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
International Organization for Standartization (ISO)	The ISO is an international statndard organization which declares international standards in all technical areas [2]. The ISO 27004 will be the basis for designing and implementing the Risk-Measurement-Framework.
Information Security Management System (ISMS)	
Model	A model in context of machine learning is a representation of a machine learning system that learned from the training sets [3].
Prediction	A predicition is an output of a machine learning model coming from an input e.g. data that the machine learning has never seen before [3].
Machine Learning	Machine learning is a research field and also describes a program or a system that trains from input data a predictive model. That predictive model makes predictions from never-before-seen data [3]. Machine learning models will be measured for security risks in this thesis.
Threat	A threat is a security violation when an event could cause harm [40]. Threats will be find in the threat models and the risk measurement.

Term	Description
Threat model	A threat model is a model to find security problems. This model should give a bigger picture away from the code [41]. A threat model will be used to declare different characeristics with their relation between attacks and an attacker.
Framework	A framework is a layered structure with a set of functions. It can specify programming interfaces, or offer programming tools [34]. In this thesis a framework will be designed, implemented and evaluated with a case study.
Vulnerability	A vulnerability is weakness of a program or system which could be exploited by a threat. [34]. The vulnerability term will be in this thesis used in context with the attacks and vulnerabilities in ML models.
Attacker	A person that exploits potential system vulnerabilities [34]. For this thesis an attacker stands in relation with its effort to attack an machine learning model.
Attacker's effort	The attacker's effort is a term that represent everything that an attacker do to attack the machine learning model. In this thesis the attacker's effort represent probability of occurrence, among other things.
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]. The risk calculation will be used as the final result after the risk measurement.
Risk indicator	Risk indicator is the generic term for all indications that contribute to the risk measurement in the framework, such as properties, values, and metrics. They will be used to measure risks and to represent the results as data, logs, or visual representations of the risk measurement.
Decision criteria	To describe the level of confidence in a given result, decision criteria pretend thresholds, targets, or patterns [1].
Measure	Variable which is assigned as the measurement result [1].
Analytical model	An algorithm that combines one or more derived measures with its associated decision criteria [1].

Term	Description
Measurement function	An algorithm to combine two or more measures [1].
Measurement method	Operations in a logical sequence [1].
Measurement	Gathering information about the IT security system by using a measurement method, function, analytical model and decision criteria [1].
Measurement result	One or more indicators with its interpretations which address an information need [1].
Information need	
Base measure	
Derived measure	
Metric	In context of this thesis, metrics will be used often to represent different results (i.e. accuracy) of the machine learning model.
High-level attributes	High-level attributes are subjective attributes that represent all characeristics of an attacker's effort. This term will be used in context of a formal threat model.
Low-level attributes	Low-level attributes are objective attributes that represent all characeristics of the attack's data. This term will be used in context of a formal threat model.

This thesis uses the terms decision criteria, measure, analytical model, measurement function, measurement method, measurement, information need, base measure, and derived measure in context with the ISO 27004 standard to design and implement the Risk-Measurement-Framework.

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. [11] and Paul Schwerdtner et al. [37]. 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.

Assumption

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 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 explains the relevant background knowledge, explain the state of research, and explains approaches from other scientific papers. The first five subsections explain theoratical parts for this thesis in context to the concept and design of the RMF in section 3. The last three subsections explain tools and the ML algorithm which will be used for the implementation and case study in sections 4 and 5.

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 information about the ML model. With white-box attacks have the attacker all information that he needs about the targeted ML model [43]. Adversarial inputs are inference data that are almost exactly the same inputs like the natural data but classified incorrectly [27]. Duplicating a ML model via model extraction attacks is model stealing [19]. Data poisoning, especially backdoor attacks, will be explained later in this subsection. Xiao et al. [48] evaluate the security risks in deep learning e.g. neural networks for common frameworks, for example TensorFlow and identify the vulnerabilites. 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. To consider attack surfaces of deep learning applications Xiao et al. uses the MNIST handwriting digits [23]. The attack surfaces are divided in three angles. Malformed input is the first attack surfaces and describe input data for classification which is read out from files what the MNIST dataset does. Xiao et al. describe that input data from sensors like camera sensors is significantly reduced because the sensors are directly connected to the ML model. The second attack surface are training data where this thesis is specified on. The training data can be polluted or mislabled when they come from external sources. That pollution or mislabling is also known as data poisoning attacks [8]. These attacks may not base on software vulnerabilites but flaws in implementations make data poisoning easier. Xiao et al. describes an example where they observerd in image parsing procedures inconsistency in a framework and desktop applications like an image viewer. The incosistency makes a sneaky data pollution possible while the training process is monitored. The last attack surface are malformed ML models which are used models by developers that took them from other developers. Further, if ML models are designed and implemented from scratch and developers use them but does not have ML knowledge, attackers can manipulating them easier. This attack surface can also assigned to data poisoning attacks because attackers can attack external models if the developers are not defend them from vulnerabilities. Beside of those attack surfaces, Xiao et al. describe different types of threats. The first threat are denial-of-service attacks which is the most common vulnerability and is caused by software bugs, infinite loops, or if memory can be exhausted. For example a bug in the numpy Python module is vulnerable and this module is the basis for TensorFlow. Another threat are evasion attacks that occur when the attacker use inputs that should classify as a certain label but instead it is missclassified as a different label. Evasion attacks are a possible attack if the ML model have software bugs (e.g. memory corruption bugs [32]). These bugs can be exploited when the classification can be overwited so the attacker can modify specific memory content or when the application control flow is hijacked to reorder the ML model execution. The last threat is system compromising with software bugs where an attacker hijack the control flow, leverage the software bug and compromise the host system.

Poisoning Attacks

Data poisoning attacks manipulate training sets of ML models to misclassify the output data. 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 output [25].

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 [17]. 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 [25]. 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. [38] evaluates the taxonomy of risk assessment and at which point in IT security management risk measurement takes place for this thesis and how it is carried out. Send et al. explain, that risk assessment is combined of risk analysis and risk evaluation

[21]. 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 [20], [22]. 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 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. This standard stands in no context with ML but will be derived to the RMF and thus also additionally to ML in section 3.

ISO 27000 family

In their book, Kersten et al. [21] 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.

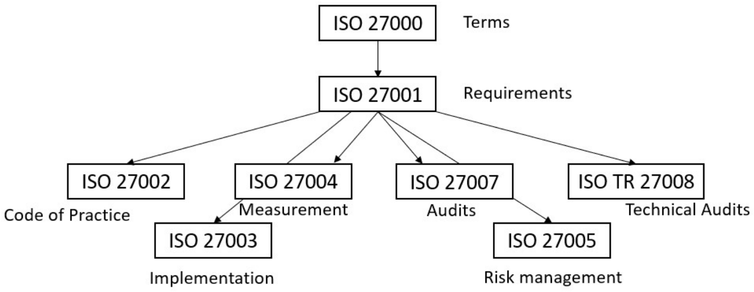


Figure 1: Overview of the ISO 27000 without special sections adapted from [21].

ISO standard for risk measurement

Kersten et al. [21] 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 and the other ISO 27000 standards are not considered further.

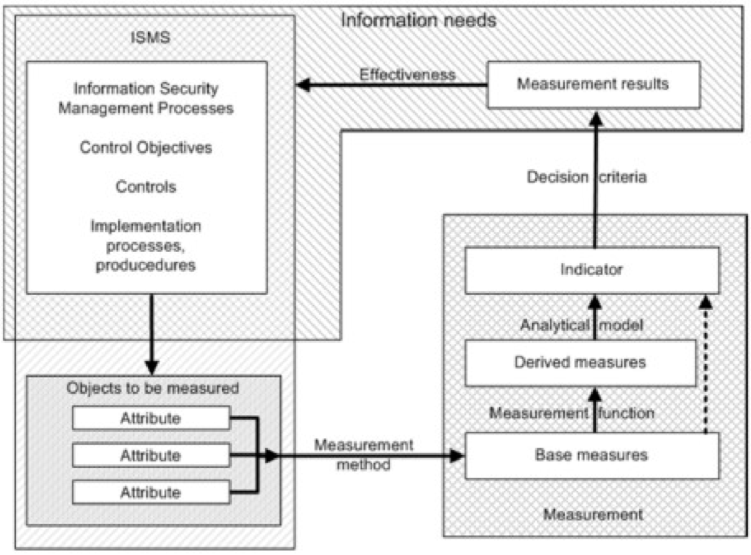


Figure 2: The information security measurement model adapted from [44]

This 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 [26]. Barabanov et al. [4] and Tarnes [44] 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.

ISO/IEC 27004:2009 specifies the requirements how to develop measures and the measurement in the following list [1]:

(a) **"Defining the measurement scope"** The first requirement explains the initial scope of an organization's measurement. It is based on different elements depending on capabilities and resources of the organization. These are specific controls and their protected information assets and information security activities. The management must prioritize all of them. Furthermore, the internal and external stakeholders should be identified and should participate on the measurement scope. It is possible that the organization sets a limit on the number of measurement results. This should ensure that the decision-makers can improve the ISMS based on the measurement results in a given time interval. The measurement results should be prioritized based on the importance of the corresponding information need [1].

(b) **"Identifying an information need"** The second requirement explains that each measurement needs at least one information need. An information need is identified in four activities. At first, the ISMS and its processes must be examined. Then the identified information need must be prioritized based on criteria, such as risk treatment, the organization's capabilities and resources, the interest of stakeholders, and the information security policy. The third activity bases on the list of prioritized information needs where a subset of information is required to be addressed on the measurement activity. The last activity is the communication to the stakeholder of the selected information need.
Based on the information needs, the relevant measures should be implemented into the ISMS [1].

(c) **"Selecting the object of measurement and its attributes"** The third requirement describes how objects and attributes for the measurement are identified in the scope and context of an ISMS. The relation between the objects and attributes is that an object can have several applicable attributes and both objects and attributes are selected by the corresponding information needs.
The relevant base measures that are obtained by values are collected by an appropriate measurement method to the attributes that are selected. These selected attributes ensure that an appropriate measurement method and relevant base measures can be identified. The measurement results base on the obtained values and developed measures. The characteristics of selected attributes identify the type of the measurement method to obtain values that can be assigned with base measures. All of the chosen objects and attributes need to be documented. Objects and attributes that are described by data should be used as values that are assigned

to the base measures. The attributes should be checked to ensure that they are appropriate for the measurement and for an effective measurement, the data collection should be defined in a way that sufficient attributes are available [1].

(d) **”Developing measurement constructs”** The fourth requirement defines the measurement construct development which starts by a measure selection that defines the measurement method, measurement function, the analytical model, indicators, decision criteria, and stakeholders. The measure selection should be defined in sufficient detail for the selection of measures that need to be implemented. If a new measure is implemented it needs to be adapted to an existing measure. Selected measures should represent the information need’s priority. Example criteria are, facilitation for data collection, facilitation for interpretation, and measures to calculate costs of analysing, and collecting the data. The third point of *”Developing measurement constructs”* explains how to define the measurement method for each measure. That measurement method will be used to quantify the measurement object by transforming the attributes into the value that is assigned to the base measure. Further, the measurement method can be subjective or objective. Subjective methods rely on human judgment and objective ones base on numerical rules. In the measurement method, the attributes are quantified as values. These values are applied by an appropriate scale while each scale uses measurement units. Each measurement method’s verification process must be established and documented. Furthermore, the precision of the measurement method and its deviation or variance should be recorded. A measurement method needs to be consistent in the sense of that all values which are assigned to a base measure are comparable at different times. These values should be also comparable to derived measures and indicators. The next part is about the measurement functions (e.g. calculations). The measurement functions may combine different techniques, for example averaging all values that are assigend to a base measure. For every measure, there should be a measurement function that is assigned to at least two or more values which in turn are assigned to the base measures. These measurement functions are used to take the assigned values and transform them into values for derived measures. The analytical model is defined for each indicator by transforming values that are defined to a base or derived measure. These values should be transformed into a value that is assigned to an indicator. Indicators are assigend values that are assigned to aggregated values which in turn are assigned to derived measures. These values are interpreted based on the decision criteria. The decision criteria are defined and should be documented based on information security objectives. Decision criteria are based on historical data, plans, and heuristics or calculated as statistical control or confidential limits. Lastly, stakeholders from base or derived measures are identified. Stakeholders can be clients, reviewers for measurement, information owners or information communicators [1].

(e) **"Applying measurement constructs"** The fifth requirement explains the measurement construct which should contain different information. These information are the purpose of measurement, measurement objects, collected and used data, the data collection process and analysis, the process that reports measurement results, stakeholders and their roles and responsibilities, and a cycle to ensure the usefulness of measurements including the relation to the information needs [1].

(f) **"Establishing data collection and analysis processes and tools"** The sixth requirement shows activities to collect and analyse data. The procedures of data storage and verification in the data collection should identify how the data is collected and stored with the necessary information. This should be done by using measurement methods, functions and analytical models. The verification is performed by using a checklist to verify a minimal data loss.
Analysis and reporting comes with the measurement results [1].

(g) **"Establishing measurement implementation approach and documentation"** The last requirement shows the amount of information which should at least be part of an implemententation plan:

   a) Information Security Measurement Programme in the organization

   b) Measurement specifiations:

      i. Organization's generic measurement, and

      ii. Organization's individual measurement construct

      iii. Range and procedures for data collection and analysis definition

   c) A plan as a calendar for measurement activities

   d) Created records through measurement activities with collected and analysis data records

   e) Formats for measurment results that are reported to stakeholders (described in ISO 27005)

   [1]

2.3. The threat model for attacker characteristics

This subsection explains a threat model without a reference to ML and which is used in the RMF. In their paper, Doynikova et al. [14] describe a formal threat model with its attributes. This threat model will be used for the RMF. Doynikova et al. suggest to split the theat model into high-level and low-level attributes. High-level attributes are subjective and can not obtained from monitoring the system under analysis. The monitoring system is described as features from network traffic and event logs. These logs and the traffic are described later in this subsection. The gathered data are divided in four groups. The first group includes characteristics like skills, motivation and intention. The second group characterizes the attacker's capabilities and show the characteristics as used resources. The third group incorporates the attacker in relation with the attacked system. This group includes the attacker's location, the privileges, his goals, the access and the attacker's knowledge. The attacker's 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 attributes can be used directly from the raw data during monitoring the system. Doynikova et al. consider these attributes as objective. The attributes 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 [15] which are explained next. The attackers goal, target of the attack or a normal action is monitored by the target characteristics. An attack is specified by content characteristics. Temporal characteristics contain time characteristics of the attack on a specific time interval and incorporate frequency. The observable attack characteristics incorporate observables from the attack that are not in the other four characteristics for example from the event logs.

Based on the low-level attributes, the high-level properties can be calculated by mapping the low-level to the high-level attributes like the attackers skills, resources and motivation. This mapping is a challenge for this thesis and the chraracteristics of both the high- and low-level attributes have to be derived for the RMF. Sections 3 and 4 map these attributes in the RMF.

2.4. Machine learning metrics

In their book, Nguyen and Zeigermann [28] describe that ML metrics are different calculations to evaluate a ML model. These metrics are linked to the loss function. A loss function serves as an optimization process and is used by the optimization algorithm to adapt while the next iteration iterates through the parameters of the model. The more a loss function is decreased, the better is the result. Further, the score is a value that is calculated while testing right after the training. The score and loss functions are summarized as metrics. In regard to these definitions, the terms score metric and loss metric will be used for the remainder of this chapter. The first score metric is the accuracy. The accuracy relates the number of data examples with true predicted labels to the number of all examined data examples.

$$accuracy = \frac{n(correctly\_predicted)}{n(all)}$$

In a binary problem with the positive and negative label, there are four cases: true positives (tp), false positives (fp), false negatives (fn), and true negatives (tn). These labels can be summarized in a $2x2$ confusion matrix.

$$\begin{pmatrix} tp & fp \\ tp & tn \end{pmatrix}$$

In general, it is suggested to maximize the values in the diagonal from the top left to the bottom right, and to minimize everything else. The values in the diagonal show the number of test examples that the ML model predicted correctly. All other values are wrong predictions. Further, to get the rates of the confusion matrix, the matrix values have to be divided by the number of examples per category.
The last metric is the precision-recall:

$$Precision = \frac{tp}{(tp+fp)}$$

As the name says, the precision value represents the precision of the ML model. It calculates the tp rate.

$$Recall = \frac{tp}{(tp+fn)}$$

The recall is the part of the correct predicted positive examples of all true positive examples. This value can be interpreted as the ML model efficiency. A third score for the balance between the precision and recall is the F1-Score.

$$\frac{2*precision*recall}{(precision+recall)}$$

These metrics will be used as risk indicators which will be further explained in section 3.

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

This thesis is divided into two approaches. Jakub Breier et al. [11] 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. [37] 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.

This third approach is for poisoning attacks and SVMs directly. Biggio et al. [8] describe an investigation on poisoning attacks against SVMs. Attacks against ML models are classified in causative and exploratory attacks [6]. Causative attacks are manipulations against training data and exploratory attacks are exploitations against the classifier. Poisoning attacks are refered to causative attacks. These attacks are important when the attacker have no direct access to the training database and provides his own training data from a web-based repository. Biggio et al. explain that the attacker's goal on a data poisoning attack is to find a point where the SVM's accuracy can be maximally decreased. Further, most learning assume that training datasets come from natural or well-behaved distributions. In their work, Biggio et al. assumes a worst case of the attacker's capabilities. The attacker also knows the ML model and can get the data from from a underlying data distribution platform. That is the security analysis methology [5]. Biggio et al. shows a method where the attacker construct a data point which decreases the SVM's classification accuracy. This method base on the on the SVM's optimal solution of the training problem. The training problem is formulated as the quadratic programming problem and for further explanation, please see [35].

An attacker can attack this SVM by inserting special attack points. Biggio et al. shows such a way assuming that the optimal solution of the SVM's training problem is retained. The SVM must be trained and the training points are as margin, and error support vectors, and reserve points referred. The attack is initialized by an attack vector which clones and arbitrary point from the attacked class and flipping the label from the point. For this thesis this work is a very generalized approach. There is no explanation of the type of poisoning attack and it does not show what affect this attack have on real-world scenarios. This approach goes more into the implementation and evaluation parts of the thesis and should show a first method how to attack a SVM.

2.6. Adversarial-Robustness-Toolbox

For this thesis the technical framework Adversarial-Robustness-Toolbox [31] is a main component. Nicolae et al. [30] 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. The attacks are evasion, extraction, inference, and poisoning attacks. The defences are detector, postprocessor, preprocessor, trainer, and transformer defences. 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.

2.7. Scikit-learn

For this thesis the Python library scikit-learn is used for the case study in Section 5. Scikit-learn support supervised and unsupervised learning ML models [36]. The underlying basis library is Numpy for model and data parameters. The data input is declared as numpy arrays. [18] For linear algebra, special and basic statistical functions, and sparse matrix, scikit-learn uses Scipy [47]. The last library is Cython [7] which combines C in Python. For the thesis case study the focus is on supervised Support-Vector-Machines (SVM).
In his book, Bisong [9] explains sci-kit learn and using sci-kit learn in context to supervised ML. Scikit-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 [12].

2.8. Support-Vector-Machine

In their book, Cristianini and Shawe-Taylor [13] 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. [13] 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.

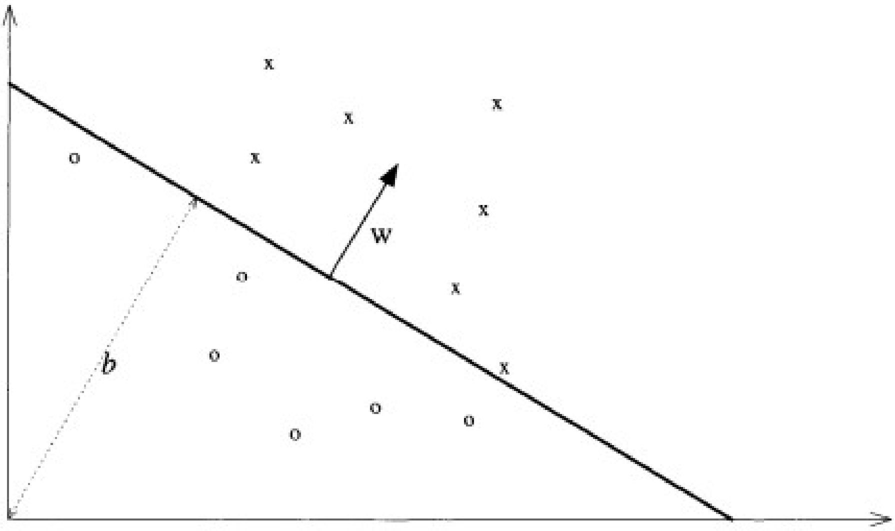


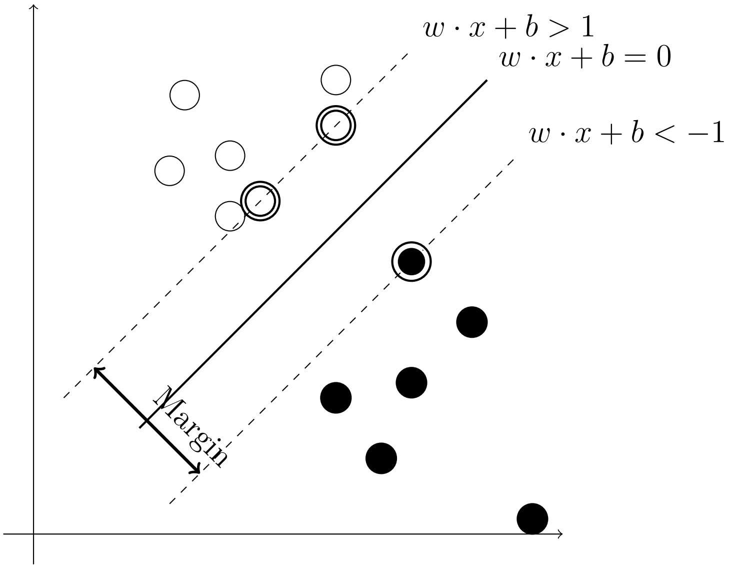
Figure 3: A hyperplane (w, b) showed in Cristianini and Shawe-Taylor [13] with a two-dimensional training dataset

Kernel-Induced Feature Spaces

If the target problem cannot be viewed as a linear combination of attributes, a SVM uses kernel presentations. 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 in case of more than two subspaces, the SVM performs a multi-class classification. The idea of multi-class classification is separating the classes to a linear classification [46].

Classification with Support-Vector-Machines

The support vector classification devises an efficient way to learn separating high dimensional feature space hyperplanes. Efficient describes algorithms that can classify sample sizes of 100 000 instances. The easiest classifier is the maximal margin classifier that separates data which are linearly 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. [13]



Support Vector

Support vectors are all inputs which lie closest to the hyperplane and the functional margin is 1. Only these points are involved for the weight vector and for that reason they are called support vectors. For further explanation of the mathematical structure, please see [13]. Figure 4 visualizes a maximal margin hyperplane example with support vectors as a binary classification.

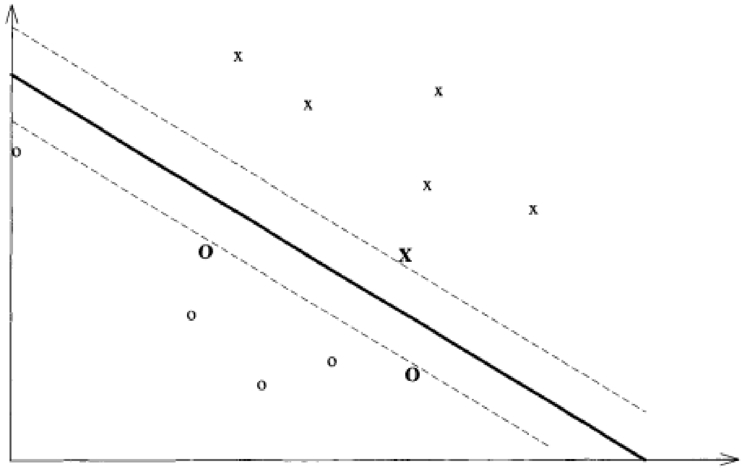


Figure 4: Maximal margin hyperplane with its support vectors adapted from [13].

3. Risk-Measurement-Framework concept and design

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 design of the RMF. The RMF is a technical framework which measures risks of backdoor attacks and measures the attackers effort. Referencing on this thesis approach from Biggio et al. [8] a security analysis for machine learning is that the attacker knows the ML model and can use the data from the data distribution platform. It is assumed for the RMF that the attacker knows the training data. This is an unrealistic assumption, but in real-world scenarios the attacker could use a surrogate training set instead, from the same data distribution platform which the developers use [5].

3.1. Using the standards for the risk measurement

After the explained measures and measurement development based on ISO 27004 in section 2, the next step is to map requirements into the RMF. This discussion what parts of them can be fulfilled and which parts can not fulfilled. That should show the process of risk measurement in the RMF.

At first, starting with (a) - ”Defining the measurement scope” where the organizations capabilites and resources define the initial scope. It starts by decisions of the management and can not be fulfilled by the RMF because that is an individual process specific for an organization and stands not in relation with the risk measurement of this thesis. The part defining internal and external stakeholder can not be fulfilled by the RMF but [39] describe how to identify them.

As next (b) - ”Identifying an information need” about the identification of an information need. The first activity of identifying the processes and examination of the ISMS can not be fulfilled of the RMF. The information need prioritization criteria like risk treatment can be fulfilled by the general riks measurement of the framework because all results are shown as transparent as possible. The organization’s capabilities and resources criteria is individual for the organization and cannot be measured. The stakeholders as already mentioned at the beginning are not considered further, therefore the fourth activity drops out completely. The third activity can be also fulfilled by showing all results of the risk measurement in a document. Based on the third activity the RMF gives a document as an output which template is shown in appendix B.

As in (c) described that objects and attributes are identified in the scope and context of an ISMS, the objects and attributes need to be related to ML metrics which are used to measure risks and calculate the final risk. The attributes, objects and measurement methods are described in the next subsections. The measurement results base on the risk indicators which are derived in each measurement method.

3.2. Characteristics of backdoor attacks

After discussing and evaluating the standards for the risk measurement in the RMF this subsection explain how the risks of backdoor attacks are measured.

3.3. Types of backdoor attacks

The following backdoor attacks should represent what they can achieve when using them. Further, this subsection should show the basis of the backdoor attacks that are used in the RMF.

The theory behind the ART backdoor attacks

PoisoningAttackBackdoor and *PoisoningAttackCleanLabelBackdoor* are the two backdoor attacks in the framework. Gu et al. [17] explain *PoisoningAttackBackdoor* attacks. The goal of this backdoor attack is to change their labels to a target label. This happens by attacking a random small selection of the training set and apply a backdoor trigger into the inputs [45]. Gu et al. 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%.
In their work, Turner et al. [45] explain *PoisoningAttackCleanLabelBackdoor* attacks. Turner et al. show an approach for executing backdoor attacks by utilizing adversarial examples and GAN-generated data. The point where Turner et al. start is analyzing effectiveness of Gu et al. attack while a simple technique is applied for data filtering. Turner et al. discovered that the poisoned inputs are outliers and are clearly wrong from the human inspection side. The attack would be ineffective if its rely solely on poisoned inputs which are labeled correctly and evade such filtering. At this point Turner et al. created an approach that do poisoned inputs which appear plausible to humans. The inputs need small changes to make them harder while classify them but the original label must still remain plausible. This transformation is performed by a GAN-based interpolation and adversarial bounded pertubations. GAN-based interpolation takes each input into the GAN latent space [16] and then interpolate poisoned samples to an incorrect class. Adversarial bounded pertubations uses a maximization method to maximize the loss of the pre-trained ML model on poisoned inputs while staying around the original input.

Additional backdoor attacks to increase the possible extent of damage

3.4. Finding the attacker's effort

Subsection 2.3 explained a formal threat model to find the attackers effort with high-level and low-level attributes where the low-level attributes are mapped to with the high-level attributes. 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 attributes, the low-level attributes will be discussed at first.

3.5. Using the formal threat model

For the risk measurement the attacker's effort is important to evaluate how high or low the risk is for an ML model. In this subsection the research questions RQ5, RQ6 are addressed in more detail. In reference to the research question RQ8 this threat model could be a possible method for the RMF to measure risks which will be proved in Section 5. As in subsection 2.3 explained the high- and low-level attributes have to be mapped to find the high-level attributes based on the low-level attributes.

The low-level attributes

To find the attacker's effort there is a need to collect data which can be measured from every attack on a ML model. These data is classified and explained in Subsection 2.3. Doynikova et al. explained which data is required to meausre the low-level attributes.

1. The first requirement is a dataset which contains information about the attack actions against a ML model. The information must be based on the skills, resources, intention, and motivation of the attacker.
2. The second requirment for the dataset is that everything is marked in such a way that the analysis shows which actions the attacker performed. But this requirement is more about having multiple attackers or the analysis accross multiple attackers.

The low-level attributes will also be used to measure the extent of damage based on the collected data.

The high-level attributes

Mapping the low-level with the high-level attributes

Derivate the properties to machine learning

3.6. 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 2.5 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.

Properties, proposals and characteristics derived from classic IT security

Correlations of the properties, proposals and characteristics

3.7. Evaluation methods for the measured risks

Analyze the dataset for vulnerabilites

Logging the execution of the attack

Machine learning metrics for risk measurement

Python plots

Calculate the risks

3.8. The final design to implement the RMF

The last point (g) of 3.1 shows the needed information for an implementation plan. This subsection fulfill parts of this. To measure the extent of damage there need to be an implementation of the low-level attributes. To get the attacker's effort another class should represent the measurement of the high-level attributes. These two classes need to be mapped.

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

Backdoor attacks from the ART

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

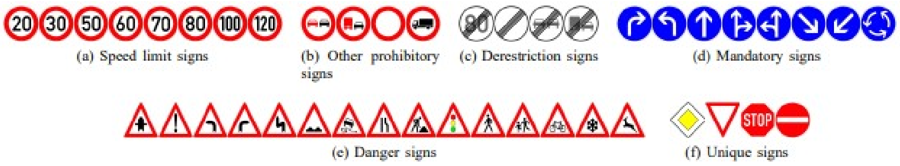
A common example to show backdoor attacks is traffic sign detection ([29], [17], [33], [24]). That makes it easier to find datasets and already finished ML models to make a case study. The following case study uses a traffic sign dataset and show the risk measurement with the RMF.

5.1. Case Study: Developing a SVM for traffic sign detection

For the case study scikit-learn [36] and for preparation of the dataset in Python OpenCV2 have different function to load and resize images [10]. In their work, Stallkamp et al. [42] 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 5 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.



(a) Speed limit signs

(b) Other prohibitory signs

(c) Derestriction signs

(d) Mandatory signs

(e) Danger signs

(f) Unique signs

Figure 5: Labeled traffic signs [42]

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

The ISO 27004 standard is very generalized. That makes it possible to design and implement frameworks which are not covered as a standard or intended to be.

6.1. Future work

A. Framework functions

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

perturbation This argument...

x This argument...

y This argument...

broadcast This argument...

```
clean_label()
```

B. Risk results template

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] Marco Barreno, Blaine Nelson, Anthony D. Joseph, and J. D. Tygar. The security of machine learning. *Mach. Learn.*, 81(2):121–148, 2010.

[6] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D. Joseph, and J. D. Tygar. Can machine learning be secure? In Ferng-Ching Lin, Der-Tsai Lee, Bao-Shuh Paul Lin, Shiuhpyng Shieh, and Sushil Jajodia, editors, *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2006, Taipei, Taiwan, March 21-24, 2006*, pages 16–25. ACM, 2006.

[7] 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.

[8] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012.* icml.cc / Omnipress, 2012.

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

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

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

[12] 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.

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

[14] 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.

[15] 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.

[16] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014.

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

[18] 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.

[19] 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.

[20] Marko Jahnke, Christian Thul, and Peter Martini. Graph based metrics for intrusion response measures in computer networks. In *32nd Annual IEEE Conference on Local Computer Networks (LCN 2007), 15-18 October 2007, Clontarf Castle, Dublin, Ireland, Proceedings*, pages 1035–1042. IEEE Computer Society, 2007.

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

[22] Nizar Kheir, Nora Cuppens-Boulahia, Frédéric Cuppens, and Hervé Debar. A service dependency model for cost-sensitive intrusion response. In Dimitris Gritzalis, Bart Preneel, and Marianthi Theoharidou, editors, *Computer Security - ESORICS 2010, 15th European Symposium on Research in Computer Security, Athens, Greece, September 20-22, 2010. Proceedings*, volume 6345 of *Lecture Notes in Computer Science*, pages 626–642. Springer, 2010.

[23] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The mnist database, 2017.

[24] 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.

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

[26] 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.

[27] 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.

[28] Chi Nhan Nguyen and Oliver Zeigermann. *Machine Learning kurz & gut.* O'Reilly Verlag, 2018.

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

[30] 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.

[31] 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.

[32] Bojan Novkovic. A taxonomy of defenses against memory corruption attacks. In Marko Koricic, Karolj Skala, Zeljka Car, Marina Cicin-Sain, Snjezana Babic, Vlado Sruk, Dejan Skvorc, Slobodan Ribaric, Bojan Jerbic, Stjepan Gros, Boris Vrdoljak, Mladen Mauher, Edvard Tijan, Tihomir Katulic, Juraj Petrovic, Tihana Galinac Grbac, Nikola Filip Fijan, and Vera Gradisnik, editors, *44th International Convention*

on Information, Communication and Electronic Technology, MIPRO 2021, Opatija, Croatia, September 27 - Oct. 1, 2021, pages 1196–1201. IEEE, 2021.

[33] 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.

[34] National Institute of Standards and Technology. Security requirements for cryptographic modules. Technical Report Federal Information Processing Standards Publications (FIPS PUBS) 140-2, Change Notice 2 December 03, 2002, U.S. Department of Commerce, Washington, D.C., 2001.

[35] Markos Papadonikolakis, Christos-Savvas Bouganis, and George A. Constantinides. Performance comparison of gpu and fpga architectures for the svm training problem. *2009 International Conference on Field-Programmable Technology*, pages 388–391, 2009.

[36] 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.

[37] 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.

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

[39] Helen Sharp, Anthony Finkelstein, and Galal Galal. Stakeholder identification in the requirements engineering process. In *10th International Workshop on Database & Expert Systems Applications, Florence, Italy, September 1-3, 1999, Proceedings*, pages 387–391. IEEE Computer Society, 1999.

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

[41] Adam Shostack. *Threat Modeling : Designing for Security.* John Wiley & Sons, Incorporated, 1st edition, 2017.

[42] 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.

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

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

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

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

[47] 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.

[48] 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 March 10, 2022 ....