Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät
Institut für Informatik

Risk assessment in Machine Learning security - a framework for risk measurement

Masterthesis

for the attainment of the academic degree
Master of Science (M. Sc.)

submitted by: Jan Schröder
born on: 03.03.1996
born in: Lemgo

Surveyor: Martin Schneider
Prof. Dr. Holger Schlingloff

submitted on: .... defended on: ....

Contents

Abstract

This thesis is Open Source and can be found on `https://github.com/EvilWatermelon/Risk-Measurement-Framework` together with the Risk-Measurement-Framework.

Acknowledgements

The following table explains and declares terms that are used in this thesis for standartization. The order of the table is organized by declaring a term before its relation to other terms.

Term	Description
Machine Learning	Machine learning is a program that trains from training sets a predictive model. [3]
Machine Learning model	A model is a representation of a machine learning program that learned from the training sets. [3]
Threat	Security violation when an event could cause harm [28]
Attacker	An attacker is a person that tries to perform something unintentionally with or without access to the machine learning model [2]
Risk	Risk is the combination between the frequency of damage and the extent of damage. The damage is the difference between a planned and unplanned result [2].
Measurement property	In the context of risk measurement in this present thesis, properties should help to find features to measure risks in a machine learning model.
Measurement proposal	In the context of risk measurement in this present thesis, proposals from approaches of other scientific works are used. With the previously declared measurement properties together, this thesis uses their possibilities, so that the risk measurement leads to the best possible results.
Attacker's characteristic	Values and features of an attacker that are summarized into a generic term and fulfill properties to measure risks.

Term	Description
Attack characteristic	Raw data of a measured attack that can be summarized in a generic term. This characteristics can fulfill properties or mapped to find attacker's characteristics.
Metric	In context of machine learning, a metric is a value that is optimized in a machine learning program [3]. In context of this thesis, metrics will be used often for measuring risks.
Vulnerability	A vulnerability is an error that cause security relevant threats [2].

1 Introduction

Machine Learning (ML) is a constantly growing field and is essential for many innovative applications such as highly-automated and autonomous driving. Resulting from this, there is an increased need to maintain security. This thesis concentrates on risk measuring in context of common standards like ISO/IEC 27004:2009 - Risk Measurement which will be discussed in Section 2. Risk measurement is a part of risk assessment to analyze the system for vulnerabilities. This present thesis evaluates how to measure risks and what the extent of damage is by visualizing all results.

This thesis explains and discusses the design of a conceptual framework and its implementation to measure risks which is called Risk-Measurement-Framework (RMF). The RMF is designed by a conceptual framework based on risk indicators as a fundamental part upon approaches by Jakub Breier et al. [8] and Paul Schwerdtner et al. [26]. The core of the implementation of the RMF is the Adversarial-Robustness-Toolbox (ART) that is included as a Python library but also an open-source framework which will be explained in Section 2.

Sections 1 and 2 are intended to clarify the goals and expectations of this thesis, explain terms, show necessary prior knowledge so that it is well defined where this thesis should go. Section 3 is one of the main parts of the thesis. The section discusses and describes the conceptual framework and gives the basis for the technical framework explained in Section 4. Section 5 explains the case study that uses the framework and shows its potential and how to use it. In Section 6, a conclusion points out possible future work and summarizes the results.

1.1 Motivation

The classic IT security is a large field and essential for every software application. In ML, security is also essential and needs more tools to find vulnerabilites and measure risks for the subsequent defense implementation. This thesis evaluates a conceptual and technical framework in the context of IT security standards. The aim is to improve security in ML, which could help researchers and companies to optimize their work. Due to the research for this present thesis, there were a lot of scientific papers that evaluate IT security management in the context of ISO 27005 but less with ISO 27004. Therefore, there is a need to put more focus on ISO 27004. So ML in relation to ISO 27004 is another motivating factor to extend the research in the context of security for ML and ISO 27004. From the previously mentioned points, it should emerge that this thesis should show the possibility of using common standards for risk measurement in ML.

1.2 Goals and expectations of this present thesis

Expectations

The expectations for this thesis are implementing and evaluating the RMF for risk measurement of ML models. The focus here is on backdoor attacks and finding the attacker's effort. Furthermore, there is a need to show the extent of damage by implemententing different attacks. In order to meet these expectations, the following research questions and their descriptions should show what this thesis is aiming at.

Goals

RQ1: Which ISO 27004 measurement metrics are useful to measure the risks of poisoning attacks?

RQ2: How can the size of a dataset be used to measure the risks of poisoning attacks?

RQ3: What are risk indicators of poisoning attacks?

RQ4: Which risk indicators can be used for the ML model apart from the dataset?

RQ5: How can the effort of an attack be measured?

RQ6: Which measurement requirements of ISO 27004 can be used to measure the effort of an attack in ML security?

RQ7: Which risk indicators from the poisoning attacks and the attackers effort are useful to evaluate the risks with the RMF?

RQ8: What are possible methods in the RMF to measure the effort of an attacker?

RQ9: Which backdoor attacks must execute an attacker and objective properties must be fulfilled by the attacker to find how much damage an attacker wants to do with his attack?

The first research question RQ1 should introduce the discussion on how to bring the IT security standards in relation with security of ML. This is answered by explaining what ISO 27004 - Risk Measurement is used for, what poisoning attacks are in ML and how to measure the risks of poisoning attacks in ML with the given standards. RQ2 is intended to define how much impact poisoning attacks have on data sets based on various variables and how quickly tampering can be detected through risk measurement. The fourth research question RQ4 stands in relation with RQ3 and RQ8 because it could be possible that risk measurement for poisoning attacks and the attacker's effort contains risk indicators which are used for both. For RQ5, threat models find risk indicators to measure risks of the attacker's effort. This indicates the risks of an attacker and how big the extent of damage could come from the attack showed in different ways that are attacks that the attacker has programmed by himself or already finished attacks that are shown by the ART. RQ6 pursues the question which metrics of the ISO 27004 standard can be used to measure risks in relation of the attacker's effort. RQ7 is intended to summarize once again how risk indicators can support risk measurement through the framework. The last research question, RQ9, is the most important question because it aims to show that the framework is able to measure risks and show the extent of damage from all risk indicators.

2 Related Work

This chapter presents the relevant background knowledge and shows approaches from other scientific papers.

2.1 Security risks and risk assessment in context of Machine Learning

Security risks

Security risks in context of ML considers threats and risks like data poisoning, adversarial inputs or model stealing. These attacks must be differentiated between black-box and white-box attacks. Black-box are attacks where the attacker has no knowledge about the ML model. With white-box attacks, the attacker needs complete knowledge about the targeted ML model [30]. Adversarial inputs are inference data that are almost exactly the same inputs like the natural data but classified incorrectly [20]. Duplicating a ML model via model extraction attacks is model stealing [15]. Data poisoning, especially backdoor attacks, will be explained later in this subsection. Xiao et al. [35] evaluate the security risks in deep learning for common frameworks, for example TensorFlow. Xiao et al. use the framework sample applications along the frameworks. One statement of Xiao et al. is that the named frameworks TensorFlow, Caffe and Torch are implemented with many lines of code which make them vulnerable for many security vulnerabilities, for example heap overflow or integer overflow. The work of Xiao et al. is only in context of deep learning e.g. for neural networks.

Poisoning Attacks

Data poisoning attacks manipulate training sets of ML models to misclassify the scores. Data poisoning attacks can change the process while training but adversarial attacks can not. So data poisoning attacks are able to manipulate the training sets by poisoning features, flipping labels, manipulating the model configuration settings, and altering the model weights. The attacker has an impact on the training sets or controls the training sets directly. So the attacker wants to influence the ML model learning score [18].

Backdoor Attacks

Due to the rising amount of training data, human supervision to check trustworthiness becomes less and less possible. That exposes vulnerabilities in training sets like backdoors. Backdoor attacks can cause far reaching consequences. Backdoored models are able to classify on most inference inputs. But it can cause targeted misclassifications or can decrease the accuracy for inputs that the attacker chooses as secret properties referring as backdoor trigger [13]. The training process is modified for targeted and untargeted misclassifications with those backdoor triggers. Then the labels are altered, the configuration settings are changed, or the model parameters are directly altered [18]. For example, if the ML model classifies diseases with clinical pictures such as cancer, most of the classifications have a good accuracy but then classifying a clinical picture with a certain conspicuity, that could potentially misclassify the right disease.

Risk assessment

Risk assessment in context of ML is derived from classic IT security risk assessment. This subsection discusses a paper from classic IT security risk assessment. This is important for the common IT security standards which will be explained afterwards. Sendi et al. [27] evaluates the taxonomy of risk assessment and at which point in IT security management risk measurement takes place for the thesis and how it is carried out. In their paper, Sendi et al. evaluated 125 works published between 1995 and 2014. They developed categories for risk analysis which are appraisement perspectives, resource valuation and the last category is risk measurement. This category is the last step of risk assessment. To evaluate risks by measuring them, there are different properties which have an impact for risk measurement. Sendi et al. explain that the type of the attack, the dependency severity between resources and the type of defined permissions between resources are needed to measure risks. Risk measurement in their paper is differentiated between non-propagated and propagated. Non-propagated risk measurement stands in relation to the resource valuation category leading to the example of business driven risk assessment. Business driven is the view of business oriented goals and processes. And non-propagated risk measurement means that a model in which the risks are measured without the impact from other resources. For example, if the risks are measured business driven, the parameters such as business process are seen without the impact from other business processes. Propagated risk measurement concentrates on the attack impact and its propagation on other resources. The risk measurement is measuring the propagated risks as a dependency graph. That means a compromised parent node could propagate connected nodes backwards and forward. Backward impact means the impact propagation on all nodes that have a dependency with the compromised node and forward impact is the propagation from the compromised node to all its dependent nodes. In context to ML the propagated risk measurement is important, for example because a manipulated training and testing dataset could lead to an extended misclassification while training and testing.

2.2 Relevant standards for risk measurement

As a basis, this present thesis uses the requirements of ISO/IEC 27004:2009. ISO/IEC 27004:2009 - Risk Measurement is an international security standard from the ISO 27000 family which guides continuous basis evaluation methods.

ISO 27000 family

In their book, Kersten et al. [16] explain and discuss the management of the information security based on the ISO 27000 standard. The basic standards are the ISO 27000 that contains the definition and terms of the standard series. ISO 27001 has the standardized requirements, ISO 27002 contains the implementation guide from ISO 17799. ISO 27003 specifies the implementation of an IT security system. ISO 27004 measurement has the metrics and key figure systems. ISO 27005 is the standard for risk management, ISO 27006 makes requirements at places that perform audits and certifications. ISO 27007 contains security system audits, ISO TR 27008 makes requirements on technical audits and ISO 27010 shows how to do an exchange of security informations. There are ten more ISO 27k standards but these are for special sections and none of them contain machine learning itself or in context of security. Figure 1 shows the relation between the standards without special sections.

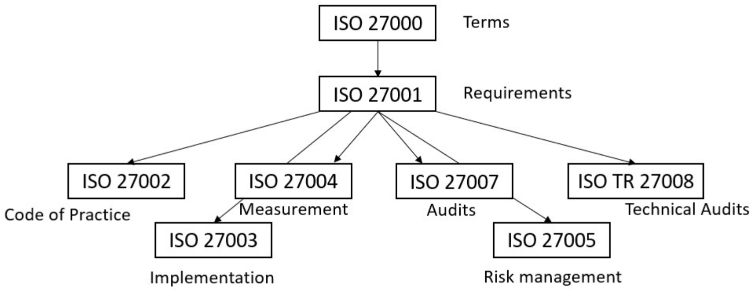


Figure 1: Overview of the ISO 27000 without special sections.

ISO standards for risk measurement

Kersten et al. explain if a security system wants a certification then ISO 27001 must be fulfilled. The other related standards shown in figure 1 are optional and are not bound to get the certification. For the RMF, ISO 27004 is the standard to measure risks.

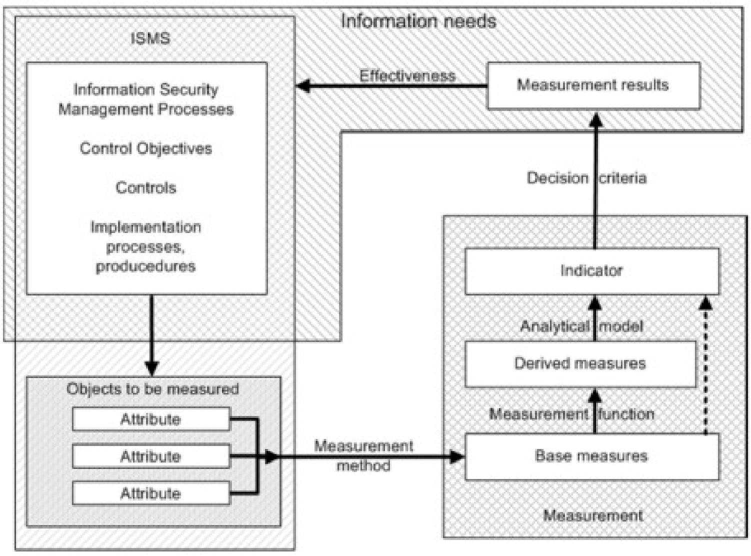


Figure 2: The information security measurement model [31]

The present ISO can be related with ISO 27001 or used as a standalone standard. As a requirement in ISO 27001, the effectiveness of an IT security system must be measured [4]. The ISO/IEC 27004:2009 standard specifies, what to be measured, when the measurement is needed, and types of measurement [19]. Barabanov et al. [4] and Tarnes [31] describe in their works the different properties of ISO/IEC 27004:2009 for risk measurement. Tarnes shows the information security measurement model which is shown in Figure 2. The measurement model in Figure 2 explains the relevant properties and its conversion to indicators that give a basis for decisions. The relevant properties show the needed information to the measurement objects [1]. For this thesis, these objects are the risk indicators that will be explained and discussed in Section 3. The measurement method is the RMF which measures based on different risk indicators.

2.3 The threat model for attacker characteristics

In their paper, Doynikova et al. [11] show a formal threat model with input data for experiments, the data handling process and describe the experiment that was executed. Doynikova et al. explain that the threat model can be split into high-level and low-level. These models contain attributes which used in this thesis as properties. High-level properties are subjective attributes that are obtained from monitoring the system. The gathered data are divided in four groups. The first group includes characteristics like skills, motivation and intention. The second group characterizes the attackers capabilities and show the characteristics as used resources. The third group incorporates the attacker in relation with the attacked system. This group includes the attackers location, the privileges, his goals, the access and the attackers knowledge. The attackers knowledge comes from the system where the objects are accessed before, access and privilege type and the detected activity. The last group relates the attacker with the attack and the steps that are included to execute the attack. The low-level properties can be used from the raw data directly during monitoring the system. The properties are classified into event logs, network traffic, namely and their source. The event log and network traffic is classified by origin, target, content, and temporal characteristics [12]. The attackers goal, destination of the attack or a normal action is monitored by the target characteristics. Content, payload or specifying and attack is monitored by content characteristics. Temporal characteristics contain time characteristics of the attack on a specific time interval and incorporate frequency. Doynikova et al. put an additional characteristic to the previously mentioned characteristics. The observable attack characteristics incorporate observables from the attack.
Now the high-level and low-level properties need to be mapped. Based on the low-level properties, the high-level properties can be calculated by mapping the low-level to the high-level properties like the attackers skills, resources and motivation. This formal attacker model is used to find, design and implement the risk indicators in Sections 3 and 4.

2.4 Approaches for risk measurement proposals and evaluation of risks of a ML model

This present thesis is divided into two approaches. Jakub Breier et al. [8] propose in their paper different proposals to measure risks with different aspects. These attacks are used in this thesis as properties to classify attacks. These different properties are attack specificity, attack time and attacker's knowledge. Attack time is split in training time and deployment time. Training time is the attack time when the model gets manipulated while it trains. Deployment time is the attack time when the hacker attacks a ML model after its release. Attacker's knowledge is the amount of information the hacker has available. Attackers specificity is the amount an attacker needs to manipulate the output of a ML model. These three properties may serve as a basis for further properties useful for risk measurement. Since these suggestions overlap with the characteristics from the threat model of Doynikova et al. these suggestions can be raised from the low-level and high-level properties.

Paul Schwerdtner et al. [26] is the second approach of this thesis. Schwerdtner et al. show a technical framework to evaluate the risks for ML models. Schwerdtner et al. give an evaluation whether it is secure to deploy a ML model or not. The ML model in Schwerdtner et al. must be a fully developed ML model that is trained and tested. Schwerdtner et al. concentrate on input data when the ML model has finished training and testing. The technical framework can test ML models under specific conditions in a scenario but can not find or measure risks while the training process. At this point the RMF would then find use.

2.5 Adversarial-Robustness-Toolbox

For this thesis the technical framework Adversarial-Robustness-Toolbox [23] is a main component. Nicolae et al. [22] evaluate in their work the technical framework ART. ART is a Python library that supports several ML frameworks for example TensorFlow and PyTorch to increase the defense of ML models. It is designed for developers who want to secure ML models. ART support 39 attacks and 29 defense functions. This thesis only focuses on the attack functions for poisoning attacks. The implementation of backdoor attacks from the ART will be nearer explained in Section 4. Every attack in the ART is based on an attack script. The class has three functions, *___init___*, *generate*, and *set_params*. *___init___* initialize with a classifier the attack, *generate* provide attack specific parameters and with the input x it applies the attack. *generate* returns in a numpy array the pertubated inputs. *set_params* returns true if the given attack-specified hyper- parameters are provided in the ***kwargs* dictionary. ***kwargs* is a argument that can pass keyword arguments dynamiclly e.g. the number of arguments may vary each time.

2.6 Scikit-learn

For this present thesis the Python library sci-kit learn is used for the case study in Section 5. Scikit-learn support supervised and unsupervised learning ML models [25]. The underlying basis library is Numpy for model and data parameters. The data input is declared as numpy arrays. [14] For linear algebra, special and basic statistical functions, and sparse matrix, scikit-learn uses Scipy [34]. The last library is Cython [5] which combines C in Python. For the thesis case study the focus is on supervised Support-Vector-Machines (SVM).
In his book, Bisong [6] explains sci-kit learn and using sci-kit learn in context to supervised ML. Sci-kit learn contains modules to implement ML models. These modules are sample datasets, preprocessing of the data, evaluation of the ML model and optimizing the performance of a ML model [9].

2.7 Support-Vector-Machine

In their book, Cristianini and Shawe-Taylor [10] explain linear learning and kernel-induced feature spaces that are relevant to understand SVM for this thesis.

Linear learning

In linear learning, linear classification classifies two training sets. Training sets are collections of training data. A hyperplane divides the space into two subspaces. [10] Figure 3 shows an example hyperplane where the parameters w and b control the function. w is the weight vector and b the bias. b moves the hyperplane parallel to itself and w declares a direction vertical to the hyperplane. The output is a set of w, one for each feature. The linear combination of the output predicts the value of the output result y.

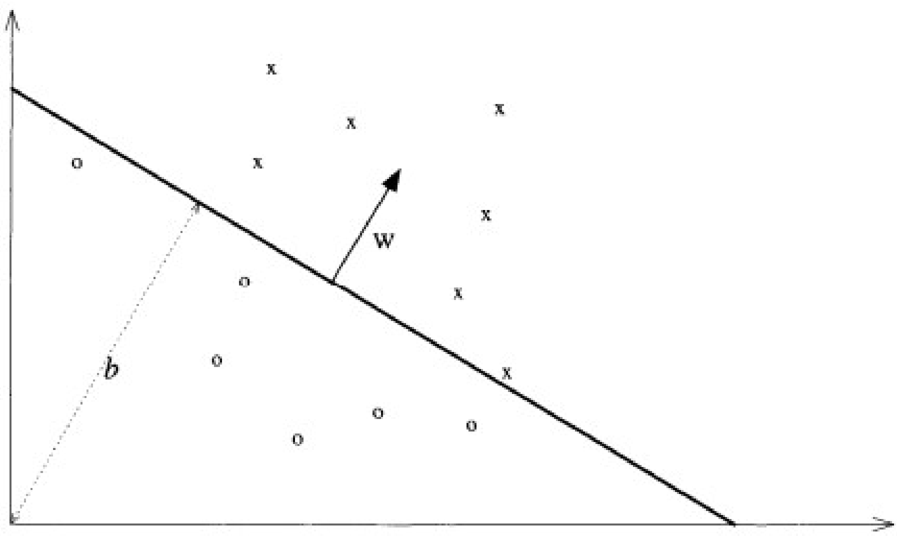


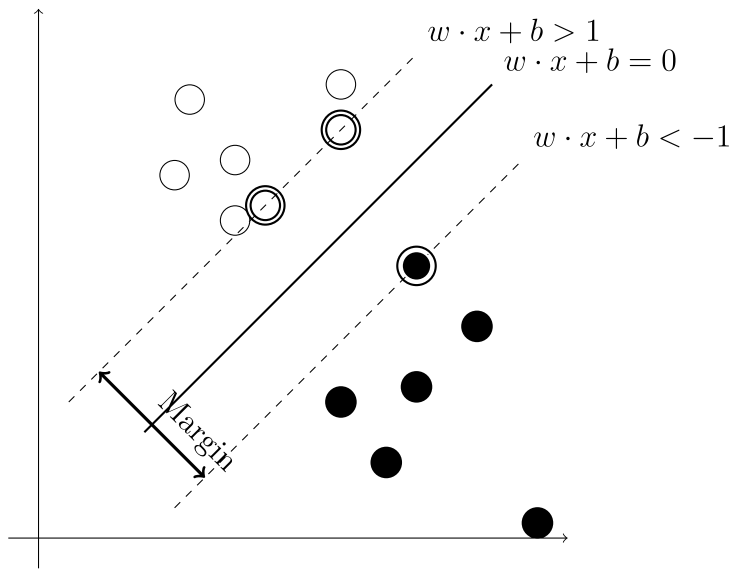
Figure 3: A hyperplane (w, b) showed in Cristianini and Shawe-Taylor [10] with a two-dimensional training dataset

Kernel-Induced Feature Spaces

If the target problem cannot be viewed as a linear combination of attributes kernel presentations are able to do it on SVMs. In Kernel-Induced Feature Spaces the data are projected in *N-dimensional* feature spaces to increase the used computational power of linear learning. To classify the data if there are more than two subspaces the SVM do a multi-class classification. The idea of multi-class classification is separating the classes to a linear classification [33].

Classification with Support-Vector-Machines

The support vector classification devise an efficient way to learn separating high dimensional feature space hyperplanes. Efficient means algorithms that can classify sample sizes of 100 000 instances. The easiest classifier is the maximal margin classifier that separates data which are linear separable in the feature space. The maximal margin classifier separates the data by the maximal margin hyperplane while the dimnensionality of the feature space is not relevant. This separation can be done in every kernel-induced feature space. [10]



Support Vector

Support Vectors are the minimum margin on both sides of the hyperplane. The maximum margin is the nearest object to the hyperplane in both classes.

3 The conceptual framework

In contrast to Schwerdtner et al., the framework of this thesis concentrates on training, especially Risk Measurement before and during training of the ML model. The conceptual framework discusses and explains the RMF. The RMF is a conceptual and technical framework which measures risks of backdoor attacks and measures the attackers effort. The attackers effort is measured by high-level und low-level properties. Section 2

3.1 Finding the attacker's effort

Subsection 2.3 explained a formal threat model to find the attackers effort with high-level and low-level properties where the low-level properties are mapped to with the high-level properties. At first this subsection will discuss which of the characteristics are useful to find the attackers effort for attacking a ML model. Regarding to the mapping between the properties, the low-level properties will be discussed at first.

The low-level properties

Using threat models to find risk indicators to measure the attackers effort

3.2 Characteristics of backdoor attacks

3.3 Risk indicators

The RMF measure risks by so called risk indicators. Properties, threat models and proposals are the basis for the risk indicators. Breier et al. in subsection 2.4 present proposals that are the approach for the proposals of the risk indicators. Doynikova et al. presents a formal threat model to find the attackers effort.

4 Implementation

The technical RMF uses Python 3.7 as the programming language and ART as the basis. Beside the attacks given by the ART, there is a function from the technical RMF to execute individual attacks. This technical RMF should be used a step ahead of using the framework of Schwerdtner et al.

4.1 Structure of the RMF

Directory tree

The RMF is structured as follows:

```
rmf/
├── attacks/
│   └── art/
│       └── backdoors.py
├── metrics/
│   └── log.py
├── visualizations/
│   └── plot.py
├── log_file.log
└── case_study.py
```

4.2 Using ART as the basis for the technical framework

The ART implemented two backdoor attacks which will be explained in 4.3. Since art is an open-source technical framework, the two backdoor attacks can also be used as a basis for simplifying the implementation of other attacks.

4.3 Implementing backdoor attacks

```
art_poison_backdoor_attack(perturbation, x, y, broadcast)
```

perturbation This argument...

x This argument...

y This argument...

broadcast This argument...

```
clean_label()
```

Backdoor attacks from the ART

PoisoningAttackBackdoor and *PoisoningAttackCleanLabelBackdoor* are the two backdoor attacks in the framework. In their work, Turner et al. [32] explain *PoisoningAttackClean-LabelBackdoor* attacks. Gu et al. [13] explain *PoisoningAttackBackdoor* attacks. Gu et al. [13] show in their work different backdoor attacks and do a case study with a traffic sign detection attack. In their work, Gu et al. developed a neural network with a backdoor trigger. The evaluated backdoors are a single pixel backdoor and a pattern backdoor. The single pixel backdoor increase the brightness of a pixel and the pattern backdoor adds a pattern of bright pixels in an image. The implemented attacks from Gu et al. are Single Target attack and an All-to-All attack. Single Target attack use the single pixel backdoor by changing a label from a digit i as a digit j. Gu et al. explained that the test data are not available for the attacker. The error rate for their Convolutional Neural Network (CNN) is 0.05%. The error rate with the backdoored images increases at most to 0.09%. An All-to-All attack change a digit label i to $i+1$. After testing the All-to-All attack the originial ML have a error rate of 0.03% while the ML with the backdoored image have an average error of 0.56%.

Additional attacks for the RMF

Beside the attacks that are called from functions of ART it must be possible to implement and execute new attacks for the evaluation to measure the attackers knowledge, skills and extent of damage.

4.4 Build in the risk indicatiors

The risk indicators are the main part for the risk measurement.

4.5 Measuring risks with the risk indicators

4.6 Implementation of the logging function

Show measured risks is able with logging from the Python logging module. The function waits for two parameters. A message string and the wanted logging level (i.e. INFO or DEBUG). The called log function in the RMF could look like this:

```
log(f"{variable_name}", 'INFO')
```

In order not to depend on the different ML libraries the rmf gets its own functions of the different metrics. That increases the support of different Python libraries for ML risk measurement. The accuracy of the predicitions are calculated as follows:

$$Accuracy = \frac{TruePositives+TrueNegatives}{TruePositives+TrueNegatives+FalsePositives+FalseNegatives}$$

4.7 Implementation of the visualization

For the visualization Python modules like sci-kit learn have implemented different plots that are signed as metrics.

5 Evaluation

A common example to show backdoor attacks is traffic sign detection ([21], [13], [24], [17]). That makes it easier to find datasets and already finished ML models to make a case study. The following case study uses a traffic sign dataset and show the risk measurement with the RMF.

5.1 Case Study: Developing a SVM for traffic sign detection

For the case study scikit-learn [25] and for preparation of the dataset in Python OpenCV2 have different function to load and resize images [7]. In their work, Stallkamp et al. [29] built a mulit-category classification dataset. The mulit-category classification dataset contains german traffic signs for image classification. That mulit-category classification dataset uses the german traffic signs from a approx. 10 hours daytime video from different roads. This case study is an example to show the functions and results of the RMF. After showing this case study there will be explain and discuss realistic case studies where backdoor attacks could have a more realistic impact for scores of ML models.

5.2 Preprocessing the original training sets

The original dataset from Stallkamp et al. is splitted between a training and testing folder. The training folder separate 42 signs into subfolders. This subfolders make it easy to use specific traffic signs which decrease the training time. The information of the folders are written in an eponymous csv-file that are not needed further in this case study. In Figure 4 the shown traffic signs can be used for training the SVM and are all labeled in the data preprocessing like the subfolder name 0 - 42.

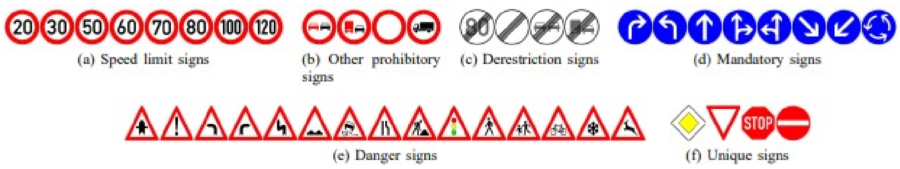


Figure 4: Labeled traffic signs [29]

All signs are resized to 300x300 pixel and are flattened for a higher efficiency. The training sets are also scaled with the scikit-learn *StandardScaler()* to increase the performance of the training time.

5.3 Differences between manipulated and original dataset

The Python plots from the case study show here based on different ML metrics the differences between the original and manipulated dataset.

5.4 Results from the risk measurement based on the risk indicators

5.5 Backdoor attacks in real applications

Beside the exemplary application from the case study, the scientific papers in this subsection show real applications where the RMF can then help in a more real environment.

6 Conclusion

References

[1] *Information technoloy - Security techniques - Information security management - Measurement*. ISO, 1st edition, 2009.

[2] Cyber-glossar, Jan 2021. `https://www.bsi.bund.de/DE/Service-Navi/Cyber-Glossar/cyber-glossar_node.html`.

[3] Machine learning glossary, Jul 2021. `https://developers.google.com/machine-learning/glossary/`.

[4] Rostyslav Barabanov, Stewart Kowalski, and Louise Yngström. Information security metrics: State of the art: State of the art. 2011.

[5] Stefan Behnel, Robert Bradshaw, Craig Citro, Lisandro Dalcin, Dag Sverre Seljebotn, and Kurt Smith. Cython: The best of both worlds. *Computing in Science & Engineering*, 13(2):31–39, 2011.

[6] Ekaba Ononse Bisong. *More Supervised Machine Learning Techniques with Scikit-learn*, page 287–308. Apress, 2019.

[7] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[8] Jakub Breier, Adrian Baldwin, Helen Balinsky, and Yang Liu. Risk management framework for machine learning security. *CoRR*, abs/2012.04884, 2020.

[9] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[10] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge University Press, 2000.

[11] Elena Doynikova, Evgenia Novikova, Diana Gaifulina, and Igor V. Kotenko. Towards attacker attribution for risk analysis. In Joaquín García-Alfaro, Jean Leneutre, Nora Cuppens, and Reda Yaich, editors, *Risks and Security of Internet and Systems - 15th International Conference, CRiSIS 2020, Paris, France, November 4-6, 2020, Revised Selected Papers*, volume 12528 of *Lecture Notes in Computer Science*, pages 347–353. Springer, 2020.

[12] Daniel Fraunholz, Daniel Krohmer, Simon Duque Antón, and Hans Dieter Schotten. YAAS - on the attribution of honeypot data. *Int. J. Cyber Situational Aware.*, 2(1):31–48, 2017.

[13] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *CoRR*, abs/1708.06733, 2017.

[14] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[15] Hailong Hu and Jun Pang. Stealing machine learning models: Attacks and countermeasures for generative adversarial networks. In *ACSAC '21: Annual Computer Security Applications Conference, Virtual Event, USA, December 6 - 10, 2021*, pages 1–16. ACM, 2021.

[16] Heinrich Kersten, Jürgen Reuter, Klaus-Werner Schröder, and Klaus-Dieter Wolfenstetter. *IT-Sicherheitsmanagement nach ISO 27001 und Grundschutz*. Springer Vieweg, 4th edition, 2013.

[17] Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Trans. Dependable Secur. Comput.*, 18(5):2088–2105, 2021.

[18] Jing Lin, Long Dang, Mohamed Rahouti, and Kaiqi Xiong. ML attack models: Adversarial attacks and data poisoning attacks. *CoRR*, abs/2112.02797, 2021.

[19] Kristoffer Lundholm, Jonas Hallberg, and Helena Granlund. Design and use of information security metrics. *FOI, Swedish Def. Res. Agency, p. ISSN*, pages 1650–1942, 2011.

[20] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[21] Tuan Anh Nguyen and Anh Tuan Tran. Wanet - imperceptible warping-based backdoor attack. *CoRR*, abs/2102.10369, 2021.

[22] Maria-Irina Nicolae, Mathieu Sinn, Tran Ngoc Minh, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Ian M. Molloy, and Benjamin Edwards. Adversarial robustness toolbox v1.0.0. *CoRR*, abs/1807.01069, 2019.

[23] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko

Ludwig, Ian Molloy, and Ben Edwards. Adversarial robustness toolbox v1.2.0. *CoRR*, 1807.01069, 2018.

[24] Florian Nuding and Rudolf Mayer. Poisoning attacks in federated learning: An evaluation on traffic sign classification. In Vassil Roussev, Bhavani M. Thuraisingham, Barbara Carminati, and Murat Kantarcioglu, editors, *CODASPY '20: Tenth ACM Conference on Data and Application Security and Privacy, New Orleans, LA, USA, March 16-18, 2020*, pages 168–170. ACM, 2020.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[26] Paul Schwerdtner, Florens Greßner, Nikhil Kapoor, Felix Assion, René Sass, Wiebke Günther, Fabian Hüger, and Peter Schlicht. Risk assessment for machine learning models. *CoRR*, abs/2011.04328, 2020.

[27] Alireza Shameli Sendi, Rouzbeh Aghababaei-Barzegar, and Mohamed Cheriet. Taxonomy of information security risk assessment (ISRA). *Comput. Secur.*, 57:14–30, 2016.

[28] Robert W. Shirey. Internet security glossary, version 2. *RFC*, 4949:1–365, 2007.

[29] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: A multi-class classification competition. In *The 2011 International Joint Conference on Neural Networks, IJCNN 2011, San Jose, California, USA, July 31 - August 5, 2011*, pages 1453–1460. IEEE, 2011.

[30] Elham Tabassi, Kevin Burns, Michael Hadjimichael, Andres Molina-Markham, and Julian Sexton. A taxonomy and terminology of adversarial machine learning. *NIST IR*, 2019.

[31] Marte Tarnes. Information security metrics: An empirical study of current practice. *Specialization Project, Trondheim, 17th December*, 2012.

[32] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Clean-label backdoor attacks. 2018.

[33] Angelos Tzotsos and Demetre Argialas. Support vector machine classification for object-based image analysis. In *Object-Based Image Analysis*, pages 663–677. Springer, 2008.

[34] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson,

C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[35] Qixue Xiao, Kang Li, Deyue Zhang, and Weilin Xu. Security risks in deep learning implementations. In *2018 IEEE Security and Privacy Workshops, SP Workshops 2018, San Francisco, CA, USA, May 24, 2018*, pages 123–128. IEEE Computer Society, 2018.

Selbständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und noch nicht für andere Prüfungen eingereicht habe. Sämtliche Quellen einschließlich Internetquellen, die unverändert oder abgewandelt wiedergegeben werden, insbesondere Quellen für Texte, Grafiken, Tabellen und Bilder, sind als solche kenntlich gemacht. Mir ist bekannt, dass bei Verstößen gegen diese Grundsätze ein Verfahren wegen Täuschungsversuchs bzw. Täuschung eingeleitet wird.

Berlin, den February 3, 2022 ....