Deep Learning Fusion for Attack Detection in Internet of Things Communications

Ossama Embarak [*1], Mhmed Algrnaodi[2]

[1] Higher Colleges of Technology (HCT), UAE
[2] Electrical Engineering Department, Ecole de technologie superieure, Montreal, Canada
Emails: oembarak@hct.ac.ae ; mhmed.algrnaodi.1@ens.etsmtl.ca

Abstract
The increasing deep learning techniques used in multimedia and network/IoT solve many problems and increase performance. Securing the deep learning models, multimedia, and network/IoT has become a major area of research in the past few years which is considered to be a challenge during generative adversarial attacks over the multimedia or network/IoT. Many efforts and studies try to provide intelligent forensics techniques to solve security issues. This paper introduces a holistic organization of intelligent multimedia forensics that involve deep learning fusion, multimedia, and network/IoT forensics to attack detection. We highlight the importance of using deep learning fusion techniques to obtain intelligent forensics and security over multimedia or Network/IoT. Finally, we discuss the key challenges and future directions in the area of intelligent multimedia forensics using deep learning fusion techniques.

Keywords: Deep Learning Fusion; IoT; Network; Multimedia; Attack Detection.

1. Introduction

Digital forensics has become an interesting domain for investigating and finding evidence of cybercrimes. With the technological advances, including the Internet of Things (IoT), multimedia, and artificial intelligence (AI) applications, the investigation and recovery of cybercrimes must be intelligent, automated, and resilient; thus, we can call them 'Intelligent Multimedia forensics' (IMF). In this essence, IMF would be able to examine and discover evidence of fake multimedia involving videos, images, audio, and texts over IoT systems and networks. One of the most recent fake videos on social media is a video of the Tom Cruise impersonator, which creates accurate videos imitating the actor [1]. The videos, created with the help of artificial intelligent techniques, gained millions of views on social media. Also, another fake clip shows a computer-generated version of the former US leader Obama mapped to fit an audio recording. These kinds of attacks lead to a political crisis and the insult of innocent individuals [2].
IMF can be defined as an area of research that analyses adversarial attacks and utilizes recent multimedia forensics techniques in IoT networks. IMF considers three perspectives; deep learning (DL) forensics [3], multimedia forensics [4], and IoT network forensics [5]. In the first perspective, Despite the advances in DL techniques, they suffer from various adversarial attacks, such as inference and poisoning. IMF can achieve security and safeguard systems against cyber threats. In any secured system, three security principles: Confidentiality, Integrity, and Availability (CIA triad), must be accomplished [5]. Attackers continually attempt to breach these principles. For example, a Denial of Service (DoS) attack tries to corrupt a system's resources, violating its availability. To detect

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022

and trace the origin of sophisticated attacks, IMF techniques would monitor and inspect activities, estimating the violations of security principles [6].

Adversaries would exploit DL models by perturbing benign samples that cannot be observed by humans, leading to data misclassifications. Several research studies have attempted to understand and analyze adversarial attacks by generating robust model layers against adversarial attacks. These model accuracies assuring must be taken into consideration during existing adversarial attacks, uncertainty, misclassification, and noise [7].

Multimedia forensics focuses identification of the image, video, audio, text source and its integrity. Multimedia techniques have become very common in phones, checking and supervision cameras, etc. Also, it becomes easier to edit photos with available programs and share them on social media, which helps spread forged images. Multimedia forensics aims to define the image, video, text, or audio integrity and authenticity. Furthermore, expert decisions expose miscarriages of justice and misleading evidence. In this area, multimedia forensics concentrates on improving and analyzing digital proof in a criminal investigation procedure [8]. The forged multimedia becomes wide-spreading during copy-move operations, which easily edit images [9]. Modifying images can facilitate the process of hiding malicious intent in images, especially with the developments of DL techniques such as Generative Adversarial Networks (GAN) [10]. Multimedia forensics is divided into specific fields of research such as steganography and steganalysis, watermarking, copy-move, deep fake, and camera source identification. With such destructive tampering techniques on the rise, security in images and video must have seniority in the research field.
In this paper, we introduce a comprehensive background of adversarial attacks in IMF over DL, Multimedia, and network/IoT supported by DL techniques and methods in IMF detection and identification over three points of view DL forensics, Multimedia forensics and Network/IoT forensics. In addition, we clarify future directions for research correlated to performing forensics investigation of IMF.
In the DL forensics, we provide a study on most attacks against DL models in training and testing methods providing the crafted attack methods by categorizing them into White-box, Grey-Bx, and Black-Box attacks. Also, providing a defense against these attacks supplied the most proper situation for using each method. In Multimedia forensics, we provide active and passive manipulation methods. The active methods need previous information on elements initially connected to the original multimedia, such as watermarking and steganography methods. The passive methods do not depend on prior knowledge but define if the manipulation process is done on multimedia or not, such as copy-move, intra, and inter-frame manipulation, and deepfake. In the network/IoT forensics, we introduce forensics in every network/IoT layer, such as the physical layer, data-link layer, network layer, transport layer, and application layer. All these perspectives are shown in **Figure 1**.
The remaining paper is structured as follows. Section 2 discusses different DL forensics attacks and it is criteria, and presents the DL forensics detection techniques. Section 3 explains the role of DL models in image and video, audio, and text tampering detection in many forensics aspects. Section 4 provides the most recent attacks on networks /IoT systems. Section 5 presents the latest detection techniques for discovering attacks using DL algorithms. Section 6 demonstrates the most recent challenges and future research directions for research in the domain of intelligent multimedia forensics. Finally, in Section 7, we conclude this work.

Attacks against Intelligent multimedia

In this section, we cover most attacks and tampering in most forensics' parts, so we covered three perspective views of attacks and tampering. The first perspective is the adversarial attack professionally modifies original multimedia so a human cannot observe the modifications. The modified multimedia was named adversarial multimedia, which was misclassified by the classifier with high confidence. These adversarial attacks aim to break the DL models. This can cause many problems in real life, such as applying illegal content which cannot be detected by web content algorithms or web crawlers. The second perspective is that tampering and manipulation techniques in multimedia such as images, video, audio, and Text cause crimes and offend others. Finally, the perspective is the attacks aim to disrupt Network/IoT communication to steal sensitive information. This attack technique is discussed in the following sections and concluded in Figure1.

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022

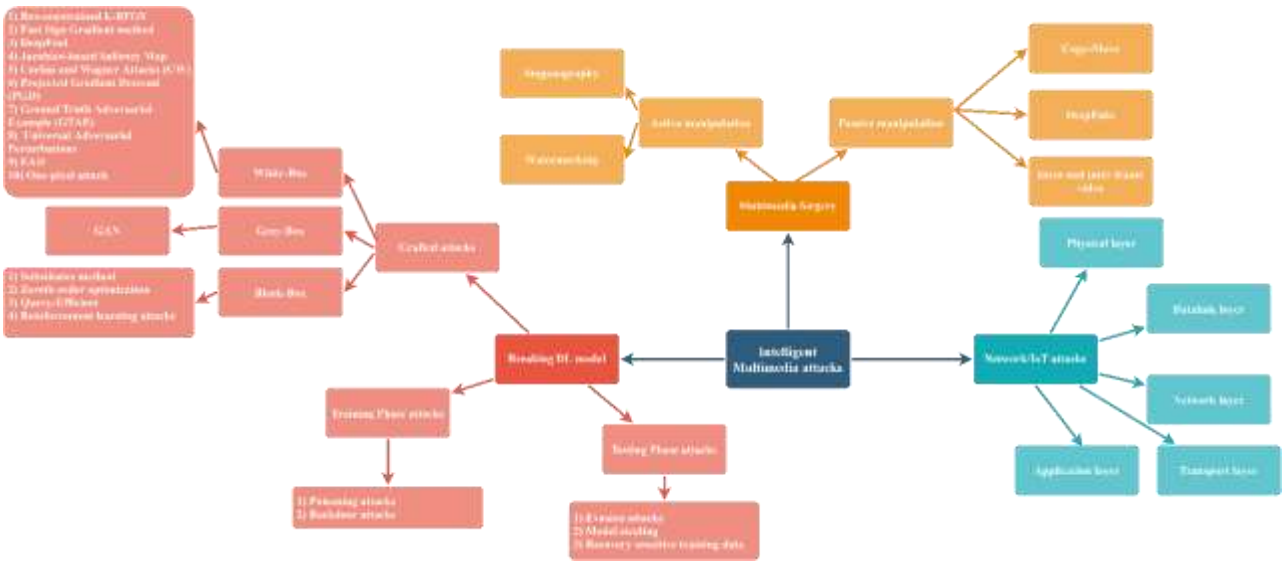

Figure 1: Taxonomy of Different intelligent multimedia attacks

2. Deep learning forensics perspective

2.1. The factors that make Deep Learning architecture easy to attack

A DL model consists of two phases. The training phase is where an algorithm learns the parameters of the model. And testing phase used trained parameters for label prediction.
The General DL model contains the data holder, model provider, client, and attacker if it exists. As shown in **Figure 2**.

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022

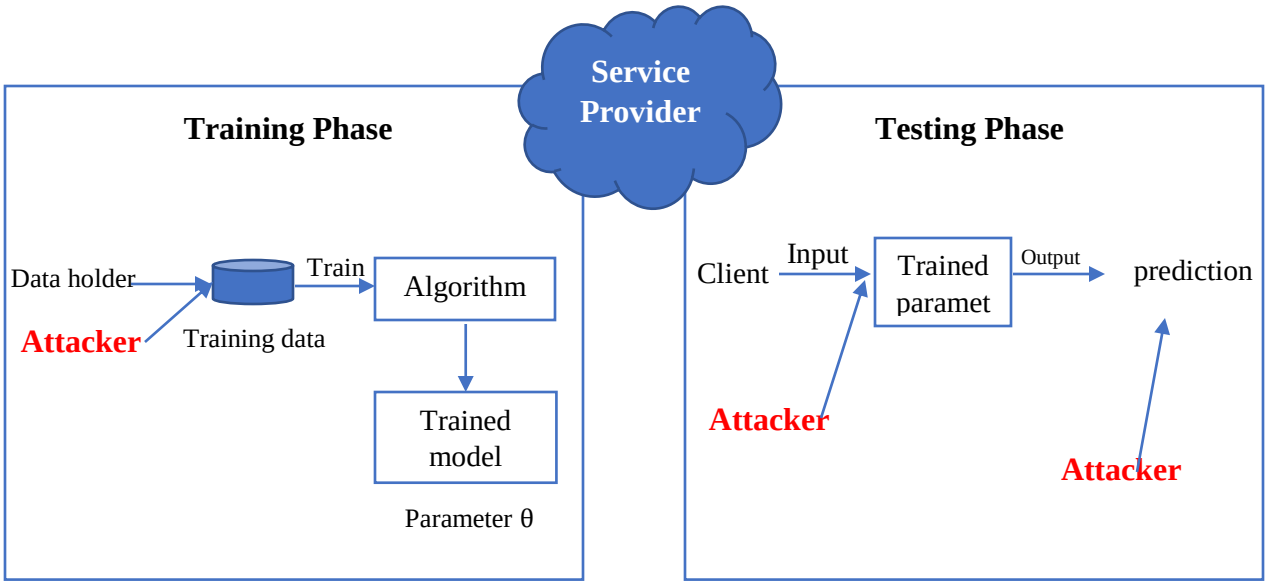

Figure 2: Overview of DL entities during adversarial attacks

The data holders own private training data. The model provider is the algorithm producer and implements training and testing phases. The clients use a prediction model using APIs. The attacker can be a curious person who is interested in secret information. DL models are easy to attack for many reasons. One of the reasons is that in the training phase, the DNN needs a huge amount of data and weights to attain high performance, which is considered computationally expensive and needs weeks using many CPUs and GPUs. These computational capabilities are too much power on hand, So the network is trained on outsourced on the cloud. But these third parties are considered to be an untrusted source of collecting huge data without guaranteeing good data validations **[11]**.

Most reasons for attacking the DL model are still open issues. There are not many studies that agree on the same problems. One of these problems is train high DL model dimensions with linearity raises the probability risk of being vulnerable to adversarial example attacks. Another study shows high risk when using incomplete training data **[11]**. They found that the attacks have less possibility in the test set than in training data. Also, a study**[12]** on the impact of overfitting on allowing the attacker to steal the model information or attributes. The overfitting makes the attacker perform membership inference attacks.

2.2. Attack's criteria in Deep learning forensics

The taxonomy of attacks depending on DL follows the proposed [3]:

• **Timing:** There are two attacks considered as timing attacks, evasion and poisoning attacks (PA). Evasion attacks model decisions and assumes that the model has already been trained, which leads to misclassified predictions. PA training data before training and modifying it which can reduce the DL model performance.

• **Information:** when the attacker can have full information about the victim model, it is named a White-box adversarial attack. In contrast, when the adversary has no information about the model, this attack is named a Black-box. Also, if the attacker has partial information on a model, such as some images, or knows the architecture of the model, which is named a Grey-box.

• **Goals:** The non-targeted attacks aim to make multimedia misleading by the classifier, while in the targeted attacks, the attacker aims to classify multimedia in a specific target class. For example, an adversary attacks a bank's financial model and acts as a trusted client of this bank.

• **Attack Frequency:** The attack can be a one-time or iterative attack. In the one-time, objective function optimization is done in only one execution, whereas the iterative method is done in many executions to produce the perturbation.

2.3. Deep Learning Attacks

2.3.1. Crafted attacks

A. White Box attacks

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022

The attacker has full access to the parameters of the model, such as its parameters, layers, and datasets. Where DL model z with target input (a, b) is information available to an attacker, his aim is to produce fake $\grave{a}$, which leads to wrong predictions by maximizing the loss $\mathcal{L}(z(a+\delta), b)$.

$$\max_{\delta \epsilon \Delta} \mathcal{L}(z(a+\delta), b) \quad (1)$$

In which δ is the noise

Box-Constrained L-BFGS:

In **[13]**, the first DL attacks aim to reduce the distortion in faked example $\grave{a}$, and the noise δ applied to target input a, which leads to misclassifying. Where the objective of this problem:

$$\min_{\delta} \ c\|\delta\|_2 \quad (2)$$

$$\text{s.t. } z(a+\delta) = \grave{b} \ \text{ all pixels in } (a+\delta) \in [0,1] \quad (3)$$

In which b is the true label, and $\grave{b}$ is the target label. For more approximation solving, the author produces the following objective:

$$\min_{\delta} \ c\|\delta\|_2 + \ \mathcal{L}(z(a+\delta), \grave{b}) \quad (4)$$

$$\text{s.t. all pixels in } (a+\delta) \in [0,1] \quad (5)$$

Fast Gradient Sign Method (FGSM)

Goodfellow et al. **[14]** produce a one-step adversarial approach. simple disturb added to generate the adversary image $\grave{a}$:

$$\grave{a} = a + \delta_{ut} \quad \text{Non-target} \quad (6)$$

$$\grave{a} = a - \delta_{tg} \quad \text{Target} \quad (7)$$

Where the perturbation for non-target δ_{ut}:

$$\max_{\|\delta_{tg}\|_p \leq \epsilon} \mathcal{L}(z(a+\delta), b) \quad (8)$$

And target δ_{tg}:

$$\max_{\|\delta_{tg}\|_p \leq \epsilon} (\mathcal{L}(z(a+\delta), b) - \mathcal{L}(z(a+\delta), \grave{b})) \quad (9)$$

This method aims to minimize the distance to the target class where ϵ neighbor ball, which defines by l_p-norm. And maximize the vector norm between the target class and present class. This attack can be easily broken by defense methods, but it is very fast to implement.

DeepFool

Another attack aims to obtain the classification model boundaries. This method considers model linearity and assumes that classes are split by hyperplanes. So minimum perturbation is obtained:

$$\underset{\eta i}{\text{argmin}} \|\eta i\|_2 \ s.t. z(a_i) + \nabla z(a_i)^T \eta i = 0 \quad (10)$$

Where ηi is the perturbation at iteration i.

Another DeepFool extension uses CNN for the multi-classification situation on many datasets such as ImageNet, MNIST. The result shows a better attack with minimum perturbation than FGSM **[15]**.

Jacobian-based Saliency Map Attack (JSMA)

Rather than using a gradient to calculate the perturbation vector, Papernot et al.**[16]** introduce an approach depending on the JSMA matrix performing of score S, which works greedily. The jacobian matrix is expressed as:

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022

$$J_S(a) = \frac{\partial S(a)}{\partial(a)} = \left\{\frac{\partial S_j(a)}{\partial a_i}\right\} a \times j \tag{11}$$

It obtains the impact of variations in the input a to the predicated label, $\grave{b}$.which depends on the idea of saliency maps. This method shows its ability to mislead the classifier in specific target DL models.

Carlini and Wagner Attacks (CW)

CW is considered the most common attack and more successful than FGSM and L-BFGS attacks. CW reformulates the problem of FSGM by trying to minimize the distortion, which is addressed as:

$$\min_{\delta} c\|\delta\|_p + \mathcal{L}\big(z(a+\delta), \grave{b}\big)) \quad s.t.\,(a+\delta)\epsilon[0.1]^n \tag{12}$$

Where $\mathcal{L}\big(z(a+\delta), \grave{b}\big)) = max_{i\neq \grave{b}}(G(\grave{a})_i) - G(\grave{a})_b{}^+$ and $G(a) = g$ are the logits, the algorithm achieves minimum $\mathcal{L}(.)$ to find a larger ` score to be classified as $\grave{b}$.

Projected Gradient Descend (PGD)

Rather than a one-step attack, Kurakin et al.**[17]** produce a single-step version of the FSGM attack that leads to a more powerful attack, and it doesn't reduce the attacker's time and effort to get the best attack. PGD is formulated as follows:

$$\delta \coloneqq P(\delta + a\nabla_\delta)\,\mathcal{L}(f(x+\delta), y) \tag{13}$$

Where P defines as the projection over the ball of interest, for choosing a, further fine-tuning will be performed on PGD.

Ground Truth Adversarial Example (GTAE)

The first algorithm calculates the exact minimum perturbation of the classifier. But this attack depending on the satisfiability of a modulo theories (SMT) solver f, make it not capable of being scalable to big networks. The SMT is performed on a dataset (x, y) and algorithm abstraction θ, and checks if in norm distance there is existence for $\dot{x}$ near x , which leads to classifier misleading.

Universal Adversarial Perturbations (UAP)

UAP is an attack on DL models depending on quasi-imperceptible universal perturbations that can fool training samples in the dataset **[18]**. The perturbation formula is shown as follows:

$$P_{x\sim D}\big(f(x) \neq f(x+\delta)\big) \geq \beta, s.t. \|\delta\|_p \leq \epsilon \tag{14}$$

Where ϵ is the perturbation size depending on $the\ l_p - norm$, and β is the probability that the sample image is tricked by produced perturbation.SO the algorithm aims to achieve the optimized classifier fooled probability. The way perturbation is performing is the same as in the DeepFool attack, in which input is biased toward the decision boundary. Every image perturbation is calculated and greedy accumulated. After that, the accumulator gives a projection over $B_\epsilon$ball of radius ϵ. The results show 4% variations and 80 % fooling accuracy.

The Elastic-Net Attacks to DNNs (EAD)

EAD is a generalization of CW attack l_2 attack and can craft more active attacks depending on $L1$ distortion metrics **[19]**.

For target input $a_0$and correct label t_0, the loss function z in EAD attack is formulated as:

$$\min_a c.z(a,t) + \beta\|a-a_0\|_1 + \|a-a_0\|_2^2 \; subject\ to\ a \in [0,1]^p \tag{15}$$

Where the fake example of a_0 and target class $t \neq t_0$,, $\beta \geq 0$ are regularization parameters of loss function z, and the $L1$ penalty, respectively. The loss function z (a, t) for targeted attacks is addressed as:

$$z(a,t) = max\left\{\max_{j\neq t}[Logit(a)]_j - [Logit(a)]_t, -k\right\} \tag{16}$$

Where $Logit(x) = [[Logit(x)]_1, \ldots, [Logit(x)]_k] \in \mathbb{R}^k$ is the logit layer produced by an in the DNN, K is the number of classes, and $\kappa \geq$ is a confidence parameter.

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022

The results on many datasets such as ImageNet, CIFAR-10, and MNIST and capable of avoiding the training phase using defensive distillation like CW attacks.

One-pixel attack
Same problem as in L-BFGS with restrictions in l_0 perturbation norm, which reduces the number of allowed pixels to change. The results on CIFAR10 for a well-trained VDD16 with accuracy 85.5% accuracy on test data achieved (63.5%) testing samples can be attacked by only changing one pixel in a non-targeted position. Which shows the weak robustness of DL models [20].

B. Grey-Box attacks

Generative adversarial network (GAN)
In GAN attacks, Xiao et al. **[21]** introduced the generation of adversarial samples using GAN. Specifically, a generator maximizes target adversarial loss and GAN loss to learn adversarial distribution. Another study uses GAN to break the collaborative framework. The attacker trains these networks to produce samples similar to the victim data, so both samples will have the same distribution **[22]**.

C. Black-Box attack

Substitute model
The first Black-Box attacks the DL models with no access to classifier parameters or dataset. The attacker can predict the output label y using only input x. The attacker may know little information about the model domain such as (faces, animals, etc.) and the model architecture such as (CNN, RNN, etc.).
The adversarial example can attack $F1$ if it can attack $F2$, which has a similar structure. So, this study [23] produces an approach substitution model training $\grave{F}$ to predict the victim classifier F, and then attack the substituting model, $\grave{F}$.which follows the following procedures:
1) an input substitute dataset,
2) training the substituting model,
3) augmenting the dataset, and
4) attacking a substituting model.

Zeroth Order Optimization Based Attack (ZOO)

Rather than only achieve label information from the classifier, Chen et al.**[24]** introduce Zoo attacks when the adversary knows data about the confidence scores of the model and the input data. So, there is no need to maintain a substituting training data or model. ZOO attacks estimate the gradient of the target model rather than use 0 coordinates in traditional gradient descent.
Given target sample , and prediction confidence $P(a)$ where a is tuned. The following equation shows the gradient information scrape the output of $P(.)by$:

$$\frac{\partial P(a)}{\partial a_i} \approx \frac{P(a + he_i) - P(a - he_i)}{2h} \tag{17}$$

The results show that ZOO is more effective than substituting methods because it can exploit the prediction confidence information rather than only predicted labels.
Query-efficient black-box attack
The main restriction of usage of black-box attacks is that the classifier takes many input queries. To enhance these attacks, the number of queries must be limited. Such as contribution in **[25]**, which uses optimization algorithms to assess the gradient information from model outputs. Another study **[26]** tries to reduce the number of queries by depending on estimating the expectation gradient around queries. Another study **[27]** aims to find the neighbors of the target image using a genetic algorithm.

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022

Attack on Reinforcement learning (RL) algorithm

RL consists of two basic elements state and actions. In **[28]**, propose an attack approach in reinforcement learning (RL). In RL, attacks can modify images used to obtain the state element and perform an adversarial policy. This is not a robust attack as it considers dual players in one game, but a player attitude fails the other policy. A Black-box attack target to change the victim, which is trained using Proximal Policy Optimization to learn fair playing. And attack is trained to fail this policy.

2.3.2. Training phase attack

A. Poisoning attack (PA)

The adversary aims to mislead the classifier by editing, removing, or injecting samples into the training data in a DL model. Just small poisoning samples can affect the model availability and integrity, as shown in **Figure 3.**

Error-generic PA – preform many misclassifications on data points regardless of the prediction result, which would

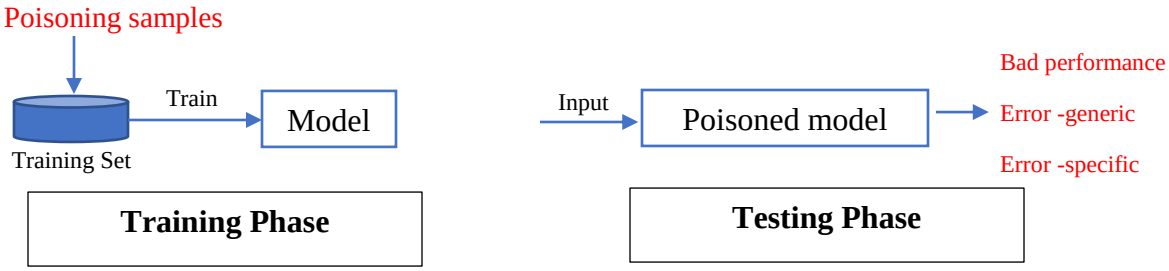

Figure 3: Poisoning attacks

lead to a DoS attack in the system.

Error-specific PA – perform specific misclassification on specific classes.

Some previous studies as an example of PA are presented in **Table 4.**

Table 1: Survey on PA studies.

Ref.	DL architecture	Application	Attacker information	Goal
Yang et al. [29]	Feed-forward Neural Network, CNN	Image Classification	White-box	Integrity attack
Shafahi et al. [30]	CNN	Image Classification	Gray-box	Integrity attack
Suciu et al. [31]	CNN, Linear SVM, Random Forest	Classification task	Gray-box	Integrity attack
Lovisotto et al. [32]	FaceNet, VGG16, ResNet-50	Face recognition	White-box	Integrity attack
Jiang et al. [33]	CNN	Image Recognition	Black-box	Integrity attack

Yang et al. **[29]** use GAN to execute PA, which consists of a generator model and discriminator model. Auto-Encoder is working as a generator and the target model as a discriminator. The discriminator calculates the impact of bad data on the model. This speed up the bad data-producing process but a leak to interact with the target model quickly. The work **[30]** performed an attack using multi techniques for data poisoning. In contrast, decreasing the performance rather than installing a backdoor. Another study by Suciu et al. **[31]** described an attacker's background knowledge and the ability for data PA regarding multiple dimensions such as Feature, Algorithm, and Instance. Lovisotto et al. **[32]** performed poisoning face images to decrease the distance between the attacker face template and target template, During the face template update. Over time, the target face will be poisoned, and the attacker

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022

can access the system. Jiang et al. **[33]** use an inclusive set of road signs which similar to attacker data. This set is modified to be like a real set that misleads CNN to correctly identify an inclusive set.

B. Backdoor attack

Backdoor attacks

The adversary installs a backdoor with a certain mark so that the target will be misclassified. As shown in **Figure 4.** And also, a review of backdoor attack studies is presented in **Table 5.**

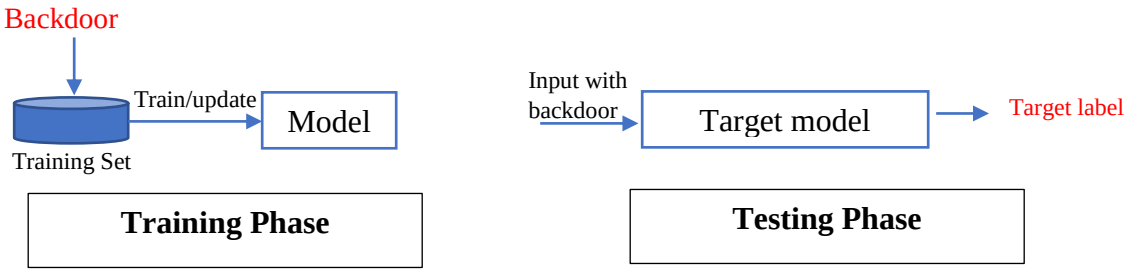

Figure 4: Backdoor attacks

Table 2: Survey on Backdoor attack.

Application	Attacker information	Accessibility to model parameter
Image	White-Box	Valid
Text	White-Box	Valid
Audio	White-Box	Valid
Video	White-Box	Valid
Ward embedding	Grey-Box	Partially valid
NLP	Grey-Box	Partially valid
Malware classification	Grey-Box	Partially valid
Deep generative model	White-Box	Valid

Some studies apply outsourcing in their work when preform DL training and testing on outsources such as MLaaS, because of lacking DL computation skills. So, it is easy for attackers as the third party to manipulate data and backdoor the DL model. Many studies have covered the outsourcing attacks. Some attacks exploit transfer learning (TL) techniques to limit data and computational resources to train the model. These attacks such as in studies.
Another type of attack is data collection. When users collect data from multiple untrusted sources, some work on these attacks**.**

Trojan attacks

Some backdoor attacks, also called trojan, when installing backdoor to published model and then redistribute it. This avoids training from scratch and doesn't need to access original data because attackers gather training data using reverse engineering.

2.3.3. Testing Phase Attacks

A. Evasion attacks

Adversarial example aims to disrupt the input data, which leads to misclassifying the DL model. this term was produced in 2013 by Szegedy et al. **[13]**, focusing on DL algorithms. In ancient studies, Evasion attacks usually target the non-neural network model such as, which is used to spam filtering, malware detection, etc. But now, many studies use evasion attacks on DL algorithms. These attacks target testing data. A survey of previous studies shows in **Table 3**.

Table 3: Survey on evasion attacks

DL architecture	Application	Attacker information

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022

CNN	Image Classification	Black-box
CNN	Image Classification	Black-box
CNN	Image Classification	White-box, Black-box
CNN	Image Classification	White-box
CNN	Image Classification	White-box
RNN	Speech recognition	White-box
CNN	Stop Sign recognition	White-box
Inceptin-v3	Image Captioning	White-box
AllConv, NiN, VGG16 network	Image classification	Black-box

B. Model Stealing

Recent papers demonstrate that attackers can extract the DL model by monitoring the outputs and confidence scores regarding selected inputs. Also named as model extraction attack. The summary of model stealing attacks is presented in **Table 4**.

Table 4: survey on model stealing attacks.

Impact	Advantage	Disadvantages
Same functionality of target model	Does not need further information from the target model	Repeated query to the target model
Hyperparameter extraction	Can perform multiple DL and ML tasks	More information on the target algorithm and training data
Recreate the target model	More effective than query methods	Computationally expensive on only two layers NN
Depending on the architecture hint	Obtain complete model architecture with black-box access	Require computational skills
Deep Reinforcement learning	More advances compared to ancient methods	Need to be applied to more applications

Tramèr et al. introduced a model stealing attack by using multiple user inquiries. The performance of the attack showed on logistic regression (LR), decision trees (DT), and neural network (NN). Inserting regular queries through model APIs returns predicated labels. This is used to evaluate Amazon and BigML. Another example on the black-box attack, where the attacker used DL to predict the model labels and build an equivalent model. This attack relay on the query the model regard to it is input and predicate its labels. The extracted model is trained by The labeled data with the same functions.

Chandrasekaran et al. introduced model stealing from active learning (AL). The composition of the AL query is used to steal the model without knowing any further information. Wang and Gong extracted the hyperparameters of DL models by using a learner. Solving a linear equation and setting the gradient to 0 to extract the hyperparameters. The results show on LR, NN, and SVM. Learning the victim model to query the gradient information of certain inputs can rapidly return the parameters of the model. The drawback of this model is high computation during just the process of two NN layers**.**

Hu et al. introduce Deepsniffer, a framework for stealing the model without any prior knowledge of the target model. the basic concept of this framework is to learn the relative between extracted architectural hints (e.g., amounts of memory reads/writes achieved by side-channel or bus snooping attacks) and model internal architectures. A recent work by Chen et al. first extraction for the Deep Reinforcement model. This method advances the model stealing methods. The attacker predicts the training algorithms with only black-box access.

2.4. Defense against Deep learning attacks

2.4.1. Defense against training attacks

A. Data Sanitization (DS)

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022

DS is a defense against PA, which filters bad samples from data before training. Cretu et al. proposed a new technique for PA a novel data by using sanitization for classifier tampering detection. Another study proposed a DS approach for poisoning detection named Reject on Negative Impact (RONI). This method was accurate in dictionary attack messages on the Spam Bayes spam filter. Koh et al. proposed anti-sanitization techniques using three attacks which provides more suggestions for the researcher that more work is needed in this area.

B. Robust Statistics

In contrast to DS, the Robust statistics considered robustness against PA. To enhance the robustness of recognition models, a proposed method of using multiple classifier systems was introduced by Biggio et al. This approach depends on avert of more than one classifier from misleading the system. Another approach for the same author shows that regardless of the classification algorithm, the bootstrap aggregation is successful against PA, which is named 'bagging .'Bagging was introduced by Breiman as a classifier enhancement approach in terms of accuracy. Depending on producing multiple classifiers and using them for aggregation, the result shows effectively during bug variation in predictions and small variation in the training set.

2.4.2. Defenses against Testing Phase Attacks

A. Robustness Improvements

Robustness is the DL model capability of detection and rejection of adversarial examples. Although these approaches help to enhance detection against testing attacks, it implemented through the training phase.

Adversarial Training. Augmented perturbed examples in the training phase can enhance the robustness of DL models. Input optimization finds perturbed examples and maximizes the prediction error. This method shows effectiveness during white-box attacks, although its time consuming because of its iterative computation. The maximum loss of the model can be shown if calculating the perturbations during training. But DL model is still suspectable to black-box attacks during using fast single steps in adversarial training.

Defensive Distillation. This method is used to move learning from large classified models to smaller susceptible models when performing the testing accuracy. This method is used for inference accuracy smoothing by simulating narrow trained data to DNN. Another study introduces a defensive distillation approach to train only models robust to input perturbations The results show that this approach makes the DL model more resistant to adversarial examples.

Gradient Masking. Used to decrease the perturbated example to decrease the sensitivity of a DL. This approach is model first-order derivatives which are calculated regarding their input. This derivative is minimized through the learning phase. This technique shows success against atter, which aims to manipulate gradient-based knowledge. Also, it is reasonable to decrease sensitivity to reduce variations in the input, which can suspect the adversarial attacks. Papernot et al. **[23]** show the drawbacks of this approach by showing its inefficiency in the black-box attack.

Feature Squeezing. This approach decreases the spaces of accessible input features to an attacker, which is done by integrating many samples in vector space into one sample. A study proposed two feature squeezing methods: (1) reduction of color image depth and (2) Smoothing to decrease changes between pixels. The results show effectiveness against input perturbations**.** They expand this work**,** where they demonstrate that median smoothening is a more efficient squeezer in mitigating the CW attack. One drawback of this approach is that it can reduce the accuracy of the classifier on benign inputs.

Ensemble Method. Enhancing classification decisions of supervised learning (SL). These techniques are recently used against adversarial attacks, although it has been introduced in research papers over the years. Abbasi and Gagné introduce a technique that enhances the robustness of CNN. This method uses multi-classifiers to identify and deny adversarial attacks and receive clean samples depending on confidence. This approach decreases adversarial attacks confidence prediction while protecting the confidence of clean samples to a specific level. The ensembles created using this method can be avoided by an adaptive attack that produces adversarial examples with low distortions. Another proposed ensemble method improves the prediction of benign samples and enchantment robustness against adversarial examples but has high computational complexity. Another study introduces the

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022

Random Self-Ensemble (RSE), which integrates the principles of randomness and ensemble to enhance the robustness of DL models.

Reformers/Autoencoders (AE). Transform the input to benign input via reconstruction. The DL model prediction will not be affected, but adversarial examples could be changed during reformed examples are near to the good examples. AE first learns a set of concealed representations for input data and then reconstructs the output using the concealed representation. MagNet has been proposed, which is a robust approach against DL attacks. This approach combines a reformer and a detector. AE employs the detector, which computes reconstruction error, and refuses high reconstruction error examples. The results show this approach reached 99% classification accuracy of introduced attacks.

B. Differential Privacy

Differential privacy aims to enhance the robustness of DL against adversarial attacks. Differential privacy includes randomness of the training set or the output to minimize the release of sensitive training data. A study proposed PixelDPdepenfing on differential privacy which reconstructs training data from model parameters to protect the DL training set from inversion attacks. PixelDP was introduced as a scalable defense which implemented on many DL models. The results show that this method is more precise in predictions using the euclidean norm attack contrasted to other methods. One disadvantage of this approach is high computational for training and testing.

C. Homomorphic Encryption (HE)

Current improvements in this method, such as the fully HE, make operations like addition and multiplication capable of dealing with encoded data with no need to decode the data. This method was introduced to attain privacy preservation during outsourcing to protect sensitive data when using predictive models, such as MLaaS platforms. Also, crypto-nets have been introduced to perform predictions on encoded data and outcomes in the encoded form. Crypto-net can be achievable for the inference stage and some limited learning in specific fields. Implementing. Hesamifard et al. proposed a CryptoDL which applies DNN algorithms over encrypted data. The CNN was trained using low-degree polynomials and HE. Low degree polynomials are necessary to achieve effective HE schemes and overcome the other HE studies practical limitations, like crypto-nets. It is observed that CryptoDL is scalable, more effective, and accurate in privacy-preserving predictions.

2.4.3. Detection-only defenses

During the restriction of producing full defense approaches, the detection-only defense method has recently become noticeable to security experts. This defense aims to produce a difference between good and bad examples and reject the fooled examples for confidence classification.

A. Kernel Density Estimation (KDE) based detector

This method was proposed by Feinman et al. for anomaly detection in DL models based on KDE, which computes feature space for the end hidden layer of a model to find other points than the data manifold. KDE is given as follows, where the DE for a point a with predicted class t is given as:

$$\widehat{K}(a, A_t) = \sum_{a_i \epsilon A_t} K_\sigma(\emptyset(a), \emptyset(a_i))$$

Where ϕ (a) is activation for the end hidden layer for point a, A_t is the data training of label t, and σ is the tuned bandwidth. This approach shows it is efficiency against the FGSM, Basic Iterative Method, **JSMA**, and CW attacks demonstrated on the many datasets. But a study shows that KDE can break the CW to produce adversarial attacks on MNIST with improved distortion.

B. Bayesian Neural Network Uncertainty-based detector

Uncertainty approaches for adversarial examples detection were introduced by Feinman et al. by identifying low confidence region points that contain pixels of input space. This approach finds further data about confidence values that are not usually accessible using approaches depending on distance evaluations such as KDE. This ambiguity approach depends on dropout for applying randomness to the DL model, a method proposed to decrease overfitting during DL training. The ambiguity estimates of the DL network on a given instance e^* and stochastic predictions $\{\widehat{y_1^*}, \dots, \widehat{t_T^*}\}$ can be computed as:

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022

$$U(x^*) = \frac{1}{T}\sum_{i=1}^{T} \widehat{y_i^*}^T \widehat{y_i^*} - \left(\frac{1}{T}\sum_{i=1}^{T} \widehat{y_i^*}\right)^T \left(\frac{1}{T}\sum_{i=1}^{T} \widehat{y_i^*}\right)$$

Using LeNet CNN trained with a dropout rate of 0.5, it is observed that the Bayesian uncertainty approach is efficient in identifying adversarial examples crafted using a multiple attack method.

C. Maximum Mean Discrepancy based detector (MMD)

MMD aims to identify attacks from a presented input. This hypothesis test tries to verify if samples X_1 and X_2 are derived from the same distribution depending on the sample's statistical hypothesis testing. If sample X_1 is derived from distribution p and sample X_2 is derived from distribution q, the null hypothesis H_0states that $p = q$ while the alternative hypothesis H_A indicates that $p \neq q$. The statistical test inputs the two samples and differentiates between H_0 and H_A. The MMD test is a kernel-based test during high dimensionality data.

Another identifying algorithm uses the asymptotic distribution of unbiased MMD. This approach contains using a subsampling technique to drive samples from the available data with replacement to consistently estimate the distribution of the MMD under the null hypothesis. It is observed that MMD can statistically differentiate adversarial examples from benign examples. One drawback is that MMD cannot identify attacks when the CW attack algorithm is used.

D. Local Intrinsic Dimensionality (LID) based detector

LID has been introduced for adversarial examples detection from a given input, which characterizes the intrinsic dimensionality of adversarial areas in the DL model depending on the smallest nearest neighbor distance. It is noticed that LID estimates can identify attacks using the Maximum Likelihood Estimator (MLE) of LID to estimate the correct distance distribution. In which a reference data sample $x \in p$, where p represents data distribution, the MLE of the LID at x is defined as:

$$\widehat{LID}(x) = -\left(\frac{1}{k}\sum_{i=1}^{k} \log\frac{r_i(x)}{r_k(x)}\right)^{-1}$$

The LID characteristics enable the distinction between multiple attack algorithms.

3. IoT Network perspective

3.1. Network/IoT attacks

A. physical layer attacks

In the physical layer, manipulation in different IoT devices can be done. One of the most commonly used methods to access devices is social engineering which can make real attacks, leading to device destruction, spying, and side-route further attacks. Even though it utilizes various technologies at the IoT device layer, social engineering methods are needed for most attacks. The attackers may have various objectives such as physically damaging the devices, reducing their life cycle, threatening the communication process, manipulating the energy supplier, etc. The devise attacks can be the beginning of other attacks. For example, in a smart home, deactivating an alarm in a house could expose the home to stolen or other damages. Also, important data leakage may happen if malicious sensor replacement is done with any sensor in the IoT environment.

Another dangerous attack named man-in-the-middle can be happened by injecting a malicious node into the network, which makes the attacker increase privileges and releases other attacks. Devise manipulation may also lead to altering the routing information that will influence the communication in the other network layers. Also, jamming radio frequencies that prevent communication can lead to DoS attacks in IoT, which influences the operations of IoT applications. Social engineering is utilized to gain access to the devices. Social engineering attacks can be limited by awareness and strict access control. Sleep deprivation attacks reduce the energy resource of devices. For example, sensors are put on sleep mode to limit data sending for energy saving. Attackers could tamper with the setting of these nodes to keep them working the time to reduce the energy life.

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022

B. Data link Layer attack.

IoT integrates many communication tools into the lower layers of the TCP/IP protocol stack, which makes it a complex heterogeneous network, such as ZigBee, WSN, MANET, Wi-Fi, RFID, NFC, etc. Every one of these tools has its drawbacks and security risks. Yang et al. investigate the security risks in IoT layers and the solution to these issues. They introduce the heterogeneity at the physical layer of the IoT, and then various changes are done on the data link layer, for example, during specific channel design based on the underlying physical layer tools. So heterogeneity must be included in both physical and data link layers for more security.

C. Network layer attack

The purpose of this layer is to find the best route for transmitting packets from source to destination. So, the attacker's main goal is to disrupt the communication channel between source and destination, which is chosen from routing protocols such as traffic analysis, spoofing, and Sybil attacks that make network illusions using fake identities. These attacks affect the application functionality and cause data leakage. Also, any modification in forwarding nodes may negatively affect the quality of service in IoT applications. The attacker in the network layer can transmit any data rather than legitimate data using cloning and spoofing in this layer. The unauthorized access to RFID can allow the attacker to read and write data which leads to sensitive data leakage. Securing networks and detecting attacks in the early stage is becoming important, so attackers make an intrusion in different ways and then launch many different attacks. Another way of attack, the attacker can exploit a threat node and perform a sinkhole by fake forwarding. In these attacks, the attacker usually exploits networks connected with sensors or mobile networks.

These attacks increase the probability of releasing more DDoS attacks and interrupting the full network. DDoS attacks disrupt normal traffic by flooding the network with more traffic that cannot be handled by the network. Another attack relies on launching fake nodes where the network deals with these nodes as actual nodes sending data. Sybil attacks can cause false data to spread in the system and pass these data to applications such as voting and election systems. To summarize, the attacker exploits the resource constraints and scarcity of good authentication and authorization schemes to launch attacks in the network layer.

D. Transport layer attacks

TCP and UDP are the main protocols in the transport layer, which are the attacker's goal attacks. TCP's main properties are dependable transmission, connection-oriented, and verified services such as e-mail services. UDP properties are connectionless, limit overhead and latency, and does not ensure reliability such as video streaming, online gaming, VoIP, and IPTV. These protocols contain multiple security issues exploited by the attacker to disrupt communication.

Ping flooding attacking or TCP flooding attacking is a DOS attack that sends many numbers of ping requests which will be replied to by sending ping replying and continue the process until the victim blocks requests and responses. If The request is denied, the attacker will send UDP packets to drain the network bandwidth, which will affect system performance. Another attack aims to expect the sequence number that is utilized to detect the packets in a TCP connection which is named the TCP sequence prediction attack. This attack has falsified the packages and damaged the network.

E. Application layer attacks

The application layer is considered a target to attackers because attacks in this layer are quite easy to launch. one of the most used attacks is the buffer overflows attack which is used in different applications. Many techniques limit buffer overflow by static and dynamic code analysis and symbolic debugging, but these mechanisms are not effective in IoT during resource constraints. Buffer overflow can lead to malicious code injection to IoT applications also many attacks on IoT applications such as SQL injection, object referencing, and cross-site scripting. In 2017 different attacks are identified as the highest risk vulnerabilities by the Open Web Application Security Project (OWASP). These vulnerabilities cause many other attacks which can allow the attacker to steal sensitive data.

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022

Botnet considering one of the most dangerous attacks that disrupt networks and systems. So, mitigation of these attacks becomes important and a challenge for IoT because of its intelligently exploiting of vulnerabilities which leads to launching many other attacks such as DDoS. Attackers are exploiting the resource constraints and not workable sophisticated protocols in IoT to release cryptographic attacks. Also, the use of old cryptographic protocols allows the attackers to crack the encoding schemes. To summarize, attacks in the application layer is forcing challenge to mitigate during their high computational expensive [6].

3.2. Network/IoT attack detection using deep learning techniques

We provide a description of most DL techniques for IM attacks over network/IoT. We focus on DL approaches depending on DBNs, AEs, CNNs, and GANs.

A. Deep Belief Networks

DBN consists of several layers with values where there is a connection between the layers. DBN classifies the data into various categories. Also consists of an unsupervised network like a Restricted Boltzmann Machine (RBM). The RBMs implement two layers in DBN, one visible and one hidden, represented by a variable number of neurons. Moreover, in every RBM, connection in the same layers is restricted, but the different layers are very connected. One most advantages of DBN is fast learning procedures, which can train using greedy learning algorithms. The greedy learning algorithm trains one RBM at a time until all the RBMs have been trained in unsupervised learning (USL).

For that reason, DBN is essential for intrusion detection. Also, fast learning and USL allow DBM to be capable of dimensionality reduction step, to extract a compact and discriminant representation of the data, devoid of using labels, especially in big intrusion detection databases. The first DBN applied for the reduction step was in 2011 by Salama et al., where SVM performs the intrusions classifications in the NSL-KDD dataset. Another DBN intrusion detection study when optimizing parameters using PSO to decrease the dimensions of input data. Then, classification is done using a probabilistic neural network.

Recently many IDs methods based on DBN, which aims to enhance the IDs accuracy on NSLKDD and KDD99 datasets. One disadvantage of DBN is that it cannot be trained end-to-end in an SL using the gradient descent method. Because of this, most training situations depend on a contrastive divergence algorithm, which is based on gradient approximation. But on the other side, during advances in GPU and CPU power computation techniques, it becomes easier to train DBN models depending on gradient descent, with no gradient approximation. Chen, Zhang, and Maharjan implement a DBN model for mobile edge computing protected by detecting malicious attacks. Detection of this automatic approach achieves better accuracy than traditional ML algorithms.

B. Autoencoder

AEs is a type of feedforward NN where the input is the same as the output, which aims to obtain a compact and discriminant representation of the input data using USL. To perform detection, representation can be input to the classifier. An AE involves representing the input data in hidden space in a nonlinear way and then mapping the encoded data to the original space. The main aim of a decoder is to reduce the reconstruction error, which is known as the difference between the original input data and the decoded data.

AE has a high capability to reduce dimensionality and extract compact input data. So, it is usually performed in preprocessing phase in IDs. Not like DBN, AEs able to end-to-end training based on gradient descent, with no use of gradient approximation technique. A proposed method aims to compact and discriminate the representation of input data depending on seven AE. This effective method obtains high accuracy using PCA and kernel PCA using the NSL-KDD dataset. But after dimensionality reduction, this approach suffers from information scarcity about the shallow classifiers for classification. Yousefi-Azar et al. achieve high accuracy using the NSL-KDD dataset. This approach inputs data By AE for feature extraction and, after that, uses narrow classifiers for the classification step. This proposed to achieve higher accuracy when using a naïve Bayes classifier than the accuracy of the previous study.

Many studies aim to achieve high accuracy for IDs, So many techniques depending integration between AE and the density estimation model. Such as proposed by Cao et al. use this kind of integration on the NSL-KDD dataset to obtain higher accuracy in identifying DoS and probe attacks. Another integration work between AE and Gaussian mixture model on KDD99 dataset. This method consists of an estimation network and a compression network. The estimation network estimates the sample densities in a low-dimensional space. The compression network pushes the

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022

data into a lower-dimensional space. Update model parameter is performed joint parameter optimization. This approach obtains higher accuracy than previous studies.

One important type of AE is sparse AEs with its regularization capability, which is used successfully in dimension reduction. An example, a proposed method integrates the sparse AE with a softmax regression classifier. This binary classification achieves better accuracy using NSL-KDD. Vincen et al. introduced a method depending on stacked AEs (SAEs), which consist of the number of AEs trained individually and then "stacked" to achieve a more deep and discriminant representation. Another use for SAE on CTU-13 dataset for raw traffic data preprocessing.

More enhancement efforts in IDs accuracy are proposed by integrating SAE with a random forest classifier. This approach was performed on the KDD99 and NSL-KDD datasets and evaluated five-class classification. But this approach obtains low accuracy against U2R and R2L attacks. An integration between four AEs proposed in to obtain high IDs accuracy using the KDD99 dataset. Also, another method introduced a radial basis function classifier and depended on the integration of SAEs using the AWID2018 dataset. One other type of AEs is variational AEs (VAEs). Not like AEs, VAEs depend on a probabilistic generative model for input data reconstruction. Therefore, VAEs are less sensitive to overfitting problems than AEs. For IDs, integration techniques between VAEs and gradient-based fingerprinting detection model]. This approach extracted NetFlow data from the UGR16 dataset and then performed VAE for feature reduction. Other methods integrate VAE with many classifiers on the NSL-KDD and UNSW-NB15 datasets. This approach obtains better accuracy during the decision tree and random forest classifiers. Another proposed, an integrated method between AE and DBN was implemented for malware detection and also dimension reduction using nonlinear mapping to obtain only the major features.

C. Convolution neural networks

CNNs can handle multidimensional input data such as 3D images. The preprocessing layer in CNN performs convolving operations in the ConvNet layer, with the utilization of gradient descent for parameter learning. CNNs is automatically learning data representations without the need for an ML algorithms step. So, CNN is efficient in many cases, such as IDs. In IDs, CNN is used in many classifying attack types focusing on commonly shared features when minimizing preprocessing of the data.CNN is most used in image processing applications. So, CNN can transform features into a 2D format for IDs.

For example, li et al. introduce a preprocessing technique by transforming symbolic attributes such as flag, service, and protocol type attribute into binary vectors using one-hot encoding. After that, continuous attributes are transformed by applying min-max normalization, which is split into ten intervals and performing one-hot encoding. After All, the obtained vectors are integrated and reshaped to form a 2-D image. Another method aims to transform binary in malware into the gray level image. First, read a file an 8-b vector from binaries, then transform each binary number to its equivalent decimal value. Finally, producing a 2-D gray-level image. Some methods in IDs transform gray-level images into RGB images. Kim et al. introduce a new approach that gives equal weight to each feature and represents each feature with 24-b for each pixel. to speed up to conversion process. Some approach select features before transforming the data into an image format. Such in this work, perform feature selection using a genetic algorithm.

CNN is used for IDs in many studies which have been introduced in the literature. Many other studies are based on LeNet, ResNet, GoogLeNet, or VGG-16. The LeNet was used most on AWID208 dataset in many to obtain high accuracy. Also, the ResNet and GoogLeNet architectures were implemented on the NSL-KDD dataset. The VGG-16 architecture was introduced in , which was performed on Malign and the Microsoft Malware Dataset. Some other works using the integration between CNN model and basic ML model, such as Nguyen et al, introduce integration be CNN and naïve Bayes or K-NN classifier on KDD99 to detect DoS attacks. The results show better accuracy for this approach. Another hybrid architecture between AE and CNN which testing on private data. In botnet attacks detection, Koroniotis et al. [5] introduce a new forensics technique Particle Deep Framework .this framework consists of three stages: extraction stage for data and guarantee its integrity, stage two using PSO to choose an optimal parameter for CNN. The final stage is to detect unnormal events in smart IoT homes.

D. Generative adversarial networks (GANs)

A GAN framework simultaneously trains two models, the generative and discriminative models, via an adversarial process. The generative model generates reasonable data samples, and the discriminative model predicts fake generator data and real data. Some GAN implementations are very important in Network forensics, such as securing cyberspace of IoT systems by integration between DNN and GAN. GAN training is considered to be a difficult task because of its instability and unable to deal with discrete data such as text.

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022

4. Results and Discussion of Deep Learning Fusion for Attack Detection

This experimental analysis makes use of the IoT-23 data. It includes data gathered from twenty compromised Raspberry Pi and three benign Devices. Stratosphere Laboratory at the Czech Technical Institute in the Czech Republic provided the data on the Internet of Things network traffic. Innocuous situations were produced by monitoring the network activity of three different Internet of Things (IoT) gadgets. These three Internet of Things gadgets are the actual deal, not a software simulation. Beyond that, there are about 325 million network flows records from 2020 included in the collection.

First, a human analyst looks through the .pcap file that was taken from the network node. After that, suspect flows are isolated, and labels are applied accordingly. The next thing the expert does is create a labeled comma-separated value (.csv) file. The.csv files with labels are manually processed using a python script. Another python script parsed log data, matched them to the trend of previously labeled.csv files, and then labeled the analyzed files according to their natural progression.

The first step in analyzing the information is to transform the labeled files received from the Tropospheric Lab into .CSV format. The characteristics, such as an IP address, were broken down into their component parts and presented as integer features. Following this, the numeric order of the categorical characteristics is encoded, and the numeric data is normalized to lie on a scale from 0 to 1. We finish our supervised machine learning pipeline by encoding the labeled targets we've been working with into the One-Hot representation. As a result, the intended encoding method will need to contain nine Sub-Classes to accommodate the eight potentially harmful attack Tags.

For the purpose of testing the effectiveness of the suggested security mechanism, we use the indicators below. When a sample has a positive label, it is considered a True Positive (TP), and when it has a negative label, it is considered a True Negative (TN). When samples with a negative label are incorrectly assigned to the Positive category, this is known as a false positive. Samples with a positive label that were incorrectly assigned to the Negative examples by the model are called False Negatives (FN). The Confusion Matrix tabular representations are utilized in subsequent performance measure computation.

Network data are gathered from a wide range of compromised IoT devices and used to train and evaluate the suggested DL-based security framework. Each subnet has a convolutional neural network (CNN) model attached to it, allowing it to access the most accurate representation of network characteristics. We need to install a new CNN unit to the switch port every time we add more Internet of Things devices to the system. The accompanying CNN component will manage the increased IoT traffic through the network. In addition, the CNN-extracted features are sent into the LSTM classifier, where attack detection may be processed with little computational cost.

With just the CNN model and the iot-23 dataset, Table 3 displays the accuracy of classification. Classifying assaults are difficult for the first model. However, it performs well when tasked with recognizing benign characteristics, doing this with a 94% success rate and a false-negative rate (FNR) of only 4%. As a consequence, assaults are rather high. It indicates a high proportion of false-positive results relative to total positive results. To mistakenly label genuine hostile IoT devices as harmless is a major security risk. Figure 5 shows the outcomes of the CNN model. Figure 8 shows the cnn+LSTM model.

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022

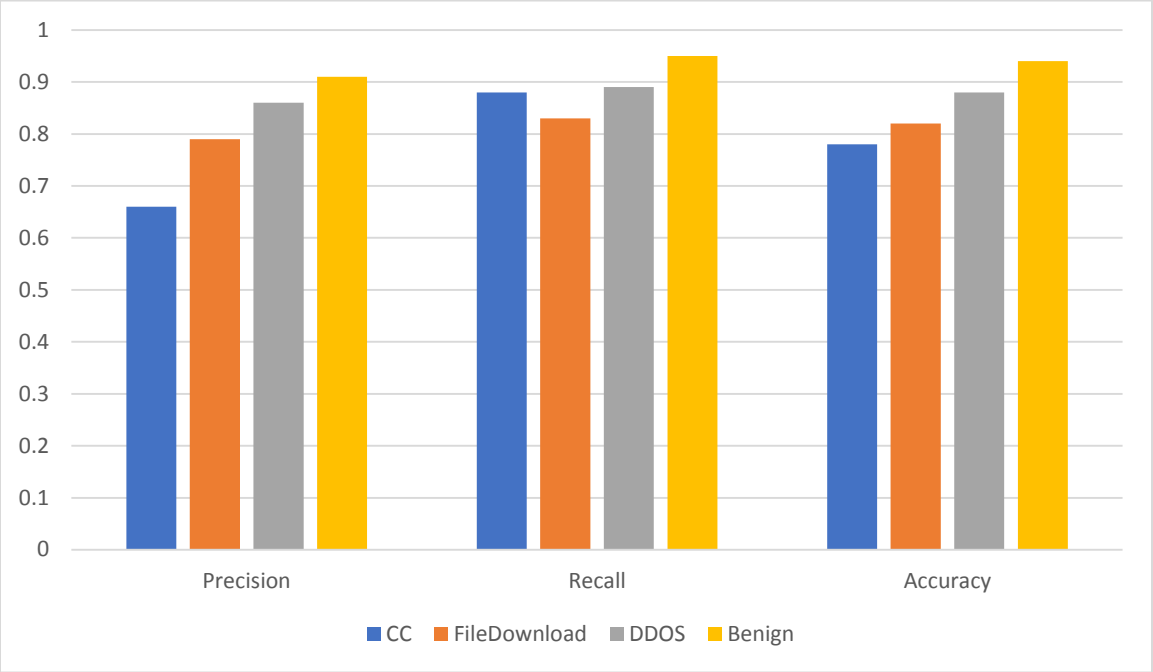

Figure 5: The results of the CNN model.

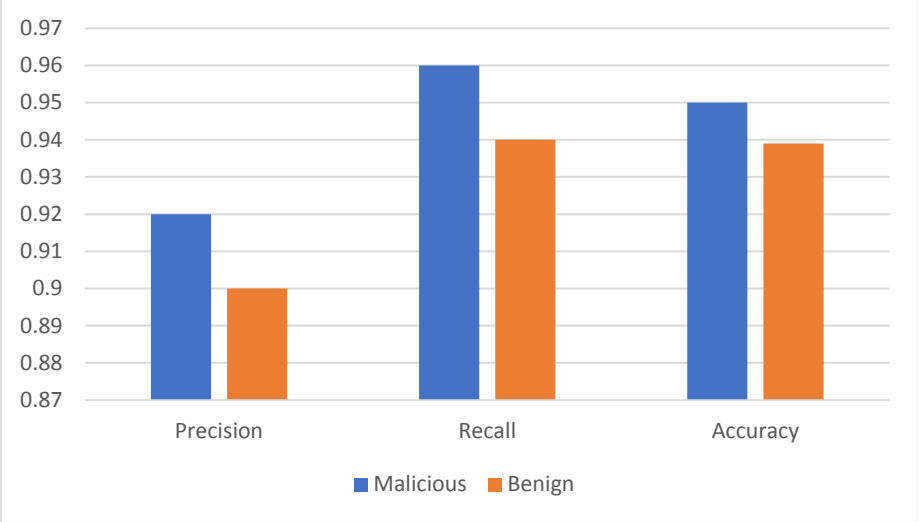

Figure 6: The results of the CNN+LSTM model.

The description of the data sets is as:

799,479 regular streams and 871,914 APT fluxes for a total of 1,671,393 flows.

The 1065 IP addresses contain 139 APT-attacking IPs and 926 regular IPs.

The Malware Capture CTU-13 dataset includes 29 network traffic files, including experimental data acquired and analyzed from 6 different kinds of harmful programs from APT assaults

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022

The dataset is split into a train set, a validation set, and a test set split based on the number of IPs. The training dataset to train the system, the validation data is used to build the model during training, and the test set is used to assess the performance of the model. The paper's assessments are conducted only on the test set, which is the data that now the model never has seen during development.

Hypothetical Situation

In this research, we perform experiments in accordance with the following conditions in order to assess the efficacy of the suggested method:

Scenario 1: The MLP and LSTM separate deep-learning models are utilized as a comparison and assessment foundation to gauge the efficacy of the APT attack detection technique employing integrated deep-learning models intuitively. Changes are made to the model's settings, including the number of hidden layers and the total number of nodes, throughout the experiment to see how well the model performs.

The second scenario involves a CNN-MLP mixed deep learning model for feature extraction and classification of frames through Softmax regression. The amount of convolutional, the amount of MLP layers, and the number of nodes in MLP layers are all variables that may be adjusted and tweaked.

Three, we use a CNN-LSTM hybrid deep learning algorithm to identify flows and then use Softmax regression to identify those flows. The CNN layers of LSTM layers are two of the factors that may be adjusted to locate the best model. Figure 7 shows the accuracy of different scenarios.

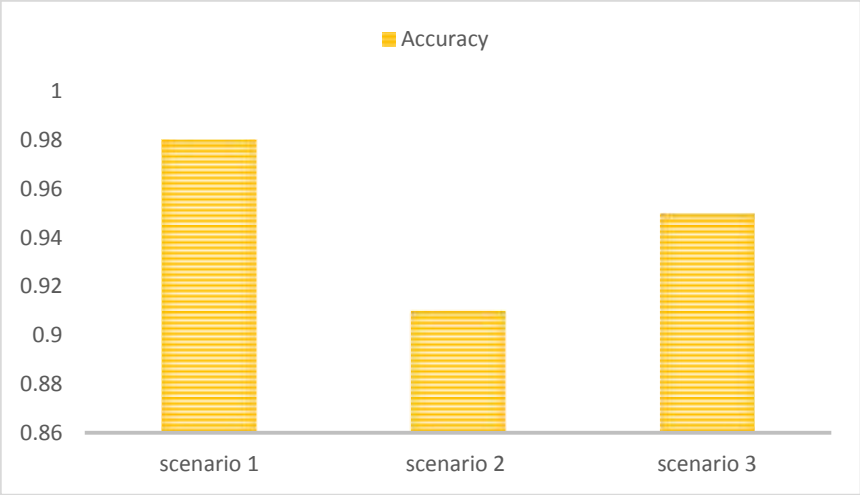

Figure 7: The accuracy under different scenarios.

5. Conclusion

In this work, we introduce a comprehensive work on intelligent multimedia forensics. Corresponding to our work, intelligent multimedia forensics involves three perspectives deep learning forensics, multimedia forensics, and network/IoT forensics. We show the effect of using deep learning techniques on protecting multimedia and network/IoT from different attacks, preventing forgery, and protecting deep learning models from adversarial attacks. We introduce a new category of adversarial attacks and defenses against them according to the training and testing phase. We provide a comprehensive work on steganography, steganalysis, watermarking, inter and intra frame, copy-move, and deep fake on multimedia using deep learning, providing advantages and disadvantages for each method. Deep learning algorithms introduce in each layer of Network/IoT to prevent DDoS, Botnet, intrusion detection, etc. Moreover, Deep learning algorithms still face many limitations and problems that are concluded in the challenges section. In conclusion, although many deep learning-based intelligent multimedia forensics methods have achieved notable results, there is still much future research work in this area that must be discovered.

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022

References

[1] S. Agarwal, H. Farid, T. El-Gaaly, and S.-N. Lim, "Detecting deep-fake videos from appearance and behavior," in *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2020, pp. 1-6.
[2] R. Agrawal and D. K. Sharma, "A Survey on Video-Based Fake News Detection Techniques," in *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2021, pp. 663-669.
[3] X. Liu, L. Xie, Y. Wang, J. Zou, J. Xiong, Z. Ying, *et al.*, "Privacy and Security Issues in Deep Learning: A Survey," *IEEE Access*, 2020.
[4] I. Castillo Camacho and K. Wang, "A Comprehensive Review of Deep-Learning-Based Methods for Image Forensics," *Journal of Imaging*, vol. 7, p. 69, 2021.
[5] N. Koroniotis, N. Moustafa, and E. Sitnikova, "A new network forensic framework based on deep learning for Internet of Things networks: A particle deep framework," *Future Generation Computer Systems*, vol. 110, pp. 91-106, 2020.
[6] N. Koroniotis, N. Moustafa, and E. Sitnikova, "Forensics and deep learning mechanisms for botnets in internet of things: A survey of challenges and solutions," *IEEE Access*, vol. 7, pp. 61764-61785, 2019.
[7] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*, 2017, pp. 39-57.
[8] P. Yang, D. Baracchi, R. Ni, Y. Zhao, F. Argenti, and A. Piva, "A survey of deep learning-based source image forensics," *Journal of Imaging*, vol. 6, p. 9, 2020.
[9] M. Barni, Q.-T. Phan, and B. Tondi, "Copy move source-target disambiguation through multi-branch CNNs," *IEEE Transactions on Information Forensics and Security*, 2020.
[10] M. Chaumont, "Deep learning in steganography and steganalysis," in *Digital Media Steganography*, ed: Elsevier, 2020, pp. 321-349.
[11] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE transactions on neural networks and learning systems*, vol. 30, pp. 2805-2824, 2019.
[12] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, 2018, pp. 268-282.
[13] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, *et al.*, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.
[14] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.
[15] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574-2582.
[16] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*, 2016, pp. 372-387.
[17] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.
[18] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1765-1773.
[19] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh, "Ead: elastic-net attacks to deep neural networks via adversarial examples," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.
[20] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, pp. 828-841, 2019.
[21] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," *arXiv preprint arXiv:1801.02610*, 2018.
[22] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the GAN: information leakage from collaborative deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 603-618.
[23] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 2017, pp. 506-519.

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022

[24] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 15-26.
[25] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *International Conference on Machine Learning*, 2018, pp. 2137-2146.
[26] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber, "Natural evolution strategies," *The Journal of Machine Learning Research*, vol. 15, pp. 949-980, 2014.
[27] M. Alzantot, Y. Sharma, S. Chakraborty, H. Zhang, C.-J. Hsieh, and M. B. Srivastava, "Genattack: Practical black-box attacks with gradient-free optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2019, pp. 1111-1119.
[28] A. Gleave, M. Dennis, C. Wild, N. Kant, S. Levine, and S. Russell, "Adversarial policies: Attacking deep reinforcement learning," *arXiv preprint arXiv:1905.10615*, 2019.
[29] C. Yang, Q. Wu, H. Li, and Y. Chen, "Generative poisoning attack method against neural networks," *arXiv preprint arXiv:1703.01340*, 2017.
[30] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, *et al.*, "Poison frogs! targeted clean-label poisoning attacks on neural networks," *arXiv preprint arXiv:1804.00792*, 2018.
[31] O. Suciu, R. Marginean, Y. Kaya, H. Daume III, and T. Dumitras, "When does machine learning {FAIL}? generalized transferability for evasion and poisoning attacks," in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 1299-1316.
[32] G. Lovisotto, S. Eberz, and I. Martinovic, "Biometric backdoors: A poisoning attack against unsupervised template updating," in *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2020, pp. 184-197.
[33] W. Jiang, H. Li, S. Liu, X. Luo, and R. Lu, "Poisoning and evasion attacks against deep learning algorithms in autonomous vehicles," *IEEE transactions on vehicular technology*, vol. 69, pp. 4439-4449, 2020.
[34] L. Truong, C. Jones, B. Hutchinson, A. August, B. Praggastis, R. Jasper, *et al.*, "Systematic evaluation of backdoor data poisoning attacks on image classifiers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 788-789.
[35] T. Liu, W. Wen, and Y. Jin, "SIN 2: Stealth infection on neural network—a low-cost agile neural trojan attack methodology," in *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2018, pp. 227-230.
[36] H. Kwon, H. Yoon, and K.-W. Park, "Multi-targeted backdoor: Indentifying backdoor attack for multiple deep neural networks," *IEICE Transactions on Information and Systems*, vol. 103, pp. 883-887, 2020.
[37] J. Dai, C. Chen, and Y. Li, "A backdoor attack against LSTM-based text classification systems," *IEEE Access*, vol. 7, pp. 138872-138878, 2019.
[38] X. Chen, A. Salem, M. Backes, S. Ma, and Y. Zhang, "Badnl: Backdoor attacks against nlp models," *arXiv preprint arXiv:2006.01043*, 2020.
[39] L. Sun, "Natural backdoor attack on text data," *arXiv preprint arXiv:2006.16176*, 2020.

Doi : https://doi.org/10.54216/FPA.090203
Received: May 19, 2022 Accepted: September 11, 2022