023. Prior to joining CSIT, he held research fellow positions, Padua and Siena Universities, in Italy and Sabanci University, in Turkey. His primary research interest lies in AI for Cybersecurity. Ehsan actively engages in professional service and activities as a reviewer for IEEE Transactions on Network and Service Management (TNSM), IEEE Transactions on Information Forensics and Security (TIFS), and IEEE Transactions on Neural Networks and Learning Systems (TNNLS), among others. He attained starting in 2022.

Mohammadreza Mohammadi (Member, IEEE) received the bachelor’s degree in computer engineering (software network) from Bu-Ali Sina University, Hamedan, Iran, in 2019. He is currently working toward the master’s degree of ICT with the University of Padua. His main research interest is in the areas of Machine Learning, Cybersecurity, IoT, and Computer Vision. In addition, he works on research contexts which are a combination of Industrial IoT security and artificial intelligence, and Intrusion detection systems (IDS) and Healthcare systems. He is also a graduate student member of the IEEE institution.

Pargol Golmohammad (Member, IEEE) received the bachelor’s degree in biomedical engineering (bioelectric) from Hamedan University of Technology, Hamedan, Iran in 2018. She is a graduate student member of IEEE institution. Now, She is a second-year Master’s student of ICT (life and health) with the University of Padua. Her main research interest is in the area of Biometrics, IoT, E-Health Security, and Machine Learning.

Yassine Mekdad (Member, IEEE) received the MSc degree in Cryptography and Information Security from Mohammed V University of Rabat, Morocco. He holds a guest researcher position with the SPRITZ research group with the University of Padua, Italy. He is currently working as a research scholar with the Cyber-Physical Systems Security Lab (CSL), Florida International University, Miami, FL, USA. His research interest principally covers security and privacy problems in the Internet of Things (IoT), Industrial Internet-of-Things (IIoT), and Cyber-physical systems (CPS). Furthermore, he works on research problems at the intersection of the cybersecurity and networking fields with an emphasis on their practical and applied aspects.

Mauro Conti (Fellow, IEEE) received the PhD degree from the Sapienza University of Rome, Italy, in 2009. He is a full professor with the University of Padua, Italy. He is also affiliated with TU Delft and the University of Washington, Seattle. His research in the area of Security and Privacy is also funded by companies, including Cisco,Intel, and Huawei. He published more than 450 papers in topmost international peer-reviewed journals and conferences. He is the editor-in-chief for IEEE Transactions on Information Forensics and Security and has been an associate editor for several journals, including IEEE Communications Surveys and Tutorials, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Network and Service Management. He is a Senior Member of ACM.

A. Selcuk Uluagac (Senior Member, IEEE) received the MS and PhD from Georgia Tech and an MS from Carnegie Mellon University. He is an eminent scholar chaired professor and the director of the Cyber-Physical Systems Security Lab in the School of Computing and Information Sciences, Florida International University (FIU), Miami, Florida, USA. Before FIU, he was a senior research engineer with the Georgia Institute of Technology and at Symantec. He is an expert on security and privacy topics with hundreds of scientific/ creative works in practical and applied aspects of these areas. He received the US NSF CAREER Award (2015), the US Air Force Office of Sponsored Research’s Summer Faculty Fellowship (2015), and the University of Padova’s (Italy) Summer Faculty Fellowship (2016). His research in cybersecurity has been funded by numerous government agencies and industry. He has served on the program committees of top-tier security conferences such as ACM CCS, IEEE Security & Privacy (“Oakland”), NDSS, and Usenix Security, Inter alia. He was the general chair of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (ACM WiSec) in 2019. Currently, he serves on the editorial boards of IEEE Transactions on Information Forensics and Security as deputy editor-in-chief, and IEEE Transactions on Mobile Computing and Elsevier Computer Networks Journal as associate editor.