Algorithms for Detecting and Preventing Attacks on Machine Learning Models in Cyber-Security Problems

A P Chukhnov¹, Y S Ivanov¹,²
¹AV Soft, st. Gostinichnaya h.5, Moscow, 127106, Russia
²Komsomolsk-na-Amure state university, Lenin Avenue 27, Komsomolsk-on-Amur, 681013, Russia

E-mail: ivanov_ys@icloud.com

Abstract. Machine learning algorithms can be vulnerable to many forms of attacks aimed at leading the machine learning systems to make deliberate errors. The article provides an overview of attack technologies on the models and training datasets for the purpose of destructive (poisoning) effect. Experiments have been carried out to implement the existing attacks on various models. A comparative analysis of cyber-resistance of various models, most frequently used in operating systems, to destructive information actions has been prepared. The stability of various models most often used in applied problems to destructive information influences is investigated. The stability of the models is shown in case of poisoning up to 50% of the training data.

1. Introduction
Machine learning is used in almost all application domains: driving, controlling production processes, searching for malicious files, etc. [1, 2, 3, 4]. Obviously, a machine learning system should be practically “perfect” (or close to such) in its predictions.

Malware varieties continue to increase either in quantity, type, functionality or target of the attack. Traditionally, antivirus solutions have relied on signature and heuristic methods, but they require malware to be analyzed before rules and heuristics are defined. Taking into account the large data amount, constantly growing types of attacks, as well as the use of malicious code obfuscation, traditional approaches (including manual ones) do not cope with the task.

Machine learning (ML) methods can solve this problem as well as cyber-security ones: detect network threats and attacks, prevent data leaks, encrypt personal data, detect malicious files and phishing [5], etc.

Machine learning algorithms in cyber-security are nominally divided into several categories:
- Linear, regression models, statistical models: Random Forest, SVM;
- Clustering algorithms;
- Boosting: Gradient boosting, etc.
- Neural network models, including those using deep learning.

It should be noted that malware detection problem involves the problem of “concept drift”, which consists in changing the data with time, which leads to a change in the statistical dependencies between the feature vector, the goal variable and the “model degradation”. This effect is due to the emergence of new types of viruses, new libraries, OS versions, etc. Thus, it is important to identify when the model is...
showing signs of degradation and to retrain it. Existing solutions [6] aim at periodic retraining, which can be done completely, partially, or gradually (incrementally) without “forgetting” the past data.

With advances in ML methods and algorithms, a new class of attacks, attacks on the machine learning models, has appeared. Machine learning algorithms can be vulnerable to many forms of attacks aimed at leading the machine learning systems to make deliberate errors. Attackers can corrupt the dataset used for additional model training or automatically generate many attacking patterns until a weak point of the model is found.

The following possible attack strategies are defined:
- **Evasion**, in which object instances and their features are deliberately modified or hidden so as not to be classified as a threat. An example of evasion can be a presentation of a spam message not in the form of text, but in the form of an image attached to the message. Deeper methods involve adjustment to the model and changing features to deceive it, for example, in computer vision systems [7].
- **Poisoning Attack** presupposes the poisoning of training data. The result of algorithm training strongly depends on the initial data on the basis of which the training is carried out.
- **Model theft (model extraction)** involves a black box exploration of the model by an attacker in order to (approximately) restore the model and use it for their own commercial purposes.

In paper [8], the authors provide a list and analysis of the published attacks on various machine learning models and their categorization according to their phase (training, additional training) and their impact on security purposes.

Different researchers are working in the field of detecting and/or preventing attacks on ML models. To protect against adversarial attacks, the authors of [9] propose a two-stage method with pre-coding of marks and using a model for an adversarial attack during the training of the own deep model. This approach made it possible to achieve 96% accuracy.

The authors from the University of Michigan [10] carried out a fairly in-depth review of existing adversarial attack techniques and methods of countering them.

To protect against data corruption, it is proposed to use anomaly detection algorithms based on autoencoders [11], which will be able to identify the input data that is complex and strange, or not similar to other data.

Considering that malware detection problem includes a “concept drift”, it is necessary to constantly retrain the model or give it an additional training. The result of algorithm training strongly depends on the initial data on the basis of which the training is carried out. The data can turn out to be bad, corrupted — this can happen either by accident or out of malice, in case of poisoning attacks. An attacker can constantly generate malicious files, very similar to some clean one, and send them to a computer virus analysis laboratory. The border between clean and malicious files will gradually blur, in consequence of which the model will “degrade”.

Thus, the problem of monitoring/inspecting the models and data used in the tasks of protecting against the cyber-attacks becomes extremely relevant.

This paper examines the cyber-resistance of the most commonly used machine learning models for recognizing phishing links. Recommendations on improving their reliability are proposed.

2. **Statement of machine learning and data poisoning problems**

Regardless of the domain, in general, the supervised learning algorithm of the ML model always involves the following steps (Figure 1).

Let there be: an image set \( \omega \in \Omega \) prescribed by the features \( x_i, \ i = 1, n \), the complex of which for image \( \omega \) is represented by vector descriptions \( \Phi(\omega) = (x_1(\omega), x_2(\omega), \ldots, x_n(\omega)) = x \); a set of classes \( \mathbb{B} = \{ \beta_1, \ldots, \beta_k, \ldots, \beta_c \} \), \( c \) – a number of classes.
A priori information is represented by a training set (dataset) \( \mathbb{D}_{\text{train}} = \{ (x_j, \beta_j) \}, j = 1, L \) given by a table, each line of which \( j \) contains a vector image description \( \Phi(\omega) \) and a class mark \( \beta_j \), \( k = 1, c \). Let us note that the training set characterizes the unknown mapping \( \mathbf{F}: \Omega \rightarrow \mathbb{B} \).

It is necessary to classify each image \( \omega \) represented by its feature assessment \( \tilde{x} \) using a mapping \( \mathbf{F}: \tilde{x} \rightarrow \beta_k, k = 1, c \) in accordance with a given criterion \( P(\tilde{x}) \) that minimizes the probability of classification error.

For the cyber-security problems, we consider a binary classification:

\[
\mathbf{F}: \tilde{x} \rightarrow \{0, 1\}
\]  

(1)

Mapping \( \mathbf{F} \) is evaluated by its test error on some fixed test set \( \mathbb{D}_{\text{test}} \), which is a part of samples from the total set \( \mathbb{D} \). Loss function \( L(\mathbf{F}, \tilde{x}) \) characterizing the value of the algorithm error on the object \( \tilde{x} \) is used for assessment. Then algorithm quality functional \( \mathbf{F} \) on sampling \( \mathbb{D}_{\text{test}} \) can be represented as:

\[
Q(\mathbf{F}, \mathbb{D}_{\text{test}}) = \frac{1}{l} \sum_{i=1}^{l} L(\mathbf{F}, \tilde{x}_i),
\]  

(2)

where \( l \) is a number of test sample examples.

The classic training method consists in finding such hyper parameters and weight \( f \) of the algorithm \( \mathbf{F} \) at which the algorithm error is minimized \( \arg \min_{f, \mathbf{W}} Q(\mathbf{F}, \mathbb{D}_{\text{test}}) \).

If necessary, the model can be additionally trained (using online or incremental training methods) or fully trained (with a complete change in weight and the loss of old information) on new data.

Taking into account the “concept drift”, in many problems, there is a need for constant additional training and retraining of algorithms using new data.

During online or incremental training, it is necessary to monitor the model quality and training data in order to prevent poisoning attacks.

The aim of poisoning attack is to change the training procedure so that the resulting classifier with backdoor \( \mathbf{F}^b \) differs from the purely trained classifier \( \mathbf{F} \) as follows:

\[
\mathbf{F}^b(\tilde{x}) = \mathbf{F}(\tilde{x}), \quad \mathbf{F}(\tilde{x}^b) = \beta_k, \quad \mathbf{F}^b(\tilde{x}^b) = \beta_k \neq \beta_k,
\]  

(3)

where \( \tilde{x}^b = (x^b_1, x^b_2, \ldots, x^b_n) \) is a poisoned image, \( \beta_h, \beta_k \) - algorithm assessments. Moreover, in the task of cyber-security, an attacker is interested in constant \( \mathbf{F}^b(\tilde{x}^b) = 0 \).

For example, intrusion detection systems (IDS) can be additionally trained on their own data or on the data from open sources. An attacker can poison this data by introducing malicious samples into a positive class, which will subsequently lead to the system vulnerability [12].
To implement the attack, it is necessary to add a poisoning set $\mathcal{D}_{\text{poisoned}} = \{(x^b_j, \beta_j)\}, j = 1, N$ consisting of $N$ poisoned samples in the set $\mathcal{D}_{\text{train}}$.

At the same time, to hide the fact of attack, an attacker minimizes the number of changeable features $x^b$ from each $\bar{x}^b$.

There are two types of poisoning attacks:
1. **Accessibility attacks** aimed at introducing so much bad data into a model that any prediction generated by the model becomes useless.
2. **Backdoor attacks** aimed at introducing a certain feature into the model that affects the result.

Bearing in mind the severity of this threat, it is necessary to identify the bottlenecks in the process of ML models training, as well as formulate the recommendations on protecting against this type of attacks.

The works study poisoning attacks for malware clustering [13], malware detecting [14], detecting DOS attacks [15], intrusion detecting [16], etc.

To detect a poisoning attack, the authors of [17] propose to use an ensemble of two classifiers, one of which is aimed at detecting “suspicious” data submitted for training to the main classifier.

The main method of protection against poisoning is outliers [18] and anomalies detection in the training set. Some researchers [19] suggest analyzing the responses of the SHAP interpreter to identify the suspicious data.

Another option for anomalies detection [20] is the use of micro-models trained on non-overlapping epochs of the training sample. Using the majority of micro-model votes, training instances are indicated as safe or suspicious.

Another common type of protection [21] is to analyze the impact of recently added training samples on the model accuracy. The model accuracy is verified on a specially isolated testing set, and when the accuracy drops, training is not accepted.

In exceptional cases, manual analysis of abnormal samples by a security analyst is used [22].

The selection of features in the development of a machine learning model is a mandatory procedure both in the preparatory phase (prior to the training) and at the stage of assessing the obtained results and subsequent adjustment of the training sample and/or the model hyper parameters. The work [22] proposes to analyze SHAP values to identify the features that are potentially vulnerable to the backdoor attacks.

3. Implementation of a poisoning attack on models in the problem of cyber-security

3.1. Description of the dataset and experiment
As an experimental problem, the task of phishing sites identification will be considered, since a phishing attack currently poses a great threat to the daily life of people and the Internet environment. Machine learning algorithms are one of the reliable methods for phishing websites identification. As attackers think up new ways to deceive users, machine learning models detecting phishing attacks should be continuously improved and additionally trained to reduce the negative effect of the “concept drift”.

We used our own balanced dataset developed in AV Soft, containing the features extracted from over 60,000 phishing and benign sites. In this set, each web page is described by more than 800 features extracted from the URL, page code, content, external sources and other categories.

Several algorithms have been trained to solve phishing site detection problems:
- Catboost
- SVM
- RandomForest.

The choice of algorithms is due to the features of the data set, as well as the requirements for computing resources during the training and additional training.

To implement the attack, the dataset was divided into training and testing parts. For each part, some poisoning samples were generated by means of Art and SecML frameworks.
We carried out 2 types of poisonous attacks.

Accessibility attack. For the first type, label flip poisonous samples were used. When reversing the label, we were guided by the assumption that an attacker would not be able to change the label of a malicious sample to a safe one, but he could purposefully add safe sites to the open databases of phishing sites.

The training dataset consisted of three parts: $\mathbb{D}_{\text{train}}$ consists of 39035 clean samples. $\mathbb{D}_{\text{add}}$ contains 13012 samples divided into clean and poisoned:

$$\mathbb{D}_{\text{add}} = \mathbb{D}_{\text{clean}} \cup \mathbb{D}_{\text{poisoned}}^* \quad (4)$$

With full training, the model is trained on $\mathbb{D}_{\text{train}}$ and $\mathbb{D}_{\text{add}}$ at once. With incremental training, the model is first trained on $\mathbb{D}_{\text{train}}$, after which additional training on $\mathbb{D}_{\text{add}}$ is performed.

The experimental results are shown in Table 1. As a result, the accuracy and efficiency of the model decreases as the poisoned samples are added to the additional training set. However, the algorithm is quite robust against reversal of the label upon complete retraining. However, taking into account the concept drift in information security tasks, constant additional training is required.

**Table 1. Attack with flip labels.**

| Models         | Type          | Type test | Percentage of infection | Percentage of infection | Percentage of infection | Percentage of infection |
|----------------|---------------|-----------|-------------------------|-------------------------|-------------------------|-------------------------|
|                |               |           | 0% off additional set (39034+15012+0) | 10% off additional set (39034+12565+647) | 30% off additional set (39034+11072+1940) | 50% off additional set (39034+9779+3233) |
|                |               |           | Acc. | Prec. | Rec. | F1 | Acc. | Prec. | Rec. | F1 | Acc. | Prec. | Rec. | F1 | Acc. | Prec. | Rec. | F1 |
| Catboost       | Inc.          | clean test | 0.9696 | 0.9697 | 0.97 | 0.9696 | 0.9697 | 0.9707 | 0.9966 | 0.94 | 0.94 | 0.946 | 0.94 | 0.76 | 0.76 | 0.837 | 0.746 |
|                | Full          | clean test | 0.9705 | 0.9705 | 0.971 | 0.9705 | 0.97 | 0.97 | 0.97 | 0.969 | 0.968 | 0.9685 | 0.969 | 0.9691 | 0.9691 |
| Random Forest  | Full          | clean test | 0.971 | 0.9711 | 0.971 | 0.971 | 0.97 | 0.969 | 0.97 | 0.969 | 0.967 | 0.9676 | 0.9669 | 0.9615 | 0.9615 | 0.9614 |
| SVM            | Full          | clean test | 0.86 | 0.867 | 0.891 | 0.854 | 0.814 | 0.823 | 0.84 | 0.841 | 0.7754 | 0.764 | 0.813 | 0.715 | 0.6278 | 0.6268 | 0.7853 | 0.5673 |

Integrity attack. Most interesting is the second type of poison attack, which was a backdoor attack. As a backdoor, we used features that are potentially available to an attacker: manipulation of the page url and content.

Only 10 features were used to install the backdoor out of 800 features used in the dataset. Backdoor injection in applied information security problems is most likely during additional training of the model; therefore, the Catboost model was considered as a model for experiments, since it supports incremental training.

Thus, the training set $\mathbb{D}_{\text{train}}$ consists of 39035 clean samples. $\mathbb{D}_{\text{clean}}$ and $\mathbb{D}_{\text{poisoned}}$ contains 19517 additional training clean samples and samples with backdoor (BD), $\mathbb{D}_{\text{poisoned}}^*$.

The $\mathbb{D}_{\text{add}}$ consists of clean and poisoned samples ($\mathbb{D}_{\text{poisonedPART}}$ and $\mathbb{D}_{\text{cleanPART}}$) taken from $\mathbb{D}_{\text{clean}}$ and $\mathbb{D}_{\text{poisoned}}$ in accordance with the proportion of poisoning:

$$\mathbb{D}_{\text{add}} = \mathbb{D}_{\text{cleanPART}} \cup \mathbb{D}_{\text{poisonedPART}}^* \quad (5)$$

We also have a clean set for controlling the $\mathbb{D}_{\text{control}}$ in size of 9185 examples and a poisoned control set for measuring the attack and defense efficiency $\mathbb{D}_{\text{control-poison}}$, with size of 9185 examples.

Two types of training, full and incremental, were considered. Each of the models was trained on $\mathbb{D}_{\text{clean}}$ and tested on $\mathbb{D}_{\text{control}}$.

Further, for the models supporting the incremental training, poisoned data from $\mathbb{D}_{\text{poisoned}}$ was successively added.
Also, for all models, a complete retraining was performed using the same part of the poisoned data mixed with $D_{\text{clean}}$.

Assessment in both types of training was carried out on a clean control dataset $D_{\text{control}}$ and on the poisoned control dataset $D_{\text{control}_\text{poison}}$.

Figure 2 shows how important features are shifted at 50% training data contamination. The values of the importance of the signs of the poisoned model are shown in orange, and the original one - in blue. Actually, the main purpose of the backdoor is to make the model react to certain features.

![Figure 2. Shifting the importance of features in case of poisoning.](image)

### 3.2. Results and discussion

Table 2 shows the algorithms used, the percentage of poisoned samples out of the total training dataset during the training and additional training.

When interpreting the results, we will be guided by the fact that attack integrity can be efficient if two conditions are met:

1. Metrics on $D_{\text{control}}$ fall slightly, otherwise the attack will be detected.
2. Metrics on $D_{\text{control}_\text{poison}}$ deteriorate significantly, which indicates the infection effectiveness.

As can be seen from the experiment, the accuracy of the Catboost model decreases by 30% even with a backdoor of 1% of the entire feature vector.

![Table 2. Backdoor attack](image)

Notable, it should be noted that the models have a significantly higher cyber-resistance on full retrain. The success of data poisoning attack shows the real threat related to these attacks for machine learning models.

It is worth noting that in this work, when generating poisoning examples, the categories of features subjected to disturbance were not taken into account. The most common type of protection is the outlier detection, also known as “data cleansing” and “anomaly detection”.
The second most common type of defense is to analyze the impact of recently added training samples on the model accuracy.

Thus, the following methods and recommendations for protection of data against poisoning can be distinguished:

- The use of classifier ensembles, consisting of different types of models and/or different subsets of features, will make the system more resistant to poisoning attacks.
- The use of a well-interpreted model (or an ensemble) to check (validate) new data for additional training with a “hard” operation threshold.
- Daily model and data telemetry: verification of data integrity, data drift check, anomaly detectors, metrics recalculation on control samples.
- Differential impact analysis. Initially it was used to check the fact of discrimination against certain groups of the population in the scoring system. It can also be successfully applied to the feature analysis and model interpretation in cyber-security. The Aequitas, Themis and AIF360 frameworks are used for research.
- Fair or private-models. Initially used for protection against attacks aimed at impersonization (extraction of personal data from the dataset), however, they also help to fight poisoning attacks through the use of the PATE private classifiers ensemble (private aggregation of teacher ensembles), each of which is trained on the isolated non-overlapping datasets with added random noise.
- Reject on Negative Impact (RONI). RONI is a method of removing the data lines reducing the prediction accuracy from a dataset.
- Residual analysis. The search of strange, noticeable patterns in your model's forecast residuals.
- Feature compression and transition to another feature space can be used to strengthen the models by combining different samples into a single one, as well as reducing the search space available to the attacker.

4. Conclusion

Experiments in the implementation of poisoning attacks on various models used in the application cyber-security tasks have been carried out.

The article presents and analyzes the results of studies on the resistance of diverse models to poisoning attacks.

The results and conclusions obtained in this study allow us to specify the sections of cyber-security associated with the use of datasets and models from unverified sources.

In conclusion, it should be noted that a promising area of further research is the development of algorithms aimed at detecting and countering poisoning attacks during the incremental learning.

5. References

[1] Amosov O S, Amosova S G, Ivanov Y S & Zhiganov S V 2020 Using the deep neural networks for normal and abnormal situation recognition in the automatic access monitoring and control system of vehicles Neural Computing and Applications 33(8) pp 3069–3083 https://doi.org/10.1007/s00521-020-05170-5

[2] Amosov O S, Ivanov Y S & Zhiganov S V 2017 Human Localization In Video Frames Using A Growing Neural Gas Algorithm And Fuzzy Inference Computer Optics 41(1) pp 46–58 https://doi.org/10.18287/2412-6179-2017-41-1-46-58

[3] Amosov O, Amosova S, Ivanov Y, Zhiganov S 2020 Using the deep neural networks for normal and abnormal situation recognition in the automatic access monitoring and control system of vehicles Neural Computing and Applications 33(8) 3069-3083 https://doi.org/10.1007/s00521-020-05170-5

[4] Pashchenko F, Amosov O, Amosova S, Ivanov Y, Zhiganov S 2019 Deep Neural Network Method of Recognizing the Critical Situations for Transport Systems by Video Images Procedia
Computer Science 151 pp 675-682 https://doi.org/10.1016/j.procs.2019.04.090

[5] Gupta B, Tewari A, Jain A, Agrawal D 2016 Fighting against phishing attacks: state of the art and future challenges Neural Computing and Applications 28(12) pp 3629-3654

[6] Gama J, Žliobaitė I, Bifet A, Pechenizkiy M, Bouchachia A 2014 A survey on concept drift adaptation ACM Computing Surveys 46(4) pp 1-37 https://doi.org/10.1145/2523813

[7] Ilyas A, Engstrom L, Athalye A, Lin J 2017 Query-Efficient Black-box Adversarial Examples (superceded) arXiv [cs.CV] Available from: http://arxiv.org/abs/1712.07113

[8] Auernhammer K, Kolagari R T, Zoppelt M 2021 Attacks on machine learning: Lurking danger for accountability Ceur-ws.org http://ceur-ws.org/Vol-2301/paper_2.pdf

[9] Wang D, Li C, Wen S, Nepal S, Xiang Y 2021 Defending against adversarial attack towards deep neural networks via collaborative multi-task training IEEE Trans Dependable Secure Comput pp 1–1 https://doi.org/10.1109/tdsc.2020.3014390

[10] Xu H, Li Y, Jin W, Tang J 2020 Adversarial attacks and defenses: Frontiers, advances and practice In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (New York, NY, USA: ACM) https://doi.org/10.1145/3394486.3406467

[11] Valkov V 2017 Credit card fraud detection using autoencoders in Keras — TensorFlow for hackers (part VII) Medium 2017 https://medium.com/@curiously/credit-card-fraud-detection-using-autoencoders-in-keras-tensorflow-for-hackers-part-vii-20e0c85301bd

[12] Müller N M, Kowatsch D, Böttinger K 2020 Data poisoning attacks on regression learning and corresponding defenses http://arxiv.org/abs/2009.07008

[13] Biggio B, Riek K, Ariu D, Wressnegger C, Corona I, Giacinto G et al 2018 Poisoning behavioral malware clustering http://arxiv.org/abs/1811.09985

[14] Xiao H, Biggio B, Brown G, Fumera G, Eckert C, Roli F 2018 Is feature selection secure against training data poisoning? http://arxiv.org/abs/1804.07933

[15] Rubinstein B IP, Nelson B, Huang L, Joseph A D, Lau S-H, Rao S, et al 2009 ANTIODE: Understanding and defending against poisoning of anomaly detectors In: Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference - IMC ’09 (New York, New York, USA: ACM Press) https://doi.org/10.1145/1644893.1644895

[16] Newsome J, Karp B, Song D 2006 Paragraph: Thwarting signature learning by training maliciously In: Lecture Notes in Computer Science (Berlin, Heidelberg: Springer Berlin Heidelberg) pp 81–105 https://doi.org/10.1007/11856214_5

[17] Chen S, Xue M, Fan L, Hao S, Xu L, Zhu H, et al 2017 Automated poisoning attacks and defenses in malware detection systems: An adversarial machine learning approach Comput Secur. 73 pp 326–44 https://doi.org/10.1016/j.cose.2017.11.007

[18] Paudice A, Muñoz-González L, Gyorgy A, Lupu E C 2018 Detection of adversarial training examples in poisoning attacks through anomaly detection http://arxiv.org/abs/1802.03041

[19] Severi G, Meyer J, Coull S, Oprea A 2020 Explanation-guided backdoor poisoning attacks against malware classifiers Available from: http://arxiv.org/abs/2003.01031

[20] Cretu G F, Stavrou A, Locasto M E, Stolfo S J, Keromytis A D 2008 Casting out demons: Sanitizing training data for anomaly sensors In: 2008 IEEE Symposium on Security and Privacy IEEE https://doi.org/10.1109/sp.2008.11

[21] Nelson B, Barreno M, Jack Chi F, Joseph A D, Rubinstein B IP, Saini U, et al 2009 Misleading learners: Co-opting your spam filter In: Machine Learning in Cyber Trust. Boston, MA: Springer US pp 17–51 https://doi.org/10.1007/978-0-387-88735-7_2

[22] Mei S, Zhu X 2015 Using machine teaching to identify optimal training-set attacks on machine learners In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence AAAI Press pp 2871–7