New data poison attacks on machine learning classifiers for mobile exfiltration

Miguel A. Ramirez¹, Sangyoung Yoon¹, Ernesto Damiani¹, Hussam Al Hamadi¹, Claudio Agostino Ardagna², Nicola Bena², Young-Ji Byon³, Tae-Yeon Kim³, Chung-Suk Cho³, and Chan Yeob Yeun¹

¹Center for Cyber-Physical Systems, EECS Department, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates
²Dipartimento di Informatica, Universita degli Studi di Milano, Milano, Italy
³Department of Civil Infrastructure and Environmental Engineering, Khalifa University of Science and Technology, Abu Dhabi, UAE

Corresponding author: Chan Yeob Yeun (e-mail: chan.yeun@ku.ac.ae).

This work was supported in part by Technology Innovation Institute (TII) under Grant 8424000394.

I. INTRODUCTION

Over the last couple of years, machine learning (ML) models demand the deployment of additional techniques in order to address security related factors since new vulnerabilities are being discovered and could pose a threat to the integrity of the ML model being the target of an attacker [1]. An attacker could exploit such vulnerabilities causing a negative impact on the performance of the ML model. It has been proven plausible to maliciously compromise the training data in order to affecting the model decision-making process which eventually causes a utter malfunction during testing (or inference) phase.

The need of public and available data is continuously on demand by plenty ML models. A clear example can be seen in smart city systems wherein large amounts of data are gathered by numerous sensors, such as smartphones. Then it comes without saying that the consequences of an attack targeting smart city systems could be devastating and very feasible due to the system heavily dependence over public data. The intention of this paper is directed towards gathering the most representative attack and defense approaches around data poisoning [2]. Data poisoning (DP) attacks aim to compromise the integrity of a target model by performing alterations to the required dataset used by the model during the training phase. This causes the model to misclassify samples during the testing phase, representing then a significant reduction in the overall accuracy.

For all the reasons previously mentioned, there is the urge to develop more advanced defense mechanisms, aiming to enhance robustness of the model to fight back potential DP attacks.
attacks occurring while training, the further understanding of such vulnerabilities could offer promising results directed towards the development of a defense mechanism capable of detecting and even mitigating the effects of the poisoning data, making by then ML classifiers with a higher resiliency than current ones. This last point has been one of the main center of focus when it comes to malware classification, application of which the target ML models in this work will address specially.

The introduction of a new approach that attains a certain level of immunization against DP in the fields of malware detection involving mobile ex filtration data [108], being this a common topic of interest regarding smart devices and smart cities environments. The main contributions of this paper are listed as follows:

- This paper covers diverse topics around machine learning security fundamentals, assumptions of attack and defense scenarios, types of data manipulation and vulnerabilities.
- Related work on potential threats involving data poisoning attacks and related defense mechanisms are showcased, particularly towards manipulation of mislabeled training data such as label-flipping techniques.
- The methodology entailing crafting a data poisoning attack based on label-flipping, targeting a malware ML classifier. A further evaluation of the effects of the poisoning attack on different ML models, comparing the results obtained with each other, quantifying the effects of the attack and vulnerabilities of each model.
- Open problems and future research work towards the field of machine learning security is discussed with special emphasis in analyzing more complex label-flipping attack scenarios and other defense mechanism oriented towards detection and mitigation against DP attacks are examined as well.

This work is organized as follows: Knowledge background information is explained along with related work on various types of data poisoning attacks and defense mechanisms on Section II. The methodology explaining both the development and evaluation of the proposed attack on target ML models, as well as for the proposed defense mechanism against the proposed attack can be found on Section III. Future work and research directions related to this work can be found on Section IV. Finally, the conclusion and final remarks are shown in Section V.

II. KNOWLEDGE BACKGROUND

In this section, an overview of the properties around attacks and defenses is analyzed. The fundamentals of relevant topics involving security in machine learning models are discussed, mainly centered in the assumptions of the attacker and different types of attacks throughout the machine learning life-cycle.

A. MANIPULATION

Training data manipulation [3] is one of kinds of DP attacks by corrupting (or poisoning) the training data during the training phase with the aim to utterly jeopardize the integrity of the ML classifier having trained a wrong classifier, examples of techniques often used by attackers are the modification of data labels, injection of malicious samples and manipulation of the training data. As a result, the overall damage to the target ML model can only be appreciated at the inference phase, having the accuracy of the model drastically reduced. This effect is commonly referred to as accuracy degradation.

Input manipulation is triggering a machine learning system to malfunction by altering the input that is fed into the system [4]. It would be in the form of an altered image by adding noise or another input that causes the classifier to perform a wrong prediction. Adversarial attacks [5]–[9] take place during the inference phase, having a ML model already trained and as a result any prediction from the model is considered of high confidence [10]. Depending on the goal of the attacker an adversarial attack can fall in one of two categories. Referring to a Targeted attack when the input in the form of crafted adversarial examples lead to the target model to misclassify the samples into a specific class defined by the attacker [11], [12]. In contrast, in a Non-targeted attack the crafted adversarial examples aim to cause the target model to misclassify. Nonetheless, there is no need nor interest from the attacker to misclassify into a particular class apart from the correct one. Evasion attacks are also another kind of input manipulation and are different from adversarial attacks in the sense that evasion attacks do not require any knowledge over the training data [13].

B. ASSUMPTIONS OF ATTACK-DEFENSE

Security threats to machine learning models are divided into data poisoning (DP) attacks and adversarial attacks, acting the former during the training phase and the later during testing, this difference is shown in Figure 1. For the purposes of this work, data poisoning attacks will remain as the main topic of interest.

One of the most common schemes of the attacker is injecting malicious samples into the target model’s training set, corrupting either the feature values or labels of the training samples. Affect the ML model boundaries by causing significant deviations to a point where the model’s reliability is
completely devastated, thus leaving the model susceptible to make wrong predictions. Then the goal of the attacker then is to disrupt the training process aiming to significantly reduce the performance of the target model, causing an accuracy-drop; also generating increasing misclassification rate of the samples during testing.

Assumptions of attackers refer to the prior knowledge (implicit or explicit) over the target model of interest of the attacker, entailing the resources available to the attacker. When conducting experiments, the devised attack is meant to be evaluated against a defense, both attack and defense state assumptions must be declared assuring the conditions that guarantee either scheme’s effectively (e.g. attaining the defense to defeat an attack, or vice versa). However, various DP attacks have shown to be successful in spite of having very little knowledge of the target model. An example of this is described in the work [13], directing a DP attack scheme to naive Bayes email-spam filters by simply sending ‘ham-like’ emails using a black-listed IP address as the sender, then being threatened and labeled as spam, nonetheless these corrupt data will be inevitably used by the spam filter for further training.

The assessment of the influence of the attacker over the training data is commonly defined as the attacker’s capability. Being the primary interest of the attacker is to alter either the feature values or labels as part of the training set. Nevertheless, the attacker is restricted to poison a limited number of samples, commonly referred as a ratio of less than 30% of the total data samples. More optimized poisoning algorithms have been in constant development during the last decade, aiming to maximize the accuracy degradation and minimize the number of poisoning samples needed to perform the attack.

C. METRICS OF INTEREST

The attack success related to DP attacks is estimated based on the amount of degradation shown by the target model performed during the testing phase. This can be further appreciated once computing the decision matrix, observing the overall misclassification in each class displaying: true positive, false positive, true negative and false negative. Moreover, the effectiveness of the attack is shown in the form a significant drop in the overall accuracy, this is referred as accuracy degradation.

The employment of additional metrics besides measuring the accuracy have been proposed to reflect and analyze in further detail the overall performance of the target model and by then make comparisons to other model performance. Various metrics for artificial intelligence have been proposed by multiple standard bodies including the International Organization for Standardization (ISO) [13] and the National Institute of Standards and Technology (NIST) [22]. The metrics for AI typically includes the accuracy, the precision, the recall, the Receiver Operating Characteristic (ROC) and its area1 [23]. Other approaches such as the one from Biggio [24] introduces the security evaluation curves as a way to characterize the performance of a ML model against an intended attack considering various level of knowledge from the attackers side. Thus this approach accomplishes a comprehensive evaluation of the overall security of the model; and by doing so, enables another means to compare assorted defense techniques.

D. ADVERSARIAL CAPABILITIES

In the testing phase, the attacker naturally will aim to attain further knowledge over the target model in order to increase the effectiveness of adversarial attacks, this can be in the form of any of the following five factors: Feature space, classifier type (e.g. DNN or SVM), classifier learning algorithm, classifier learning hyperparameters and the training dataset.

- A white-box assumption is commonly defined as an scenario in which the attacker does have complete knowledge over all the five elements already described previously, as well as any defense mechanism already set on top of the model [25]–[27].
- A black-box assumption is the opposite to white-box assumption, when no knowledge of the target model, albeit query it can be plausible. Nonetheless, it is important to remark that, just having access to the training data grants the upper hand to the attacker over any defender, representing this training data the unadulterated ‘clean’ dataset, in question [28]–[33].
- A gray-box assumption is often referred as a middle ground between white-box and black-box scenarios, where the prior knowledge on the attacker’s side can include the feature space, the target classifier; this includes the model architecture, model parameters and the training dataset; however, the defense mechanism on top is unknown to the attacker. The gray-box setting usually is used to evaluate the defense against the adversarial attack [17].

E. EXPLAINABLE ARTIFICIAL INTELLIGENCE

Explainable Artificial Intelligence (XAI) is a concept that looks for the development of artificial intelligence models with a further improved understanding regarding the main aspects of their decision-making process, this is for us humans to understand why and how a model approaches a prediction. The aim of explainable AI is to develop more reliable strategies that will endow models with a higher level of transparency while retaining high-performance levels, mainly in the form of accuracy [106].

XAI is extremely necessary nowadays in various models since explainability is often translated into an adequate level of trust in the predictions. Regardless of high performance, a model’s predictions cannot always be taken as ground-truth, especially in applications of critical importance in terms of reliability, such as with cybersecurity as well as other future-generation AI partners. In the field of cybersecurity transparency is a must, the lack of it represents dangers.
Therefore it has become essential to maintain a good balance between explainability and performance, looking further into the trade-offs they both confer in the field of cybersecurity and the newly introduced technologies around it [107].

Decision trees represent one example of a good approach towards high explainability since it confers higher transparency than other ML models, enabling the user to understand the decision-making process more easily. Nonetheless, deep learning models perform better than decision tree models, it results to be the algorithm with less explainability.

F. RELATED WORK: LABEL-FLIPPING ATTACKS

The most common way to generate this kind of poisoning is by maliciously tampering the labels in the data [31], this can be easily achieved by just flipping labels, thus generating mislabeled data, this is shown in 2.

Label flipping can be performed either randomly or specifically depending on the aims of the attacker; the former aims to reduce the overall accuracy of all classes, the later does not aim to perform significant accuracy reduction, rather it is focus on the misclassification of a determined class in particular.

In the following paragraphs several examples of data poisoning attacks on different types of models are discussed. The center of focus is directed towards data poisoning attacks performed during the training phase involving label-flipping techniques against ML classifiers exclusively.

Paudice et al. [32] proposes an optimal label flipping poisoning attacks compromising machine learning classifiers. Label flipping actions are performed following an optimization formulation focused on maximizing the loss function of the target model. This approach is considered computationally intractable due to the inclusion of heuristic functions enabling the label flipping attacks to downscale the computational cost.

The applications of this approach limits itself to binary classification problems and the assumptions of the attack involves complete knowledge over the learning algorithm, loss function, training data and also the set of features used by the ML classifier, turning it basically into an attack on a White-box model. Albeit the list of assumptions appeal to unrealistic scenarios, the analysis emphasizes on worst-case scenarios. The effectiveness of the propose method is demonstrated in three datasets from UCI repository: MNIST, Spambase and BreastCancer; succeeding in increasing the classification error by a factor of 6.0, 4.5 and 2.8, respectively [35].

Xiao et al. [36] reports a successful attack on a SVM model after performing label flipping using an optimized framework capable of procure the label flips which maximizes potential classification errors, causing a significant reduction in the overall accuracy of the classifier. As a potential drawback, this technique naturally implies a high computational overhead as a main requirement.

Federated Learning (FL) in recent years has become relevant in privacy-preserving applications. This is possible since the data gathered from each device or worker is kept locally stored in each device [64], then enabling the training process of a sub-machine learning model individually. As a second step, only the resulting gradients obtained after training are exchanged to a centralized server instead of the raw data, then the centralized server performs the entire training life-cycle by multiple iterations until attaining a desirable accuracy. Due to the nature of FL, malicious users could perform a label-flipping attack [39] by deliberately inserting crafted gradients leading to classification errors during the test phase.

In the past it is been proven that a single poisoner can undermine the whole training process and as a result the integrity of the model. Therefore, a robust FL model needs to regard on concerns related not only to data privacy, but also rely on a certain degree of resilience against poisoning attacks and data manipulations.

G. RELATED WORK: DEFENSES

k-Nearest-Neighbours defense scheme [35] is designed to detect malicious data and counteract the effects of the same, being this defense referred as Label sanitization (LS). Label sanitization (LS) bases its defense on the decision boundary of SVM, observing the remoteness of the poisoned samples, commending these samples to be re-labelled. Steihhardt et al. [58] proposes a nearest-neighbor-based mechanism to detect outliers and SVM optimization right afterwards, getting as a result a domain-dependent upper bound associated to the estimated highest drop in accuracy due to a DP attack. A special assumption is made for this scenario, declaring the removal of non-attack outliers inconsequential to the performance of the target model.

Liu et al. [34] addresses the privacy/defense related issue in FL models by showcasing a novel framework called privacy-enhanced FL (PEFL). PEFL grants the central server the ability to detect malicious gradients and block poisoner workers. By comparing the malicious gradients, submitted by the poisoner workers, as a set of parameters to the same ones belonging to the honest workers; the difference between malign and benign gradient vectors can be evaluated by calculating the Pearson correlation coefficient [65]. Abnormality behavior is related to a lower correlation coefficient, then the action of the defense mechanism consists on simply setting the weights of the malign model to zero.
PEFL claims superiority among other similar systems such as Trimmed Mean [64], Krum [66] and Bulyan [67]. Since the proposed scheme does not assume to have any knowledge of the total number of poisoners, posing then a more appropriate defense more suitable for real-case scenarios. Furthermore, PEFL poses a higher resilience against accuracy drops compared to Bulyan and Trimmed Mean due to weight adjustment performed on each gradient which guarantees trustworthiness within the remaining parameters. Observing in the end a maximum attack success rate of 0.04, evidence of the robustness of the model against label-flipping.

SVM Resistance Enhancement [68] is targeted to avoid label-flipping attacks, being SVM particularly vulnerable against this kind of attacks, causing total misclassification due to the computation of erroneous decision boundaries. Thinking ahead about the effects of suspicious data points within the SVM decision boundary, the proposed approach considers a weighted SVM accompanied by KLID (K-LID-SVM). This work introduces K-LID, a new approximation of Local Intrinsic Dimensionality (LID), metric associated to the outliers in data samples. K-LID computation relies on the kernel distance involved in the LID calculation, allowing LID to be computed in high dimensional transformed spaces. Obtaining by such means the LID values and discovering as a result three specific label dependent variations of K-LID capable of counter the effects of label-flipping.

K-LID-SVM attains higher overall stability against five different label-flipping attack variants: Adversarial Label Flip Attack (alfa) attack, ALFA based on Hyperplane Tilting (alfa-tilt), Farfirst, Nearestl and Random label flipping; using five different real-world datasets for a benchmark test: Acoustic, Ijcnn1, Seismic and Splice and MNIST. The defense system attains a drop of 10% on average in misclassification error rates, this method can distinguish poison samples from honest samples and then suppress the poisoning effect. Therefore, it succeeds in decreasing the potential magnitude of the attack significantly and demonstrating a superior performance than traditional LID-SVM.

In this chapter we will propose an attack strategy able to effectively compromise the integrity of more than one type of ML classifier, causing a significant drop in accuracy and miss classification in one or various classes.

H. MOBILE MALWARE CLASSIFIER

Morcos et al. [108] develops a mobile exfiltration data. This dataset is considered the main data of interest in this work as it represents the data used to feed the ML model comes from a mobile exfiltration detection engine for Android phones. From any application in executions system calls can be observed and studied as partial information on onboard activities, as seen in [4].

Based on this onboard activities system calls can be extracted, then the statistical behavior of a potential malware can be appreciated by the frequency in which determined system calls appear during execution, forming an histogram. The overall approach is described in five steps:

- Defining app and system-call activity data segmentation and mapping to activity representation.
- Identifying normal behavior for a given configuration.
- Identifying exfiltration behavior from sandboxing.
- Training stateful ML models.
- Deploying and comparing.

The feature engineering process entails then selecting the most representative system calls for this malware detection as the features of interest, this process receives the name of Monolithic approach. Once setting this representation this data can be feed into an ML model classifier in further steps, then a ML model can generalize well among the malware data and detect when exfiltration or data leakage occurs.

Another approach that has resulted more accurate for malware detection is by declaring the features as 3-gram of system calls. This is referred as the 3-gram approach, here each feature refer to a specific sequence of three system calls; therefore only after having recorded the set of 3 system calls in the declared order a value can be set. The 3-gram approach serves as the main and only data representation of interest in this work.

In the 3-trigram approach, different variations of the gathered application data have been created by performing a clustering process at different levels of the process tree. In the
I. DISCUSSION

It is important to highlight the attacker’s capability in other scenarios and the assumptions imposed by the attacker, being sometimes to optimistic while in other scenarios represent more realistic conditions [72]. In the literature there are many examples related to assumptions in regards to the capability of the attacker. For instance, TRIM [76] assumes the ratio of the poisoning examples declared by the attacker as known. Deep-kNN [79] assumes access to ground-truth labels allowing the system to compare each sample’s k neighbors with the class labels.

It is important to make a clear distinction on the properties of the crafted poisoning data, poisonous data obtained by performing label-flipping is one. Albeit, the scheme promises high accuracy degradation, it is far from representing the most effective option for an attacker. This is because label-flipping is considered among the most basic poisoning techniques and most of the existing defense mechanism can detect these ones as outliers and reject them with relatively ease.

III. METHODOLOGY

The dataset to be employed is the mobile exfiltration data in the work of Morcos et al. [108]. This dataset is considered the main data of interest in this work as it represents the data used to feed varios ML models.

The distribution of the data has been analyzed using the software Weka, having distinguished labeled samples in a histogram for each feature of the dataset. The results of this analysis show no features with separable data as it can be seen in Figure 6.

A. TRAINING ML CLASSIFIERS

Five different ML model have been selected as the target of the intended attack: Random Forest, Decision Tree, SVM, Logistic Regression and KNN. The development of these five ML classifiers for malware detection has been performed as case study, every ML model has been trained and tested with a part of the same mobile ex-filtration dataset. The dataset has been partitioned in 60% for training, and 20% for testing. The code can be found in Appendix A.1.

The main purposes of this case study is to perform a comparison in performance, for this step we assume the existence of only clean data (ground-truth) with no existence of poisonous samples. The metrics of interest as part of the testing results are: Accuracy, Precision, Recall, F1 score and AUC (Area Under the Curve), the importance of the latter will be explained in the next subsections. The results obtained with each ML algorithm are shown in Figure 8.

Being accuracy the most important metric for our study. Note that Random Forest is the model with the highest accuracy of them all, with an accuracy of 99.54%, however it is important to remark that all the other machine learning models perform with an accuracy over 97%, proving a high level of reliability among all the ML algorithms of interest.

The confusion matrix for each ML model is computed, this allows the correct visualization of the performance of every algorithm, comparing in 2 dimensions the number of actual and predicted results; true positives and true negatives, false positives and false negatives, respectively. It can be appreciated in Figure 9 the number of miss-classified samples (false positives and false negatives) compared to the cases where predicted samples have been correctly classified according to their respective class.

B. PROPOSED DATA POISONING ATTACKS

The proposed attack approach is based on Label Flipping [3], as explained in the previous chapter, this method tampers the labels of the samples injected into the training set, generating the poisoning data. The proportion of poisoning samples with respect to the total number of samples contained in the training set figures as an important parameter to evaluate, naturally the most efficient LF attacks are the ones that require the less number of poisonous samples, this has to do with the attacker capability to inject poison samples in high numbers.

It is important to remark that the target ML models of interest have been trained in the previous steps with clean data, assuming no poisoning samples. Then, the proposed attacks explained in the following sections will be acting during the re-training of the ML model, then a comparison between its original performance (without poisoning samples) and the performance of the model after the attack in question can be further analyzed.

1) Label-flipping attack: Level 1

A random label-flipping attack is one if not the simplest type of attack to craft against a ML classifier, nonetheless it is considered one of the attacks that can cause the most damage to mostly any ML classifier’s accuracy. It consists in switching the labels associated to each sample in a random way and inject them into the training set of the model. The mobile ex-filtration dataset contains a feature indicating the damage to mostly any ML classifier’s accuracy. It consists in switching the labels associated to each sample in a random way and injecting them into the training set of the model. The mobile ex-filtration dataset contains a feature indicating the label of the sample, this can be either Benign “0” or Malign “1”.

The chart in Figure 11 and 10 well serves as a comparison between the model original performance (0% poisoning samples) and the performance of the model after the attack in question can be further analyzed, the results are displayed by performing a variation in the proportion of poisoned samples, accounting for a 25%, 50%, 75%, 100% poisoned samples.
The series of steps necessary to perform the proposed attack are explained in better detail in the form of a process map in Figure 12.

We have computed the confusion matrix for each ML model. For this case we are interested in reporting the results of the confusion matrix when accounting with ‘clean’ data only and also under the each poisoning scenario when varying the ratio of poisoning samples. The intention with this is to appreciate and compare clearly the steady increase in the miss classification rate by the model when being presented an increased number of poisoning samples. The effects of the overall missclassification of the two classes of interest have been studies and reported for each algorithm, comparing the effects of the attack with the ‘clean data’ condition (no poisoned samples present). For this please refer to Figures 13, 14, 15 and 16.

The same comparison can be made when computing the ROC curve for each model under ‘clean’ data assumption as well as for the same poisoning scenarios already described before. Then a degradation in the AUC by increasing the ratio of poisoning samples is visible, indicating a reduced true positive rate with an increased false positive rate. This behaviour can be better appreciated in Figures 17, 18 and 19.

2) Label-flipping attack: Level 2

As a second line of attack, rather than randomly, the aim now is to perform label flipping targeting a specific class [3]. Then causing a significant reduction in the accuracy of the target model is no longer the priority of the attacker, but the primary aim of the attack is to deliberately achieve the misclassification of a determined class in particular. Naturally a potential attacker will opt to misclassify the malign malware samples by switching their labels from ‘malign’ to ‘benign’. The aim of the attacker in question is to tamper the model’s ability to recognize and classify malign malware as ‘malign’, thus letting through any type of threatening malware to a computer system.

In consequence, during the testing phase the machine learning model will misclassify malign samples as benign instead, tampering completely the decision-making process of the classifier in question. In this case, the assumption of the attacker is the following. The attacker must have prior access to the dataset and know the set of the features in the dataset.
Then there’s is the possibility for the attacker to employ Explainable AI in order to determine the feature/predictor importance among all the ones in the dataset and figure out the number of features that generate the most impact in the classification task, this classification task is directed towards the samples that are labeled as ‘malign’.

For this purposes, we have employed Explainable AI to emulate the process a potential attacker will undergo by generating decision trees (Figure 20) using the tool IBM SPSS Modeler and identify the features with the highest importance for the classifier. These features can be seen in the Pareto diagram in Figure 21.

We will refer to the process map shown in Figure 22 to better understand the algorithm. Once selected the features of interest we have defined a ‘decision criteria’ to determine the samples subject to the attack. This decision criteria is based on the following conditions.

- Sample must be originally labeled as ‘malign’.
- The sample must have values associated to the determined features of interest (monogram or 3-gram of system calls) that overpass a determined threshold. In this case we have defined that threshold as any value that is more than “0”.

The chart in Figure 23 well serves as a comparison between the model original performance (0% poisoning samples) and the performance of the model after the proposed attack. As we did for the label flipping attack: Level 1. Again, the results are displayed by performing a variation in the proportion of poisoned samples, accounting for a 25%, 50%, 75%, 100% poisoned samples. For this case Figure 23 includes the results of the DP attack: Level 1 as a mean of comparison.
Notice that the effect of such an attack will not only affect the accuracy of the model but also to increase dramatically the number of false negatives compared to false positives, this behavior can be better appreciated in the confusion matrix for the decision tree algorithm in Figure 24.

This impacts specially the recall metric which is associated to the number of false negatives, as in the previous attack (Level 1: random label-flipping) this can be better appreciated by comparing the generated ROC curves in Figure 25, this ROC depicts the performance of the decision tree algorithm.

C. DISCUSSION

Nowadays, there is an urgent need of an attack-agnostic defense system, since most of the works in the literature contemplate combating one type of attack in specific. Nonetheless label flipping still represents an important way for benchmark to assess the robustness of any given type of ML classifier. Therefore, all the proposed approaches have been studied theoretically as viable candidates for a defense against DP.

Label flipping attacks nowadays still impact negatively machine learning classifiers once identified new vulnerabilities in specific scenarios related to specific applications, the nature of the dataset of the malware detector that is being studied could promise an outstanding area of opportunity. In addition, label-flipping still represents an important way for benchmark to assess the robustness of any given type of ML classifier.

Therefore, the intended research proposal aims to craft an attack more than just one type of machine learning model and analyze their response and drop in performance. Particularities regarding the decision-making process of each algorithm will come into play, nurturing posterior analysis and further insights around the features involved in the malware dataset. Once having exploited and analyzed all upcoming vulnerabilities, an algorithm for a defense approach will become more clear and more suitable alternatives suited for either detection of mitigation tasks will be proposed, developed in further detail and tested in conjunction with different ML classifiers in order to assess their effectiveness and feasibility.
• Perform data pre-processing (avoid unbalanced data, missing values and skewed features)
• Split data into training and testing sets

• Define a percentage as the desired proportion of samples to be manipulated (25%, 50%, 75%, 100%)
• Randomly selection of a sample
• Perform label flipping on sample
• Repeat the steps until desired percentage of poisonous samples is reach

• Tune/select hyperparameters of the model
• Train ML model (Logistic Regression, Decision Tree, Random Forest, k-NN)

• Test the model using ‘clean’ data
• Obtain metrics: Accuracy, precision, recall and f1 score
• Compute a confusion matrix (examine misclassification rate)
• Compute ROC curve and calculate AUC

FIGURE 12. Process map: DP attack: Level 1 (Random Label-flipping)

FIGURE 13. Decision tree confusion matrix comparison

FIGURE 14. Random forest confusion matrix comparison

FIGURE 15. Logistic regression confusion matrix comparison

FIGURE 16. KNN confusion matrix comparison
lead to critical undesirable outcomes. Any breakthrough in the field of machine learning will always implicitly allow the introduction of new vulnerabilities. Such vulnerabilities will always pose an open window of opportunity for adversarial entities to exploit, leading as a consequence to an opportunity to develop new defenses counterbalance the outcome in the same matter as it occurred with the invention of the internet and the introduction of cyber-security systems.

Assumptions for both proposals for an attack and defense will be imperative. This will imply reporting the specific details regarding the required knowledge over the ML model and the training set needed to attain a successful attack over a classifier. Similarly, it will become substantial to describe the specific instances in which the defense will be deployed. Regardless of the nature of our proposed defense system, it is important to maintain an open mindset; one which might well consider very optimistic or ideal scenarios, as well as others not so ideal, but rather practical for real life applications. In relation to this last point, we could add as a special consideration that; in contrast to other works, our work might not be suited to be assessed or tested with a common malware dataset, but rather with another kind of malware exfiltration engine, then a this work could promise great potential to become a novel solution.

In this section we propose several potential and possible solutions to counteract the proposed series of label flipping data poison attacks depicted in this work. For this, we will showcase the different approaches that could be followed in order to propose a defense mechanism.

A. FIRST APPROACH: DETECTION AND MITIGATION
As seen in random label flipping attack type 1, the accuracy drop did not overpass 50%, this is due to the presence of low confidence points near the decision boundary, regardless of their assigned label (benign/malign). Such behavior could be the result of the nature of our binary classifier and needs further study in the short term to understand the ML decision process in better detail. Then this might be a good area of opportunity. As a first step, it could be possible to use the data gathered from the five ML models of interest once poisoning the data to find an acceptable threshold that could be used to reject this data and remove these points from the training set, discarding them as outliers.

Then, this approach will serve as a detection mechanism, being able to accept or reject the samples once identifying them within the acceptable threshold. Thereby detection techniques are meant to spot abnormalities in the form of noisy points (outliers) or due to the presence of poison samples themselves, this often requires constant monitoring of the target model observing the effects of the poisoning data over the performance on the model.
On the other hand, there is another approach known as defense mitigation. A defense system capable of mitigating the effects of data poisoning, reducing the overall classification error as a result. A defense mitigation technique is very suitable since it is considered a proactive defense instead of a reactive one, being the first a technique able to generalize better, accounting for a wider variety of attack scenarios to suppress. Nonetheless it will be important to remark that this type of defense mechanism often entails a more sophisticated process than detection defense mechanisms. Albeit a mitigation defense mechanism may not exclude poisoned samples as part of the training set, it certainly involves a more sophisticated development process. This is because viable solutions for mitigation sometimes are more fundamented into probability of accounting for mislabeling samples than relying on the training labels alone. An example of this is seen in [70] wherein multi-objective optimization have represented a solution to this problem for SVM classifiers.

**B. SECOND APPROACH: DATA SANITIZATION**

Similarly, to the previous idea. Another approach could seek detecting malicious data and counteracting the effects of the same before getting the sample points to the training set, such approach commonly receives the name of Label sanitization (LS) as seen in the work in the defense scheme proposed in [35]. In this paper, label sanitization (LS) bases its defense on the decision boundary of SVM, observing the remoteness of the poisoned samples, commending these samples to be re-labelled.

Another way to detect outliers beforehand is seen in [58]. Notice that in our objectives detection and mitigation actions appear as the top priority. Data sanitization is meant to become a third type of defense to be used against an attacker that tries to tamper the training set, thereby, we would be more focus on the quality of the data before feeding it to the ML model of our choice.

**C. THIRD APPROACH: DEFENSES ACTING DURING INFERENCE PHASE**

Most of the works in literature do not consider defense systems acting during the inference phase once accounting for a solid defense active throughout the training phase. Possibly because such approach might not seem necessary since the hyperparameters of the model have been already set during training, guaranteeing that the integrity of the model is not compromised. For this reason, a defense approach acting during inference phase is not considered as a main point of interest in this present work.

Once completed the training phase, the proposed defense mechanism should endow the target model with the necessary level of robustness to counteract the effects of poisoning data. Thereby the model should naturally reflect an improvement in performance during the inference phase, thus reducing considerably any potential drop in accuracy to a minimum if not negligible percent.

**V. CONCLUSIONS**

It is not new that ML model can be attacked by compromising the data needed in the training phase, jeopardizing entirely the decision-making process of classifiers. Several malware detection engines entailing ML applications have been developed in the past, this have proven not to be exempt of presenting vulnerabilities to be exploded by attackers. Attacks by label flipping with malicious intent until this day represent an important point of focus among researchers in the area. Moreover, there is an urgent need of a defense system to battle model-agnostic attacks, such as label-flipping. In contract, several works in the literature contemplate combating one type of attack in specific targeting one AI algorithm time of.
We have successfully proposed, developed and assessed two variations of the label flipping attack, this attacks have proven to be suited and tuned for a particular application based on a mobile ex-filtration data framework. Both attacks have demonstrated their capability to drastically reduce the overall accuracy and missclassification rate in one or both classes of different binary classifiers. Indeed this paper promises to focus a thesis research work into counteracting the effects of training data poisoning attacks, such as label flipping, on several types of machine learning models.

APPENDIX A ML CLASSIFIER SCRIPTS

Python code of the ML classifiers can be found in the following GitHub repository: https://github.com/MiguelRamirezAguilar/PALM

The aim and main contribution of this research centers around enhancing robustness of ML models against tampered training data to be used during re-training, continuous training of ML models in malware detection application is fundamental with many concerns regarding the reliability of this special types of binary classifiers. Then, once having understood the complexity of the proposed attack and its effects on the target model, we will be capable of detailing a more sophisticated algorithm based on the concept of described in this work; either as a defense technique that detects and later rejects the poisoning samples present in the training set or a more complex approach that entails a defense mechanism that can mitigate a drop in accuracy caused by training the model with poisoning data.
APPENDIX B MOBILE EXFILTRATION DATASET

The mobile exfiltration dataset employed in the training of the models covered in this work can be found in the following GitHub repository:

https://github.com/MiguelRamirezAguilar/PALM

REFERENCES

[1] B. Biggio, I. Corona and et al., “Evasion attacks against machine learning at test time,” in Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases, 2013, pp. 37-42.

[2] A. Paudice, L. Muñoz-González and et al., “Detection of adversarial training examples in poisoning attacks through anomaly detection,” ArXiv.org, 2018. [Online] Available: https://arxiv.org/abs/1802.03041.

[3] Z. Hu, B. Tan and et al., “Learning Data Manipulation for Augmentation and Weighing,” ArXiv, 2019. [Online] Available: https://arxiv.org/abs/1910.12795.

[4] M. Comter, “Attacking Artificial Intelligence: AI’s Security Vulnerability and What Policymakers Can Do About It,” Belfer Center for Science and International Affairs, Harvard Kennedy School. Cambridge, MA, USA, Aug. 2019. [Online] Available: https://www.belfercenter.org/publication/attackingai.

[5] D. Miller, Z. Xiang and et al., “Adversarial Learning Targeting Deep Neural Network Classification: A Comprehensive Review of Defenses Against Attacks,” Proceedings of the IEEE, vol. 108, no. 3, pp. 402-433, 2020.

[6] X. Ma, B. Li and et al., “Characterizing adversarial subspaces using local intrinsic dimensionality,” ArXiv.org, 2018. [Online] Available: https://arxiv.org/abs/1801.02613.

[7] Y. Ma, J. Xie and et al., “Explaining Vulnerabilities to Adversarial Machine Learning through Visual Analytics,” IEEE Transactions on Visualization and Computer Graphics, vol. 26, no. 1, pp. 1075-1085, 2020.

[8] L. O. Nweke, “Using the CIA and AAA Models to Explain Cybersecurity Threats,” KNIME, 2021. [Online] Available: https://github.com/zalandoresearch/fashion-mnist.

[9] W. Xu, Y. Qi and et al., “Automatically evading classifiers,” in Proc. Netw. Distrib. Syst. Symp., 2016, pp. 1-5.

[10] J. Hayes, L. Melis and et al., “LOGAN: Membership inference attacks against generative models,” Proc. Privacy Enhancing Technol., vol. 2019, no. 1, pp. 133-152, 2019.

[11] A. Salem, Y. Zhang and et al., “ML-leaks: Model and data independent membership inference attacks and defenses on machine learning models,” ArXiv.org, 2018. [Online] Available: https://arxiv.org/abs/1806.01246.

[12] R. Shokri, M. Stronati and et al., “Membership inference attacks against machine learning models,” in Proc. IEEE Symp. Secur. Privacy (SP), 2017, pp. 3-18.

[13] L. Xiao and C. Eckert, “Adversarial label flips attack on support vector machines,” in 20th European Conference on Artificial Intelligence (ECAI), Montpellier, France, 2012, pp. 870-875.

[14] D. Miller, X. Hu and et al., “Adversarial learning: A critical review and academic study,” in Proc. IEEE 17th Int. Workshop Mach. Learn. Signal Process. (MLSP), 2017, pp. 1-6.

[15] B. Biggio, B. Nelson and et al., “Attack on Deep Neural Network Applications,” IEEE Access, vol. 7, pp. 691-706.

[16] C. Yang, Q. Wu and et al., “Generative poisoning attack method against neural networks,” ArXiv.org, 2017. [Online] Available: https://arxiv.org/abs/1703.01340.

[17] "FMNIST," 2019. [Online] Available: http://yann.lecun.com/exdb/mnist/.

[18] "CIFAR-10," 2019. [Online] Available: http://www.cs.toronto.edu/~kriz/cifar.html.

[19] "CIFAR-100," 2019. [Online] Available: http://www.cs.toronto.edu/~kriz/cifar100.html.

[20] L. Munoz-Gonzalez, B. Pfitzner and et al., “Poisoning attacks with generative adversarial nets,” ArXiv.org, 2019. [Online] Available: http://arxiv.org/abs/1906.07773.

[21] "MNIST," 2017. [Online] Available: https://github.com/zalandoresearch/fashion-mnist.

[22] "FEDIST," 2018. [Online] Available: https://github.com/zalandoresearch/fedist.

[23] "DIP," 2019. [Online] Available: https://github.com/DeepInspection.

[24] "CIFAR-10," 2019. [Online] Available: http://www.cs.toronto.edu/~kriz/cifar.html.

[25] "ADV," 2019. [Online] Available: http://www.cs.toronto.edu/~kriz/adv.html.

[26] "MNIST," 2019. [Online] Available: http://yann.lecun.com/exdb/mnist/.

[27] "CIFAR-10," 2019. [Online] Available: http://www.cs.toronto.edu/~kriz/cifar.html.

[28] "CIFAR-100," 2019. [Online] Available: http://www.cs.toronto.edu/~kriz/cifar100.html.

[29] "FMNIST," 2019. [Online] Available: http://yann.lecun.com/exdb/mnist/.

[30] "DIP," 2019. [Online] Available: https://github.com/DeepInspection.

[31] "ADV," 2019. [Online] Available: http://www.cs.toronto.edu/~kriz/adv.html.

[32] "CIFAR-10," 2019. [Online] Available: http://www.cs.toronto.edu/~kriz/cifar.html.

[33] "CIFAR-100," 2019. [Online] Available: http://www.cs.toronto.edu/~kriz/cifar100.html.

[34] "FMNIST," 2017. [Online] Available: https://github.com/zalandoresearch/fashion-mnist.

[35] "DIP," 2018. [Online] Available: https://github.com/DeepInspection.

[36] "ADV," 2018. [Online] Available: http://www.cs.toronto.edu/~kriz/adv.html.

[37] "CIFAR-10," 2018. [Online] Available: http://www.cs.toronto.edu/~kriz/cifar.html.

[38] "CIFAR-100," 2018. [Online] Available: http://www.cs.toronto.edu/~kriz/cifar100.html.

[39] "FMNIST," 2017. [Online] Available: https://github.com/zalandoresearch/fashion-mnist.

[40] "DIP," 2016. [Online] Available: https://github.com/DeepInspection.

[41] "ADV," 2016. [Online] Available: http://www.cs.toronto.edu/~kriz/adv.html.

[42] "CIFAR-10," 2016. [Online] Available: http://www.cs.toronto.edu/~kriz/cifar.html.

[43] "CIFAR-100," 2016. [Online] Available: http://www.cs.toronto.edu/~kriz/cifar100.html.

[44] "FMNIST," 2015. [Online] Available: https://github.com/zalandoresearch/fashion-mnist.

[45] "DIP," 2014. [Online] Available: https://github.com/DeepInspection.

[46] "ADV," 2014. [Online] Available: http://www.cs.toronto.edu/~kriz/adv.html.

[47] "CIFAR-10," 2014. [Online] Available: http://www.cs.toronto.edu/~kriz/cifar.html.

[48] "CIFAR-100," 2014. [Online] Available: http://www.cs.toronto.edu/~kriz/cifar100.html.
[50] M. Li, Y. Sun and et al., “Deep Reinforcement Learning for Partially Ob-
served Data Poisoning Attack in Crowdsensing Systems,” IEEE Internet of
Things Journal, vol. 7, no. 7, pp. 6266-6278, 2020.
[51] P. Zhao, H. Jiang and et al., “Garbage In, Garbage Out: Poisoning Attacks
Disguised with Plausible Mobility in Data Aggregation,” IEEE Transac-
tions on Network Science and Engineering, vol. 8, no. 3, pp. 2679-2693,
2021.
[52] V. Bindschaedler and R. Shokri, “Synthesizing plausible privacy-
preserving location traces,” in Proc. IEEE Symp. Secur. Privacy, 2016, pp.
565-566.
[53] B. I. P. Rubinstein, B. Nelson and et al., “ANTIDOTE: Understanding
and defending against poisoning of anomaly detectors,” in Proc. 9th ACM
SIGCOMM Conf. Internet Mes. Conf. (IMC), 2009, pp. 1–14.
[54] P. K. Koh, J. Steinhardt and et al., “Stronger data poisoning attacks
break data sanitization defenses,” ArXiv.org, 2018. [Online] Available:
https://arxiv.org/abs/1811.00741.

[55] J. Steinhardt, P. W. Koh and et al., “Certified defenses for data poisoning
attacks,” in Proc. NIPS, 2017, pp. 3520–3532.
[56] T. Gu, B. Dolan-Gavitt and et al., “BadNets: Identifying vulnerabilities
in the machine learning model supply chain,” ArXiv.org, 2017. [Online]
Available: https://arxiv.org/abs/1708.06733.
[57] A. N. Bhagoji, S. Chakraborty and et al., “Analyzing federated learning
through an adversarial lens,” in Proc. 56th Int. Conf. Mach. Learn., 2019,
p. 654–663.
[58] N. Baracaldo, B. Chen and et al., “Mitigating poisoning attacks on ma-
chine learning models: A data provenance based approach,” in Proceed-
ings of the 10th ACM Workshop on Artificial Intelligence and Security,
AISe$@$ICS, Dallas, TX, USA, 2017, pp. 103–110.
[59] L. Zhao, S. Hu and et al., “Shielding Collaborative Learning: Mitigating
Poisoning Attacks Through Client-Side Detection,” IEEE Transactions
on Dependable and Secure Computing, vol. 18, no. 5, pp. 2029-2041, 2021.

[60] “KDDCup,” 1999. [Online] Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
[61] G. Xu, H. Li and et al., “VerifyNet: Secure and verifiable federated
learning,” IEEE Trans. Inf. Forensics Security, vol. 15, pp. 911–926, 2020.
[62] J. Benesty, J. Chen and et al., “Pearson correlation coefficient,” in Noise
Reduction in Speech Processing, Springer Topics in Signal Processing, vol.
2, Berlin, Germany: Springer, 2009, pp. 1–4.
[63] P. Blanchard, Rachid Guerraoui and et al., “Machine learning with adver-
saries: Byzantine tolerant gradient descent,” in Proc. NeurIPS, 2017, pp.
119–129.
[64] E. M. E. Mhamd, R. Guerraoui and et al., “The Hidden Vulnerability
of Distributed Learning in Byzantium,” in Proc. ICML, 2018, pp. 3521–3530.
[65] S. Weerasinghe, T. Alpcan and et al., “Defending Support Vector Machines
Against Data Poisoning Attacks,” IEEE Transactions on Information Forensics
and Security, vol. 16, pp. 2566-2578, 2021.
[66] B. Biggio, I. Corona and et al., “Bagging classifiers for fighting poisoning
attacks in adversarial classification tasks,” in Proc. 10th Int. Conf. Mult.
Reliability and Security, 2011, pp. 350–359.
[67] H. Xiao, B. Biggio and et al., “Support vector machines under adversarial
label contamination,” Neurocomputing, vol. 160, pp. 53-62, 2015.
[68] S. Chen, M. Xue and et al., “Automated poisoning attacks and defenses
in malware detection systems: An adversarial machine learning approach,”
Comput. Secur., vol. 73, pp. 326–344, 2018.
[69] J. Chen, X. Zhang and et al., “De-Pois: An Attack-Agnostic Defense
against Data Poisoning Attacks,” IEEE Transactions on Information Forensics
and Security, vol. 16, pp. 3412-3425, 2021.
[70] M. Mirza and S. Osindero, “Conditional generative adversarial nets,”
ArXiv.org, 2014. [Online] Available: http://arxiv.org/abs/1411.1784.
[71] F. Gulrajani, M. A. Ahmed and et al., “Improved training of Wasserstein
GANs,” in Proc. NIPS, 2017, pp. 5767-5777.
[72] X. Zhang, X. Zhu and et al., “Training set debugging using trusted items,” in
Proc. AAAI, 2018, pp. 1–8.
[73] M. Jagielski, A. Oprea and et al., “Manipulating machine learning: Poison-
ing attacks and countermeasures for regression learning,” in Proc. IEEE
Symp. Secur. Privacy (SP), 2018, pp. 19-35.
[74] X. Chen, C. Liu and et al., “Targeted backdoor attacks on deep learn-
ing systems using data poisoning,” ArXiv.org, 2017. [Online] Available:
https://arxiv.org/abs/1712.05526.
[75] Ł. Dąkótko, G. Kamath and et al., “Sever: A robust meta-algorithm for
stochastic optimization,” in Proc. ICML, 2019, pp. 1596–1606.
[76] N. Peri, N. Gupta and et al., “Deep k-NN defense against clean-
label data poisoning attacks,” ArXiv.org, 2019. [Online] Available:
https://arxiv.org/abs/1909.13374.
[104] R. Zhang and Q. Zhu, “A game-theoretic defense against data poisoning attacks in distributed support vector machines,” in *Proc. IEEE 56th Annu. Conf. Decis. Control (CDC)*, 2017, pp. 4582–4587.

[105] M. A. Ramírez, S.-K Kim, S. Yoon, E. Damiani, H. A. Hamadi, C. Agostino, N. Bean, Y.-J Byon, T.-Y Kim, C.-S Cho and C. Y. Yeun, “Poisoning Attacks and Defenses on Artificial Intelligence: A Survey,” in IEEE Access, vol. XX, pp. XXXXX-XXXXX, 2022.

[106] N. Capuano, G. Fenza, V. Loia and C. Stanzione, “Explainable Artificial Intelligence in CyberSecurity: A Survey,” in IEEE Access, vol. 10, pp. 93104-93139, 2022, doi: 10.1109/ACCESS.2022.3204051.

[107] Z. Zhang, H. A. Hamadi, E. Damiani, C. Y. Yeun and F. Taher, "Explainable Artificial Intelligence Applications in Cyber Security: State-of-the-Art in Research," in IEEE Access, vol. 10, pp. 93104-93139, 2022, doi: 10.1109/ACCESS.2022.3204051.

[108] M. Morcos, H. Al Hamadi, E. Damiani, S. Nandyala and B. McGillion, "A Surrogate-Based Technique for Android Malware Detectors' Explainability", 2022 IEEE 18th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2022, pp. 112-117.

MIGUEL ANGEL RAMIREZ AGUILAR

Born in Mexico City, Mexico, 1994. Received his Bachelor’s Degree in Mechatronics Engineering from Universidad Nacional Autonoma de Mexico (UNAM), Mexico City, Mexico, in 2018, and is currently a MSc. student in Electrical and Computer Engineering specializing in Artificial Intelligence at Khalifa University, Abu Dhabi, United Arab Emirates. Currently he is a Graduate Researcher at Khalifa University, United Arab Emirates; previously worked as a Computer-Aided Design engineer at Ford Motor Company, Mexico, R&D Intern at SuitX “U.S. Robotics”, CA, USA, and as research assistant in robotics at Universidad Nacional Autonoma de Mexico, Mexico. Published his thesis “Rediseño de dedo protésico,” Ptolomeo, 2020. [Online] Available: http://132.248.52.100:8080/xmlui/handle/132.248.52.100/17328 with main emphasis in robotics, optimization and prosthetics. His current research interests are within the scope of Machine Learning oriented to Cybersecurity, Deep Learning and optimization algorithms. In addition, previous research experience related to the fields of control design, robotics and virtual instrumentation. Mr. Miguel Angel Ramirez Aguilar, IEEE non-member.