PLATTFORM INDUSTRIE4.0

WORKING PAPER

Handling security risks in industrial applications due to lack of explainability of AI results

Imprint

Published by
Federal Ministry for Economic Affairs and Energy (BMWi)
Public Relations
10119 Berlin
www.bmwi.de

Text and editing
Plattform Industrie 4.0
Bertolt-Brecht-Platz 3
10117 Berlin

Design
PRpetuum GmbH, 80801 Munich

Status
September 2019

Illustrations
gettyimages
Degui Adil / EyeEm / p. 13
MEHAU KULYK/SCIENCE PHOTO LIBRARY / p. 15
Photographer is my life. / p. 4
seksan Mongkhonkhamsao / p. 8
Suebsiri Srithanyarat / EyeEm / p. 22
oxygen / p. 14
Yuichiro Chino / Titel, p. 10, p. 17

This publication as well as further publications can be obtained from:
Federal Ministry for Economic Affairs and Energy (BMWi)
Public Relations
E-mail: publikationen@bundesregierung.de
www.bmwi.de

Central procurement service:
Tel.: +49 30 182722721
Fax: +49 30 18102722721

Table of contents

List of figures

1. Introductory remarks

AI can be used sensibly in various areas of industrial development, production and manufacturing. This includes

- AI in **production planning**, e.g. by sequencing
- AI in **production**, e.g. for the monitoring of conditions
- AI in **process automation**, e.g. for the control of robots
- AI in **quality assurance**, e.g. for optical component testing
- AI in **logistics**, e.g. for route-optimised processes

In order to use AI safely and successfully in industry, however, a number of requirements must be met. This includes understanding the goals that AI can or cannot achieve and knowing the influencing factors inherent in using AI systems.

Basic aspects on this topic have already been described in a Plattform Industrie 4.0 paper entitled “Artificial Intelligence (AI) in Security Aspects of Industrie 4.0” and published at the Hannover Messe Industrie 2019. This publication goes beyond this by presenting the significance of explainability of AI for security aspects. The main question is how humans can understand the decisions of an AI system and detect any hidden errors in architecture, configuration, and training in order to correct them: what did the system actually “learn” and which influencing factors were decisive in this process?

Modern systems of artificial intelligence (AI) are moving further and further away from the notions of how biological brains function. For example, artificial neural networks today recognise patterns in image, sound, text or video with a dramatically higher number of nested neuronal layers than is assumed for the human brain. In many cases, this results in superhuman performance characteristics in terms of precision and accuracy of pattern recognition. On the other hand, however, systems of this type can no longer be interpreted using human powers of explanation. The precision and the ability of humans to understand the statements made by an AI system are increasingly constituting a trade-off.

This document focuses exclusively on the discussion of the security of procedures in the field of machine learning for industrial applications. In practice, they are mainly supervised learning algorithms in which training data are used to create an approximation that can produce reliable statements (predictions) in industrial processes.

In Chapter 2, basic cross-application techniques are explained in brief for the currently dominant field of artificial neural networks (ANN) in order to enable readers to understand industrial security problems even without a pronounced knowledge of AI. Chapter 3 deals with the problem of explainability and gives a brief overview of today's technology. There are no simple solutions (yet). Chapter 4 describes the most important method currently used by attackers to mislead AI. Here, too, there are (still) no solutions to problems without the pronounced expertise of specialists. However, users need to understand the hazard in order to assess their own risks. Chapter 5 provides comprehensive practical advice on risk minimisation in the use of AI in Industrie 4.0. The appendix in Chapter 7 briefly explains the basic terms of machine learning used in the text.

The use of artificial intelligence can pose new security problems. At the same time, AI is also suitable in some respects as a basis for new types of weapons, which can by no means only be used against AI-based systems, but also cause a generally extended threat situation. This paper is intended for both sides and provides explanations and information for operators.

2. Design, training and pre-trained systems

When designing a machine learning system, essential system parameters are pre-determined even before training is conducted. As with a traditional system, the input and output behaviour must be described, i.e. what are input parameters (syntax, representation and standardisation) and what result should the system deliver. It may be useful or necessary to apply (traditional) pre-processing steps (e.g. filtering, discretisation) to the input data. The quality of the results is intentionally or unintentionally influenced by preparatory steps of this nature. A suitable learning model for the ML system is then selected on this basis. In addition to neural networks, other models such as logistic regression (logit models), support vector machines, decision trees, etc. were and are used. At the centre of the current development and of this text are deep learning models with artificial neural networks. An essential part of the design of a neural network is its inner structure, i.e. the number of inner nodes (neurons), the number of layers, the density of the network of nodes, and the transfer function used. Although these decisions influence the performance and robustness of an ML system, they are often not transparent for the future user. For example, little information is available about the models and structures on which ML functions of large cloud service providers are based.

With a few exceptions, all machine learning algorithms require so-called "training" to determine their free parameters (weights of the respective approximation used). A large number of error corrections of these parameters are made on the basis of a randomly controlled initialisation in which the observed deviation between the "labels" (to describe the belonging to predefined classes) of data is compared with the results generated by these data. The weights are then changed so that this deviation is reduced. This process is called back propagation because, according to the rules of differential calculus, the derivation of the error in the result backwards after the weights through all neuron layers provides the direction for adjusting the weights: the negative gradient of the error as a function of the weights points in the direction of the steepest descent of the error. It is therefore used with a heuristically fixed increment for the entirety of the weights for their adjustment. Under suitable conditions of the error function, this gradient descent converges towards a unique global minimum. Figure 1 places the training process in an overall context of the use of AI – from system design through training to real use. It also shows which external influencing factors are important in which phase. Figure 2 then tries to explain the training process in detail.

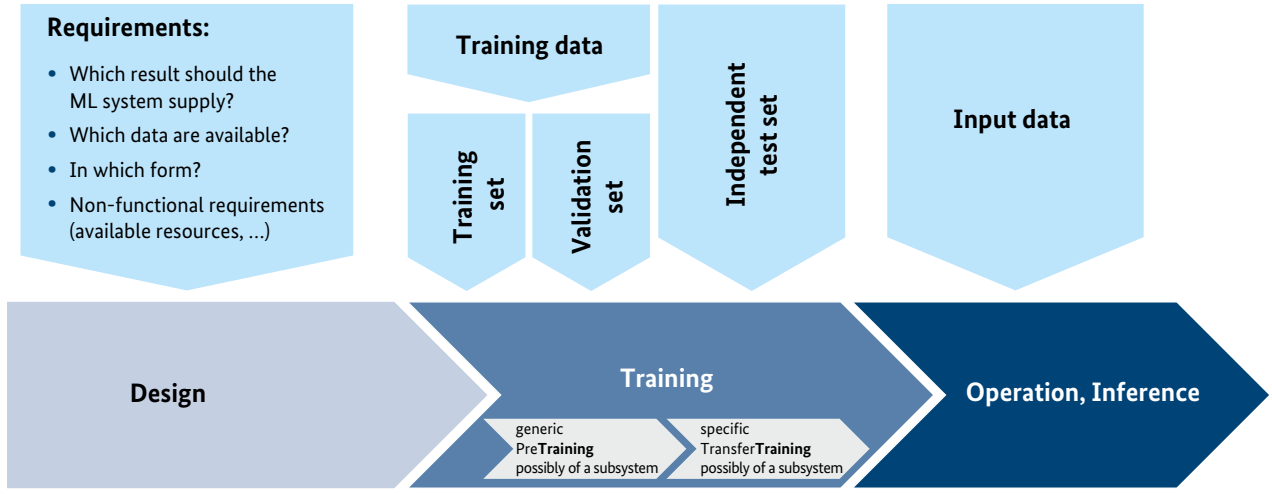
Since modern systems, especially neural networks, have very large numbers of free parameters, a meaningful training process must rely on very large numbers of training data to determine these parameters. The well-known AlexNet, with which the triumphal march of CNN (convolutional neural networks) in image recognition began in 2012, had around 65 million weights to distinguish 1,000 categories, and 1,000,000 examples with labels were available. According to common ideas about statistics, the examples obviously fall short by far. Various techniques have therefore been developed by which the number of examples can be increased (data augmentation). For example, images can be moved a few pixels up, down, or sideways without changing the content-related reference to the label. However, the question quickly arises as to where the limit for this multiplication lies and when the quality of the acquired knowledge suffers.

For each step of the gradient descent, the training data are bundled in so-called mini batches; the gradient is therefore determined for the average error and the composition of the batches is random (see Fig. 2). The epoch is described as that part of the training in which all existing data were used once. The process is then repeated and after each epoch, a success check is performed using some of the data that was previously separated from the training data.

As soon as it can be seen that the accuracy determined on this validation data does not increase any further, the training is terminated. Stagnation of accuracy in the validation set is an indicator of the start of overfitting (see explanation in the collection of technical terms, Chapter 7) – from now on further training would worsen the system's ability to generalise the classification with respect to unknown data. A final check on the basis of the test data, which has also been split off beforehand and which the system has therefore never seen in the validation, concludes the training for the selected hyperparameters (increments, normalisations, fixations, etc.). Further training runs are then carried out with different parameter sets until an acceptable result is obtained.

More than ten years ago, the technique of "transfer learning" (see explanation of terms) was developed, with which the chronic lack of data can be remedied in many situations. Instead of determining all weights of an ANN from scratch, the weights of the lower layers of an already trained ANN are reused. Only the upper layers are redefined for newly

Figure 1 High-level phase model



Source: Plattform Industrie 4.0

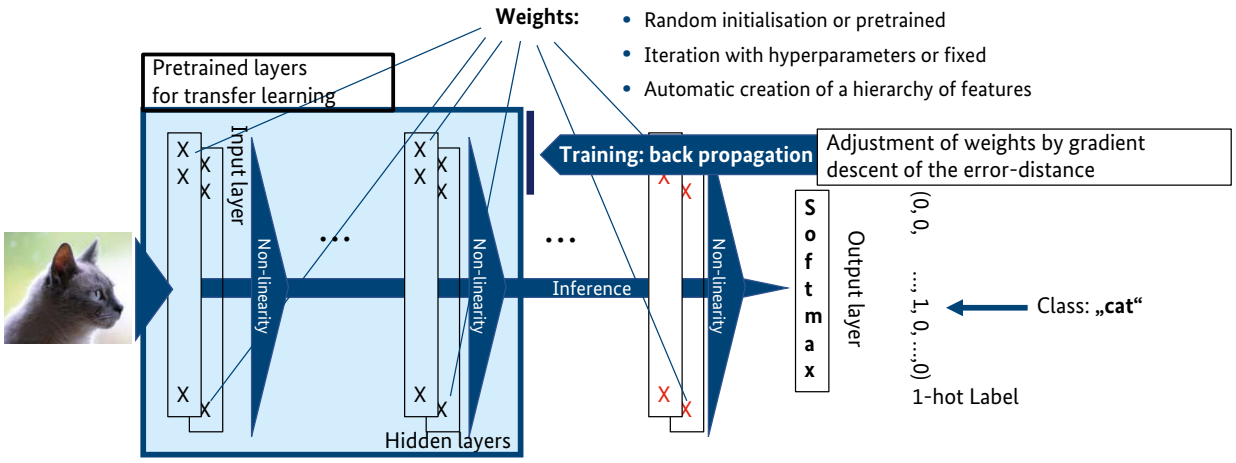
defined classes. Today, a very popular approach is to use "pre-trained" networks of this type. Transfer learning is much faster, since only a fraction of the weights have to be re-trained, and above all far fewer data are sufficient to achieve high hit rates in the final test.

A large number of pre-trained networks have become available in the open source area. Transfer learning can be the solution especially when it comes to recognising complex content that requires complex parameter structures whilst only very few sample images are available. A typical industrial example is quality control with non-destructive material testing. If complex defect patterns have to be detected in complex workpieces, it could take decades until sufficient defect patterns are available to train a neural network. New perspectives arise if pre-trained networks already exist from other projects that can be trained using transfer learning to recognise new classes.

The use of pre-trained networks creates risks as soon as it is not completely transparent what the pre-training actually consisted of. What has the new network already learned in its deep layers? What "biases"[1] are contained in these weights? Can the supplier who implements my ML system make statements about all (!) training data my network has ever seen? Are there any security risks with which my system has been infected in its "previous life" before the transfer? Such questions are the subject of this paper.

1 "A bias generally refers to distortion effects. In statistics, a bias is understood as an error within the framework of data collection and processing."

Figure 2: Training and pre-trained systems for image recognition. Other forms of visual perception (scene recognition, object recognition and localisation, and the distinction between classification and identification) are outlined in Chapter 7.



Data for:	
Training	Proliferation through augmentation (usually far fewer real data than weights in the system – statistics problem) Process in “mini batches“ (due to stability of convergence, typical size: 16–128 elements) and “epochs“
Validation	Control of the development of precision and timely discontinuation of training, variation of the hyperparameters
Test	Exclusively for final check of precision

Source: Plattform Industrie 4.0

2.1 AI applications for system protection

One of the best-known security measures to increase IT security in corporate networks is the intrusion detection system (IDS), which is applied to computer systems and computer networks. It can supplement firewalls and, in the advanced version as intrusion detection and prevention system (IDPS), actively and automatically prevent cyber-attacks piece by piece.

Three versions are offered on the market today: host-based (HIDS), network-based (NIDS) and hybrid (HIDS) intrusion detection systems. While HIDS analyses information from log files, kernel files and databases, NIDS is used to examine data packets in the network, while HIDS combines both principles in one tool.

A distinction must be made between two IDPS types for the systems mentioned: the detection of abuse versus the detection of an anomaly. For the detection of abuse, specific patterns are extracted from the modelling of an attack, against which the system is systematically searched. By contrast, anomalies are identified if the behaviour of the system deviates significantly from 'normal'. Therefore, a detected anomaly does not necessarily constitute an abuse.

IDPS, supplemented with AI functions, can be trained specifically with attack patterns, e.g. with denial of service attacks (DoS), but conversely also with so-called “normal states” in order to detect anomalies. In general, the most difficult problem of such procedures is the reliable detection of these normal states, especially when normal states can change dynamically and the appearance of new patterns is normal.

Market forecasts predict that by 2020 new technologies and methods, such as analytics, machine learning and behaviour-based recognition, will be integrated into most IDPS tools and offered on the market.

Despite this progress in the arsenal of defensive measures, modern IDPS-AI tools should not be seen as a universal solution against cyber-attacks, especially not for the future, but as a new milestone in the “tortoise and the hare race” between more intelligent attackers and defenders.

2.2 Security attacks against AI applications

- In the context of a security analysis of systems (products, applications...) two questions must essentially be answered:

1. What are the potential threats?

2. How can a potential threat be averted?

Re 1): Typically, threats are classified according to the CIA principle based on traditional, data-centred IT security according to CIA (confidentiality, integrity, availability). Added to this is authenticity.

Re 2): Threats to confidentiality are classically fended off by encryption or access control to data (or their physical storage and processing systems). Availability is secured by redundancy concepts and isolation from attackers. Integrity in the context of data processing means the possible verification of an unauthorised modification of the data. This is done classically by cryptographically secured checksums.

- Threat analyses of AI applications require an extended concept of integrity:
  - Data-centric “IT security” understands integrity as the possible proof of an unauthorised modification of data (see above). In the context of “industrial security”, the concept of integrity must be extended to the functionality of devices and systems: a system then has “integrity” if its executable functionality has not been changed (undetected); i.e. the system executes exactly those functions which are expected of it and are documented accordingly. The system should not be able to be manipulated unnoticed by third parties.

- In the case of AI-based functions of a device or system, the execution of a function cannot be foreseen or described in the individual case, since the system is not “predictable” for every case in the context of its internal decision making. In addition to the traditional integrity of devices and systems, the aim is to ensure that decisions

  AI1: are unbiased,

  AI2: do not exceed defined system boundaries, and

- AI3: answer the “question” of the questioner “analogously” depending on the available data. A traditional verification test against individual functions or features defined in the specifications is usually not possible or not effective, since it is not obvious or should not be specified at all “how” and on the basis of which features an ML system reaches a classification.

In particular, the question now arises as to how malicious attacks against the properties AI1-AI3 can be fended off or at least detected. Can traditional security methods such as checksums, encryption or redundancy be applied here? Are known mechanisms sufficient or do new methods have to be developed? In particular, how can the properties AI1 and AI3 be tested?

- Example from “safety”: an AI-based system should behave within the limits of certain expectations. An autonomous machine/robot should not leave the safety area or should in principle prevent collisions with other machines or people. Of course, this behaviour characteristic must not be affected by malicious manipulation of the underlying AI application or input data.

- Question: is “manipulation of input data” a threat which can be defended against? Are there (upstream) plausibility tests as protection mechanisms for AI applications?

- Similar questions are known from the test of (strongly) nonlinear cryptographic functions with very large definition and value sets (> 2128). Despite the correct verification of given test vectors, it may turn out later that there were still errors in the implementation.

There is still a need for research to solve the above-mentioned questions.

3. Explainability of AI decisions

3.1 The need for explanations through human argumentation

The extent to which it is necessary, conceptually meaningful and technically possible to make AI decisions explainable or interpretable in the sense of a classification for human argumentation is a controversial topic today. In some fields of application, there are obvious business justifications for the desire for explanation: why was the production line brought to a standstill by an autonomous robot and who is to blame? Why was my credit application rejected even though my perceived financial status is better than average? As a rule, these are questions of guilt, liability and the correctness of diagnoses for which there is a desire for humanly comprehensible argumentation. On the other hand, it can be assumed that this desire will diminish and be replaced at least in part by greater confidence in machine decisions.

There will be more and more areas in which machines will be more precise, much faster and much more reliable. "Would you rather undergo heart surgery by a renowned professor with a mortality risk of 8 % or by a robot with a mortality risk of 2 %? Would it be important for you that the professor, in contrast to the robot in the negative case, could explain to your surviving dependants in an understandable way why the intervention failed? Moreover, in many science-based practices today it is perfectly normal to rely on statistical experience rather than rational explanation, even in critical areas. Medicine and pharmacy provide numerous examples. The mechanisms of action of common drugs such as aspirin were only researched decades after their widespread use and the general testing practice for pharmaceuticals goes from laboratory animals via special patient groups to "normal" medication, but not via rigorous explanation.

It is undisputed, however, that explainable AI (XAI) can be helpful when assessing what a system has actually learned and in fathoming whether content distortions in training data are reflected in decision-making behaviour. For example, sensitivity analyses of the decision with regard to the pixels of input images showed that, given the remarkably high reliability in distinguishing between wolves and huskies, a main criterion in the successful image recognition system was the snow in the background. Of course, the system should not learn that a wolf-like animal on snow is probably a husky. The providers of the training pictures were certainly not aware of this feature of their training set. The example clearly shows, however, that training data can have dangerous bias properties which – intentionally or unintentionally - can cause technical, ethical and political shortcomings. Such mistakes can also be the subject of discussions in the context of the GDPR.

3.2 Complexity of the explanation problem

Only the new algorithms of machine learning (ML) have produced and created awareness for the problem of the diminishing explainability of AI decisions. ML solves classification problems by approximation: observations are approximated by a "hypothesis". With millions of observations, mathematical functions can be calibrated ("trained") with millions of parameters – this is the new quality of extremely complex hypotheses (e.g. in neural networks). While programmed decision trees or the parameters of linear regression and other linear models were still largely explainable in their man-made construction, this no longer applies to modern neural networks with over 20 to hundreds of neuron layers. People do not think in such complex argumentation structures and the features that neural networks form in these layers have no equivalent in human pattern recognition.

The problem that arises is therefore that advanced ML techniques develop superhuman abilities in prediction, but at the same time lose explainability in human reasoning. AI specialists can use sophisticated analysis techniques to reveal why their system made which decisions from a technical point of view (sensitivity analyses). In this respect, neural networks are not "black box systems" for AI specialists. XAI is therefore primarily concerned with making the superior performance of modern ML technology available to domain-specific users without AI expertise through improved explainability. The as yet unclear objective on the way to context-sensitive AI is to determine in which form human expertise and reasoning might help to reduce dependence on exorbitant training data sets and thus inspire learning algorithms that learn like biological brains with far fewer examples.

3.3 Research approaches

Sensitivity analyses have long existed that relate to components of training data: which pixels in images affect classification decisions and to what extent. Since both input and classification are humanly understandable, such techniques provide information about hidden biases in training data.

The expected modularisation in ML technology represents an analogy to the conceptual revolution of “structured programming” in the 1970s, when “spaghetti code” was replaced by software in today's sense.

The greatest conceptual challenge remains the identification of explanatory features in an apparently intelligently acting system in humanly comprehensible conceptual images and argumentation. How can an argumentation of 24 levels, for example, be explained to people who cannot intellectually grasp more than 3 to 4 levels? The current approach is mainly to use further AI to explain AI: neural networks that explain to people what the intellectually superior neural networks meant by their statements.

3.4 Existing techniques

In principle, an attempt can be made to use complex and therefore also very precise systems as a black box, and to use simpler but more explainable systems to explain the predictions of the precise complex system. The simple system uses the classifications of the complex system as labels. This approach is very controversial. It is often referred to as “global” because here the system as such, not a single prediction, is to be made explainable.

Other techniques focus on local events in the sense of the system's statements for specific classifications: “why did the system make this statement for this particular input?” Various software solutions are now available for this area, which are used in real projects:

- LIME: “Local Interpretable Model-Agnostic Explanations” can be used to analyse any black box system to determine the importance of specific features for a given prediction. It is currently the most popular package.
- “Shapley Value Explanations” are based on the game theory concept of the Shapley value, which determines the fairest method for distributing the yield in a cooperative game. It is currently the most up-to-date and scientifically sound approach, but requires very high computing power because a large number of replacement models must be trained. The “SHAP” package tries to make the existing concepts compatible with the Shapley value approach by approximation with realistically possible computational effort.
- The LSTMVis project aims at describing the explainability of neural sequence models generated by recurrent neural networks [1].
- There are various other approaches and offers which can, however, be basically integrated into these leading methods.

3.5 Examples

- AI in text analysis: recognition of individual words, their synonyms and the context on the basis of domain-specific knowledge for conclusions on content or intended statements or evaluations. Areas of application are subject-specific training and support. AI may, from a superficial point of view, convey a certain semantic recognition - in fact, it is not the recognition of the meaning of facts that is learned, but only the contexts in which people's words were previously used. A “higher” intelligence is not the basis of this field of application.
- AI-based translation systems: mainly available online, they are based on learned text correlations of different foreign languages, which consist of existing translation material. If only high-quality training texts previously produced by qualified people (technical translators) are learned, AI systems can produce good quality results with a sufficiently broad and high-quality range of training texts. However, AI systems do not improve the quality of inferior training texts. Low quality training texts from “pre-training” can have a permanent negative effect if the text currently to be translated has not been learned in detail and the algorithm used delivers a “translation text” “under all circumstances”. A qualitative grey area exists for online offers. A “higher, semantic” recognition using AI is not the basis for these systems either.

- Highly automated robots in production require, among other things, 3-D environmental information to prevent collisions in production, for example. Two information elements must be processed very quickly by AI: a) image analysis of the environment, b) prediction of the motor movement, e.g. of the robot, of the robot arm but also of the workpiece.
- AI in intelligent automation uses all data for available means of production, for workpieces to be machined, for necessary processes, but also for quality controls, intermediate storage, post-processing, etc.

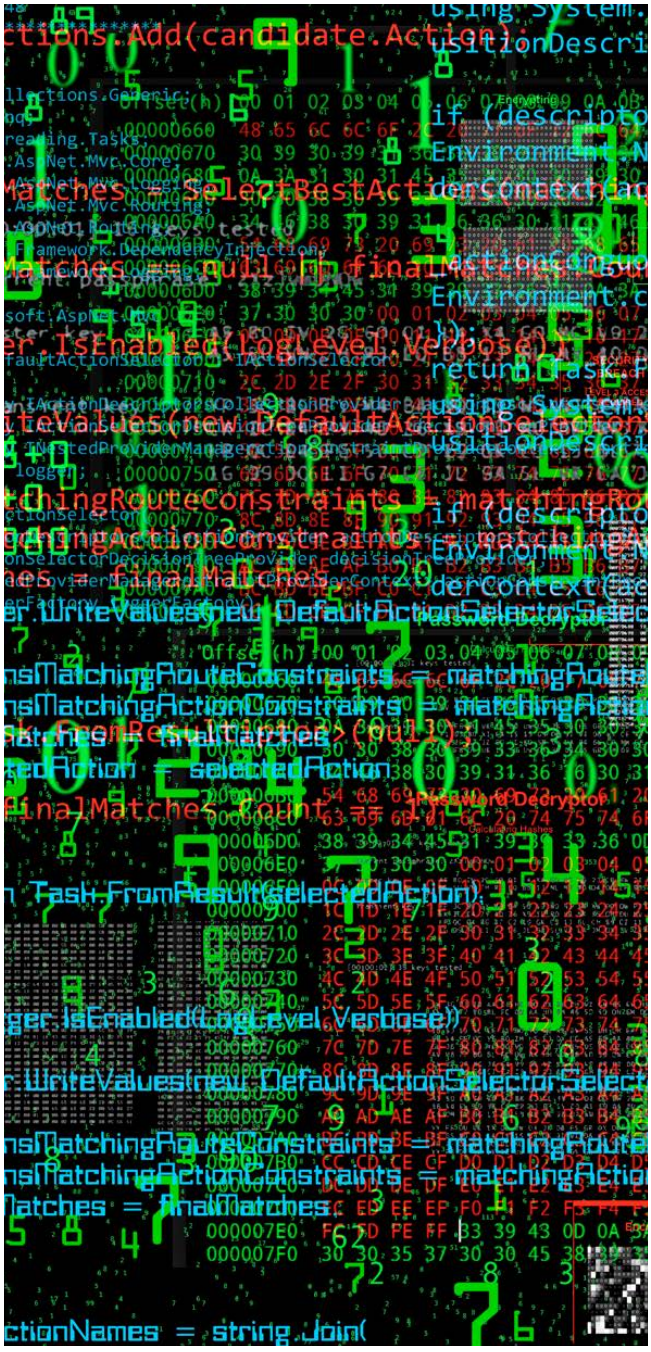
4. Attack vectors on AI or by means of AI

AI-based systems are subject to risks that did not exist in earlier systems. Such risks arise in training processes with undocumented or humanly incomprehensible data, through misunderstood architecture, process or performance characteristics and through other structural characteristics that users at least partially do not know. Information and advice are required so that operators do not unconsciously take risks. In addition, AI-based attack techniques have developed over the past five years, which can be particularly dangerous for AI-based systems, but which also challenge the entire industrial security discussion and require more comprehensive protection concepts than those established to date.

Adversarial examples are input data that are specifically constructed to lead pattern recognition systems to incorrect classification. The system recognises something completely differently than a human observer would. How is this possible and what techniques are used by attackers? This chapter illustrates the process using the familiar and simple example of the Modified NIST data set (abbreviated MNIST data set) of handwritten digits. It was created in the 1990s by the NIST agency to automate mail sorting in the USA and consists of 60,000 28X28 pixels of 8-bit greyscale images. Professional CNN achieve hit rates of over 99 %, while simple systems such as logistic regression achieve over 93 %. Practically all new ML concepts are tested on this data set because it is so easy to use and easily accessible.

The problem starts with the power of the modelling by ML: the images obviously lie in a 28X28 = 784-dimensional space of integers with values between 0 and 255. There are 256^{784}, and therefore more than 10^{1888} possible images. If it is considered that physics estimates that the visible universe has "only" about 10^{80} atoms, then it can be seen that the number of possible MNIST images is very (!) large. "Normal" photos from a digital camera span even dramatically larger spaces. Random compositions of pixels in MNIST images are perceived by the human eye as noise where nothing is recognisable. Images that a human being recognises as numbers (0, ..., 9) form only a tiny subset - with perhaps a few million elements, but are infinitesimally small compared with 10^{1888}.

It would therefore be obvious to search for images that a network trained with MNIST recognises as a number, even though they do not look like that to humans. For this the same algorithm can be used that was used to determine the weights of the network, but which is now applied to the 784 pixels. In fact, a large number of noise images can be found to which the trained network assigns high probabilities that a certain digit is involved: what noise is for humans, the network considers as 3, 4, 7, ... with a high degree of confidence. With a slight change in the target function for the gradient descent, for example, a search can be started for patterns that are interpreted as "3", but for humans look like a "7", for example. The perversion can be taken to the extreme and this procedure can be used to develop different images for each image from the MNIST test set that look identical to humans, but are classified by the trained network as any other digit with a confidence of more than 90 %.



There are ways to fight such attacks. A very simple way is to switch to pure black and white pixels. The misclassification is caused by grey noise and is therefore largely eliminated. However, this only works in images that still make sense in this representation. This is applicable to writing, but not normally to photos, especially if the resolution is low and the content only becomes vivid through grey tones. Another possibility is to include the incorrectly recognised images with the correct labels in the training. This technology also works but to apply it, the adversary's examples must be known. However, the defender will normally not know the target function the attacker will use to generate his images. Therefore, the identified gap can only be closed after a successful attack.

Even more dramatic is the threat scenario that arises from the transferability of adversarial examples between different ML systems [3]. An example that deceives a given system is very likely also capable of deceiving a completely different ML system. Examples that fool a particular, say, two-level CNN are often able to fool a CNN with more or less or differently configured layers, and even simpler techniques, such as logistic regression, or very complex techniques, such as ResNet or Inception, are vulnerable to the same attacks. The attacker does not have to know the technology of his victim; it can be a black box for him, but he still has good chances of starting a successful attack.

Adversarial examples can thus be produced in stock, so to speak, in order to launch targeted zero-day attacks at a given time. It is an extreme case of asymmetry between attacker and defender who cannot know the attack vectors that are hiding at the ready.

Resistance and defence against adversarial examples is a task for ML experts, who are able to develop an individually adapted framework by means of suitable, very specially aligned system configuration and training measures, which demand great effort from potential attackers in order to develop successful attack examples. The literature on this subject is very extensive and complex. It is primarily a matter of creating situations for ML in special cases of application that can guarantee resistance to adversarial examples in a differentiated way. A comprehensive general solution to the problem is still the subject of research, but this is already possible today for specific applications.

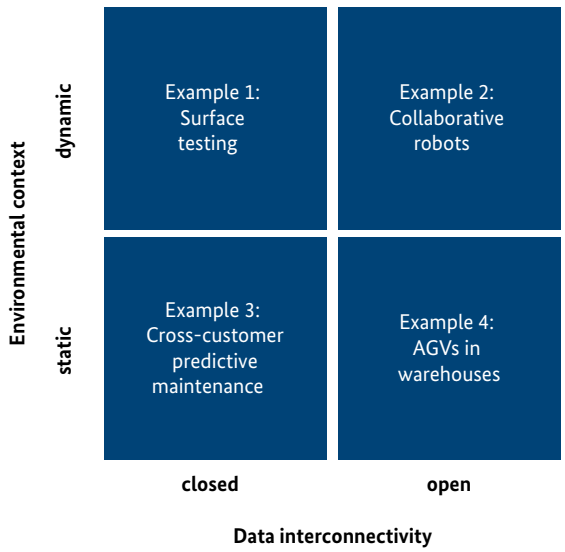
5. Security risks associated with architectures and applications

Distributed industrial architectures and applications represent well known and important features of Industrie 4.0. However, the secure interaction of AI architectures with this new infrastructure is largely unexplored. Security risks, such as industrial espionage, sabotage and data theft, must be considered in reflection of the chosen AI architecture and intended application. AI applications in production can have different classifications and criteria for this, such as

- **Static** versus **dynamic** application of AI; this refers to the systemic adaptation of the AI software;
- **Closed** or **encapsulated** application with only one user or one application, versus open application with dynamic number of users or applications..

The terms “classification” and “criteria” could also include the term “degree of freedom” or, more generally, “complexity” of the AI used. Depending on the number of degrees of freedom of the AI application, different threat scenarios must be considered, which makes this case distinction important.

Figure 3: Four-quadrant representation of the degrees of freedom of an AI application. AI in autonomous application and pooling are not shown.



Source: Plattform Industrie 4.0

Example 1: An example for a **static**, **stationary** and **closed** application is the use of a camera in the course of a fully automatic quality control of the surface of a metal moulded part before this free-form part is transported to the painting line. The camera examines the surface of the blank for possible deformations that deviate from the nominal shape, such as a dent or a wave that may have occurred during deep drawing. After painting, such very slight deformations would become very visible to the eye and the free-form part, e.g. a wing, would be discarded. The AI software evaluates the camera image and compares it with reference images. The reference images have been permanently further developed through constant learning. The camera and the AI software work separately from other production and quality steps. This measuring and decision circle can therefore be regarded as autonomous from other hardware and software in the production environment. Data from other camera systems are not required because it is used on other free-form surfaces, such as a vehicle door, where other target dimensions exist. In contrast to human surface checking, the system of camera and AI software can deliver significantly more accurate results and can produce a consistently high quality.

Closed or encapsulated application in this example means a routine activity of AI in production, which is supervised by a single operator. A camera, evaluation software and a component or workpiece with high repetition in the geometric dimensions characterise this application. The security risk for cyber-attacks aimed at industrial espionage or sabotage is manageable, as the perpetrator and the offence can be identified very quickly.

Example 2: Another example of a **dynamic**, spatially expanding and **open** application is the interconnection of collaboratively cooperating robots and their control via a control station. This something for the future. Several production robots in a car production constantly exchange environmental data almost in real time via a control station containing AI software. The AI is intended to anticipate possible collisions of very fast running robot arms and workpiece parts to be machined and to prevent them by intervening in the process sequence. In addition, programmers can import new production processes into each individual robot, which in turn must be transferred to the AI software as a new parameter set. The AI is networked with many robots, each of which offers different production processes. The process programming of each robot is constantly optimised by several operators. The measuring and decision circle can be described as open, since the amount of production means involved constantly varies and therefore forms a dynamic factor, and the processes are constantly revised by operators and are therefore also highly dynamic.

Example 3: An example of a largely **closed** application of an AI is the post-classification of security alarms from a (cyber-)security application by a security expert. In comparison to Examples 1 and 2, it describes controlled, **dynamic** further learning of AI during operation with corresponding dynamic adaptation of the decision behaviour. In this case, the software manufacturer supplies test algorithms that can produce false alarms in the context of the respective application environment or the respective company using the software. Since the individual operational environment parameters can vary greatly, the manufacturer cannot supply any AI parameters that classify the result of the test algorithm in the respective environmental context. Typical classifications of such alarms include classification according to real attack alarms, false alarms, false configuration alarms, company policy alarms, etc., which require different reactions in each case. Due to the post-classification of alarms by a security expert, many "labelled" data records are created during the operation of the software, which are fed directly to the AI for dynamic re-learning during operation, thereby contributing to automated classification.

Example 3 can easily be extended to Example 1 if a quality control by an employee results in a different evaluation of the test result of the AI and the deviating evaluation can be made accessible to the AI of the test system as a "labelled" data record via corresponding software.

Example 4: The use of intelligent, self-navigating transport platforms (Automated Guided Vehicles, AGV) in warehouses and distributed production landscapes, such as in aircraft construction, serves as a good example of AI applications that are both **open** and **static** in terms of learning behaviour. The openness results from the demand-oriented use of more or less participating AGVs in the applications. For example, a warehouse may put additional units into operation during the hectic Christmas season in order to increase order processing capacity. A further dimension of openness results from the diversity of the elements to be transported. Logistics applications with contact to external parties (e.g. goods packaged by customers) in particular can become accessible from the outside in unexpected ways. In principle, the underlying intelligence of the individual transport platforms is not significantly modified, i.e. there is no further learning of the navigation system and it can therefore be marked as static. This is not least due to the requirements of functional safety, which exclude or at least severely restrict the modification of certified systems. Typical AI functions in this area are concentrated on single platform navigation and fleet management. However, it should not be overlooked that dynamic learning processes are increasingly being implemented at the level of fleet management. A coexistence of dynamic and static elements in the same application is foreseeable here.

In the above cases, the main differences between the AI applications are therefore as follows.

What does this mean for a faulty or deliberate manipulation of AI, e.g. for the purposes of economic sabotage? AI in closed application can be procured, put into operation and controlled by a single person. In the event of sabotage, the perpetrator would be immediately exposed and a countermeasure quickly initiated. In the open application, possible causes would first have to be investigated, such as incorrect programming of the robot controller, absence of an update of new process parameters or absence of a message that another robot is integrated into the process sequence. A possible perpetrator responsible for a case of sabotage would be difficult, if not impossible, to identify. The large number of machines and people involved and the high complexity would protect the saboteur from exposure.

The above question of local impact goes hand in hand with a further dimension, namely controllability, triggered by the use of AI in the cloud (i.e. data centres). Typically, AI applications are trained in the data centres. Thanks to the higher computing power available in these data centres, they are able to meet the performance requirements of the AI applications. The AI inference systems are then used in the edge. In exceptional cases, both processes can be performed both in the data centre and in the edge (distributed AI), depending on the complexity requirements.

The distinction between open and closed data interconnectivity also provides information on possible hazard situations:

Table 1

	AI in static application	AI in dynamic application
System properties	more likely closed, compact	more likely open, complex
Complexity in the AI application	more likely low	more likely high
Required computing power of the AI software	more likely low	more likely high
Participating means of production and operators who supply inputs for the AI software	more likely few	more likely many

Table 2

	Static application	Dynamic application
Local impact in the case of sabotage	more limited	larger
Force of sabotage incident	smaller	greater
Recognisability of the situation	more likely possible	more likely impossible
Elimination work required after detection	more likely less	more likely more

Table 3

	Closed system	Open system
Recognisability of the attack	high, probably	difficult, requires strong, complex protection mechanisms
Effects of the attack	more likely small, since proliferation is limited	potentially very great, as further spread is difficult to prevent
Work required to eliminate the attack	more likely small amount	very much, correlates with the extent of the infestation

Difference between cloud and distributed edge applications

- Intelligent networks in the industry demand very fast system behaviour, which cloud applications cannot always deliver within limited latency requirements. Low latency requirements are often critical for industrial applications such as automatic controls, coordinated robot motions (see Example 2), video surveillance for product quality testing (see Example 1), power distribution, false alarm prevention, etc.

- The large number of assets connected to the cloud requires large bandwidth for the transmission of raw data, and the cloud requires large computing and storage resources to process this large volume of data. In addition, the raw data often contain sensitive information and attacks on this data in the cloud can also cause considerable damage to the companies involved.

  Network operators are also increasingly shifting intelligence to the edge in order to optimise networks and reduce network congestion. Distributed edge computing represents a desired topology from the point of view of a network operator. Edge computing is a distributed open platform at the edge of networks, close to things and data sources, that integrates the capabilities of networks, storage and applications and also supports local inference systems.

  This architecture is complemented by Multi-Access Edge Computing (MEC), which combines elements of information technology and telecommunications networks and enables network operators to open their networks to authorised third parties such as application developers and content providers.

  New AI systems from network operators optimise these load requirements and meet Industrie 4.0 requirements for agile connectivity, real-time services, data optimisation, application intelligence, security and data protection significantly better.

  In this situation, however, it is important to recognise the challenge and to separate the functionalities of AI for operation in the cloud from those for operation in the edge on end devices, such as smart devices. High-performance chipsets and edge computing platforms are required to implement and execute the high-performance algorithms of artificial intelligence for inference. It is also important to support the continuum of knowledge exchange between the AI functions in the edge and those in the cloud.

Control of networks by AI

In this scenario, the training data sets for characteristics recognition of different network requirements are of particular importance, since the network AI is considered a critical infrastructure whose possible misconduct can cause collateral damage similar to that of autonomous systems, which must ultimately be assessed as trade-offs.

In all probability, the idea of using the third known algorithm – namely reinforcement learning for these requirements – for the chaotic system states typical in networks is likely to fail because the trial-and-error principle is unlikely to be feasible in critical infrastructures. The reinforcement learning algorithm moves closer to the optimum step by step by going through a variety of iterations, combining proven behavioural patterns and randomly trying out new behaviours.

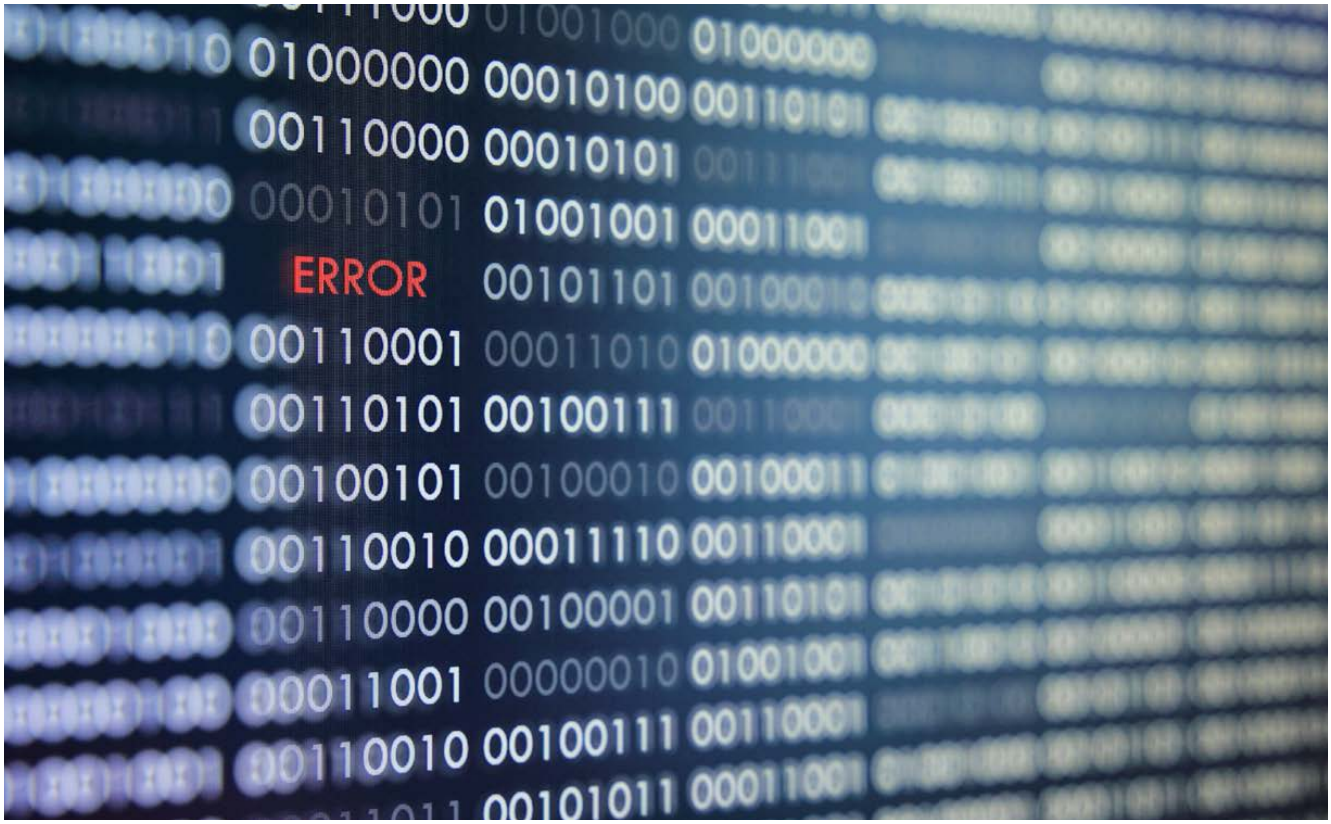
5.1 Security risks when using AI systems

The risk of using AI systems is basically the risk of making wrong decisions. The reasons for wrong decisions on the part of AI can have various causes. These problems, initially independent of security aspects, can be read in the standard literature on AI [4]. A relationship between these general problems and security aspects is discussed under the respective individual points.

The listed reasons are additionally assigned to the respective phases of the AI phase model in Figure 1

List of reasons:

- **AI design and training phase, unclassified relevance of training data:** training data are often mass data, ideally covering the value spectrum combinations for each of the n parameters of these data in coarse grids, and labelled according to combinatorics. Particularly important here are also demarcation data records, where the labelling should lead to a different assessment by AI with comparatively minor changes. In the above Example 1 (surface quality control), parameters for the measurement of 'roughness' would have to be determined here in such a way that the quality control would just return the



value 'error-free' or would just the value 'incorrect'. Very often, however, insufficient attention is paid to the relevance of the data to the coverage of the value spectrum. In this case, the value spectrum of individual parameters is either not completely covered or goes far beyond the target corridor in one direction and remains far below the maximum/minimum value of the desired target corridor in the other direction.

In addition to the general difficulty of covering the n-dimensional grid of parameter values, the data can also be manipulated as an attack vector at the time of training, so that the relevance spectrum would not be covered by the manipulation.

As a result, wrong decisions can occur for the peripheral areas, or even random decisions for parameter values that are outside those of the training data sets, during the runtime of AI.

- **AI training phase, false training data, generated by deliberate manipulation of the AI:**

  - Intentional manipulation of training data to provoke misconduct is referred to as poisoning.

    Training data determine the behaviour of the ANN; if the training data is manipulated, the ANN can behave differently than expected. With clever manipulation, the manipulation of the training data cannot be seen and also the ANN behaves apparently still as expected. For certain input data, however, unexpected or even undesired behaviour will occur.

    This is comparable to the behaviour of software that contains hidden, malicious code. The software usually behaves as it should and the damage is not obvious, but under certain conditions the malicious code becomes active and causes harm. The situation in the case of an ANN is comparable with closed source. It is very difficult to find the malicious code and to detect the malicious behaviour in time. This is because the source code cannot be analysed as it is not available.

- The occurrence of malfunctions in a deterministic system with several sources of error is extremely difficult. Under supposedly identical conditions, the system sometimes behaves correctly and sometimes incorrectly. There is no 100% correlation between input and abnormal behaviour. As a rule, such problems can only be found and solved by systematically limiting the test conditions and gradually analysing the system behaviour (debugging).

  For ANN, the analysis problem is intensified, because its functionality is not strictly deterministic viewed from the outside (e.g. wrong recognition of a digit by an NN due to pixel noise in digit images cannot be perceived by a human observer).

  Common implementations of ANN provide a history/log of input data. With the help of this history, the behaviour of the ANN can be reproduced and analysed. This is used, among other things, to identify and eliminate the causes of the ANN's malfunction. This mechanism can also be used to analyse and limit poisoning.

- In principle, the danger of poisoning is also reduced by federated machine learning. The models of ANN trained in isolation are aggregated in a superordinate ANN model. The individual ANNs of the federation will then be replaced by the aggregated model.

  If one of the systems is attacked with "poisoned" learning data, this misconduct has little effect because it is superimposed by the correct models.

  Federated machine learning is not suitable for all applications of neural networks.

- **Two risk types are to be considered:**

  - The AI software was manipulated in pre-training and thus introduced into production incorrectly;

  - The AI software is manipulated in the production by cyber-attacks in the decision; the cyber-attacks can be carried out by a single production plant or quality control plant, which constantly reports false data back to the AI and thus leads to false decisions by the AI.

- **AI training phase, the ANN is overfitted** (overfitting, see Basic Terms, Chapter 6): During the training of AI, the labelled data sets are often fed to the AI several times in learning iterations until the deviation during testing with the help of a validation data set becomes small. In the case of the overtrained system, i.e. too frequent learning iterations, AI learns the results of the labelled data sets more or less by heart. The evaluation result based on the validation data supplied which the system does not know from training, deteriorates because the system begins to lose its generalisation capability. Checking the learning process with validation data indicates when the training should be terminated to avoid overfitting. If small parameter deviations occur in productive operation compared to the rules learned by heart, AI can no longer generalise well enough after overfitting and generates wrong decisions.

- **AI training phase, the ANN is underfitted** (underfitting, see Basic Terms, Chapter 6): The AI system has insufficient training data or the variance in their parameter values is too small, so that AI delivers good results in test data operation, but is too generalised in productive operation, cannot grasp the complexity of the task and may unilaterally classify the data sets supplied to it in the same way and only classifies them differently in the case of very large deviations from the training data.

- **AI training and operational phase, false dynamic learning:** in the context of supervised machine learning, where learning is done dynamically by adding further data during the productive use of the ANN (see Example 3: Controlled dynamic learning), a misinterpretation of similar situations can be permanently induced by incorrect assessment of a current situation (generation of false negatives/positives, see below). Dynamic learning is a complex topic that can generally only be implemented meaningfully through highly specialised human intervention with a high level of application-related expertise. The naive notion that intelligent systems continue to learn and improve during operation is fundamentally wrong. The unfiltered incorporation of the system's findings into a set of basic truths can lead to the loss of the AI system's ability to recognise.

- **AI operating phase, changed environment compared to learning environment:** Basic risks arise in pre-trained systems when AI is used in changed environments where situations may occur sooner or later that were not included in the example data during the learning phase. Generalisation by AI always means a complex form of interpolation between known data from known situations. Events that are completely different from the learning environment cannot be handled by an AI system.

The two basic types of wrong decisions are:

- **False positives:** AI provides a high level of criticality when evaluating the criticality of a state, even though a low level of criticality is concerned. In the context of Example 1 (quality inspection), AI would decide on a production defect although this is not the case. In Example 2 (collaborative robots), AI would detect an imminent collision of the collaborative robot product entities, but this would not take place, and possibly provoke a corresponding reaction of the system, which would not be necessary or in the worst case could only lead to a collision. In Example 3, (cyber-attack) alarms that are not alarms are detected.

- **False negatives:** AI provides a low level of criticality when evaluating the criticality of a state, even though a high level of criticality is concerned. In Example 1, existing quality defects would not be detected. In Example 2, an imminent collision of the entities involved would not be detected or would be detected too late. In Example 3, a (cyber) attack would not be detected.

Both cases can be caused by the reasons listed above at the time of learning, or by changes in the production environment.

5.1.1 Consequential risks: loss of production and corresponding additional costs

The range of consequential risks due to wrong decisions on the part of the AI can be assumed to be arbitrary. Depending on the processes of a company and the subsequent reactions, production stoppages, faulty production or destruction of machines can occur. For example, if the Industrie 4.0 entity is automatically shut down due to a false positive and the number of false alarms is very high, it may be impossible to resume production at all.

The security risks that arise in production environments differ from the type of use and environment. In particular, a distinction must be made here between the use of static and dynamic AI systems, as described above. In general, the more open the system is and the more variable the resulting parameter values are, the greater the security risks with regard to decisions made by AI.

Note: The likelihood that such a risk will occur through the use of AI is not necessarily higher than through the testing of fixed patterns and the use of other monitoring mechanisms. In open systems, testing through fixed patterns almost completely fails, since these do not provide for any changes (or only those defined changes) in parameter values and combinations.

5.1.2 Security risks when using static AI systems

With reference to the above-mentioned reasons for wrong decisions, the security risks are discussed with reference to Example 1 and partly to Example 3.

- **AI design and training phase, unclassified relevance of training data:** in a closed system, wrong decisions in a production start-up phase are detected relatively quickly by follow-up checks, so that in the event of any possible failure to cover the parameter areas (both by disregarding the relevance areas and by manipulating the training data records), appropriate follow-up learning or new learning can take place. In Example 1, corresponding learning data records could then be added that cover or better describe the borderlines between 'incorrect' and 'error-free'.

- **AI training phase, false training data generated by deliberate manipulation of the AI:** in the case of Example 1 and Example 3, manipulation would probably be performed with the intention of not recognising existing quality problems in order to provoke a high level of damage (false negatives), which is only noticed later on. The corresponding consequential risks arise as described above.

- **AI training phase, the ANN is overfitted:** in the case of Example 1, the risk of wrong decisions by an overtrained system is not necessarily increased as long as the learning records already contained all cases that occurred. In general, however, deviations from the learning data records are to be expected in productive operation, so that wrong

decisions can be expected the greater the variance of the real data records. A security risk arises especially from a conscious/unconscious manipulation of the operational environment, as the number of wrong decisions increases massively in both directions and productivity decreases.

- **AI training phase, the ANN is underfitted:** in Example 1, the risk of the undertrained system is low, since this is immediately noticeable when commissioning an Industrie 4.0 system due to the high number of wrong decisions and not only through conscious/unconscious manipulation of the system.

- **AI training and operational phase, false dynamic learning:** a security risk exists if during the supply of further labelled data sets, incorrectly labelled data sets are deliberately stored, either by an attacker from outside or by an internal perpetrator. The aim here would also be to provoke false negatives in order to detect the manipulation as late as possible and thus maximise the damage.

- **AI operating phase, changed environment compared to learning environment:** in Example 1, the following changes in the environment compared to the learning environment could lead to both false positives and false negatives:

  - Modified surfaces of workpieces can lead to reflections which dazzle the image recognition.

  - The lighting conditions in production are not identical with the "learning laboratory".

In Example 3, the operational environment differs almost fundamentally from the laboratory environment (company-specific environments), so that wrong decisions of AI or a fixed rule are always present. These can be massively reduced in the course of learning.

5.1.3 Security risks when using dynamic AI systems

The security risks discussed in this chapter are, on the one hand, the same as the risks associated with the use of static AI systems, but either additional qualities of the risks discussed are added or additional risks may arise.

With reference to the above-mentioned reasons for wrong decisions, the safety risks are discussed with reference to Example 2 (collaborative robots) and partly to Example 3.

- **AI design and training phase, unclassified relevance of training data:** compared to closed systems, the limits of the various parameter values of an open system are not always completely pre-conceivable, the number of parameters and their value spectra are often larger, so that the coarse grids of the parameter value coverage in the training data are also necessarily larger. This means that the danger of not covering parameter value ranges increases significantly in open systems. Even a conscious manipulation of training data is very difficult to detect afterwards, as it becomes even more impossible for humans to assess why AI has decided in a certain direction. This is also due to the huge combinatorics of values than is the case with closed systems.

- **AI training phase, false training data generated by conscious manipulation of AI:** The same additional risks occur as with manipulative changes in the environment.

- **AI training phase, the ANN is overfitted/underfitted:** in the case of Example 2, the risk of overtraining and undertraining in productive operation is low, and thus also the risk of an attacker exploiting these aspects. Overtraining and undertraining of the system would already be detected during test operation, because the dynamics of the system would already lead to false reactions here, and relearning would have to be triggered directly.

- **AI training and operational phase, false dynamic learning:** the same scenario exists as for static/closed AI systems.

- **AI operating phase, changed environment in comparison to learning environment:** in example 2, the environment is massively changed by the dynamics of the system by definition. For example, further robots are used in production that did not exist at the time of the pre-training. There is therefore a constant danger that the AI has not been (pre-) trained generically enough to deal with the new situations, i.e. the risk of collisions increases, as shown in this example. A security risk arises in particular if machine data is manipulated in the productive environment or data from non-existent machines is generated and infiltrated. Therefore, both a provocation of false positives (impending accident is detected, although it

does not occur) and also false negatives (real impending accident is not detected) can occur, depending on which effects the attacker is targeting. Effect 1 can be a permanent fault due to false alarms, which can bring the system to a standstill. Effect 2 can be the provocation of a very big accident, which is not recognised and can lead to a major financial damage.

5.2 Security measures when using AI/security measures by AI

- The following applies to almost all technical systems: the volume of all possible inputs cannot be tested because it is too large.
- Therefore, deterministic systems test boundary values and extreme values, a defined series or sequence of test values as well as some random inputs (monkey test). Since the system can be assumed to have a well-defined behaviour, these tests can be used to deduce the correct behaviour for all permitted inputs (definition set).
- For ANN, this method does not have the same significance since ANN react to inputs with certain probabilities. These inputs usually vary solely due to the upstream sensor technology (e.g. the same image at different ambient conditions, light, temperature and noise of the image sensor).
- Use only pre-trained ANN models from trusted verified sources and verify their integrity.
- Use your own training data or only data from trusted verified sources.
- Check the behaviour of the ANN after each change to the system.
- Regularly check the behaviour of the ANN – if possible also unconditionally.
- In addition, behavioural samples can be verified by human employees. This is a proven strategy of very large IT service providers for testing and training of ANN - e.g. “Mechanical Turk”.

The ANN can also be used for security analysis. For this purpose, an ANN is trained in the “normal” behaviour of systems. This ANN then monitors the systems and reports anomalies. Successful ANN systems for the safety monitoring of IoT devices have been implemented as federated ANN. False positives, i.e. false alarms, must be avoided as far as possible in security monitoring. For the analysis of false positives, the ANN usually requires a correlation history between raw data, features and AI behaviour, so that the traceability of AI decisions is decisive for success.

6. Concluding remarks

This paper was written with the aim and claim of explaining the opportunities and risks arising from the use of Artificial Intelligence with respect to security aspects in Industrie 4.0. The aim of the team of authors was to present the state of the art, at least in general terms, and to give advice on practical implementation. The focus was on the modern aspects of AI, which have only been emerging for about five years, and in which it has become increasingly clear how great the importance of the explainability of AI has become. In this respect, the paper goes well beyond the first publication published in April 2019 [5].

As soon as the basic framework of machine learning is exceeded and influences come into focus, making it apparent that the great achievements of ML are associated with a loss of explainability, the need for competent advice is considerable. This paper can provide only a broad outline of the need for advice. However, it cannot be a substitute for qualified individual services. For many of the problem areas mentioned there are (still) no universal and at the same time simple answers. Further research is still necessary here. However, daily practice shows that it is possible to adequately address most of the problem areas referred to here. A risk assessment must be conducted on a case-by-case basis.

7. Basic terms of AI

Pre-trained systems are a technique to apply **transfer learning**. The layers at the upper edge of a neural network are removed and replaced by new layers with other classes. Only these new layers are trained, while the lower layers remain fixed. Modern deep nets, especially deep CNN for image recognition, build up a hierarchy of parameter values in their layers during their training – the automatically generated features in lower layers serve to recognise edges and other basic structures. They are used in higher layers to construct more complex features. Therefore, instead of redefining all weights in new training, it is an obvious approach to take over the lower layers of an already trained network and to redefine only the last layer(s). This process quickly achieves good results with comparatively little data. For example, metallic surfaces learned from ball bearings and crankshafts can serve as a basis for detecting screws that require few images of the respective screws.

The process of setting the free parameters of an AI system is called **training**. The distance between the estimated values provided by the system and the corresponding labels is iteratively reduced by adjusting these parameters, which are called weights in neural networks. The gradient of this distance, as a function of the weights, points in the direction of the steepest ascent. In order to improve the weights, steps must be taken in the direction of the native gradient. Under suitable conditions regarding the distance function, the **gradient descent** procedure converges towards zero. If a new common parameter set is calculated from the parameters of several AI systems, this is referred to as federated learning. **Federated learning** offers security and privacy benefits.

The assignment of training data to a class is caled **label**. In order to be able to calculate with labels, these are indicated in the so-called 1-hot representation. In the numbered classes, the label of class i is a vector that contains a 1 in component i and a 0 at all others. If the result vector of an AI system is normalised in such a way that all coordinates are positive and result in a sum of 1, then the respective values can be regarded as probabilities with which an input belongs to the corresponding classes. A correct estimate, i.e. the highest probability for the actual label of an element in the test set, is called a **hit** (according to **top-1** criterion). If the actual label is among the five highest probabilities, reference is made to a hit according to **top-5** criteria.

Visual perception is divided into **scene, image and object recognition**. Scene recognition focuses on images that do not contain any dominant objects, such as a marketplace, a harbour or a football match. Image recognition focuses on a representation with a single clearly dominant object: my car, my boat, my cat. Object recognition and localisation is aimed at representations that contain multiple elements from one or more known classes and mark them by framing them in a bounding rectangle. This could be, for example, an illustration of a road junction in which all pedestrians, cyclists, cars, buses, etc. are enclosed in labelled **“bounding boxes”. Face recognition** is a special case in which the features of a face are trained as independent classes and identified in a portrait with bounding boxes. It is then possible to make assignments to data stock from the relative positions in the portrait. Thus, in addition to the statement “that is a face”, a certain face or a member of a certain group can be identified.

In contrast to the first phase of AI (1956 to mid-1980s), the current focus (machine learning) is on the extraction of structural information from **data sets** with the aim of assigning data to terms or numerical quantities (classification, regression), identifying hidden characteristics or systematics, and developing strategies to achieve defined goals. The most important business part of AI today consists of the various variants of classification: for a previously unknown input signal from a defined spectrum (image, graphics, sound, word, text, video, temporal sequence of numerical values, ...), the AI system should make statements about which class of data the input signal is most likely to match. For example, what is shown on a 600X800 pixel RGB image? A car, a cat, a screw, a bump in the metal of a wing, ...? The ability to answer such questions accurately is acquired by AI systems through supervised learning. The basis for **supervised learning** is data that is provided with a **label**, i. e. an identification of the content. Starting from a division of the data into **classes**, in which all data elements per class carry the same label, the AI system “learns” to distinguish by the structure of an **approximation** function and to find correct assignments to the defined classes. Since both the learning process and the later recognition are pure arithmetic operations, all data must be converted into numbers. For example, a raster image can be represented as a matrix like a photo, whose coefficients are the triples of the RGB values of the colours of the pixels.

The labels are represented in the so-called **"1-hot" representation** as vectors, which have a 1 at the respective number of the class corresponding to the content and otherwise consist only of zeros. The system can count on this: if it generates an output vector that contains the probabilities that the input pattern corresponds to the respective classes with their numbers, then it can be compared with a vector that specifies 100 % for a certain class and 0 % for all others. The system can then be modified so that the distance between the own estimate and the label from the data set (the error of the estimate) becomes smaller. The **chain rule of the differential calculus** can be used to change all parameter values in the vector space of the **weights** of the error function in the direction of the negative gradient (**gradient descent**). This defines a way in which the error converges towards zero as a function of the weights in an iteration process and unknown labels are correctly estimated (**prediction**). The AI system does not therefore remember previously seen images and labels but has found a method of **generalisation**: it does not have to recognise the neighbour's dog but it has acquired the ability to say: "that is a dog with a 99 % probability".

In order to increase the amount of training data, a so-called **augmentation** is often performed: the individual examples are moved slightly to create additional data that match the same label. Rotation and the addition of noise are further possibilities. It is important not to change the relative frequencies in the data stock, otherwise additional biases will occur.

To carry out a training run with a data record that has already been prepared, it is first divided into three subsets: the training set (approx. 80%), the **validation set** and the **test set** (each approx. 10%). With stochastic order and grouping, all elements of the training set are used exactly once for the gradient descent – this is an **"epoch"** of training. After each such epoch, the average error on the validation set is determined. It can be observed that this error drops to a certain point, but then begins to rise again. At the same time, the error that can be detected in the training set continues to shrink. This means that the system goes on to remember the training data, but at the same time begins to lose the ability to generalise, i. e. to recognise the classes of unknown data. This state is called **overfitting**. It occurs particularly when the amount of data is small, but the number of parameters in the system is high. As soon as the error in the validation set increases, the training run is cancelled.

There are various heuristically selected parameters, the so-called **"hyperparameters"**, which influence the training run. The step size (**"learning rate"**) in the gradient descent and its change during iteration, the temporary fixation of the weights of individual neurons (**"dropout"**), the extent of periodically performed **regularisations** of weights and similar parameters influence the learning outcome. At the end, the errors of all determined variants are determined on the test set. The weights contained therein represent the final result of the training; the remaining **error on the test set** is the degree of precision achieved in the classification. It is necessary to split test and validation set to prevent the system from learning the hyperparameters. The system is only confronted with the test data at the very end of the process, so it is unknown until the end. The greater the number of hyperparameters, the greater the computational effort required for the training runs. This results in the often astonishingly high effort of calculating complex networks with many and wide layers as well as many possible combinations of hyperparameters.

8. References

[1] https://arxiv.org/pdf/1606.07461.pdf

[2] Image: https://pixabay.com/de/photos/katze-haustier-tier-grau-hauskatze-3261420/, (2015)
Copyright, image rights https://pixabay.com/de/service/terms/, (2019)

[3] https://arxiv.org/pdf/1605.07277.pdf

[4] Stuart Russell, Peter Norvig “Artificial Intelligence, A modern Approach”, Global Edition, (2018)

[5] Artificial Intelligence (AI) in Security Aspects of Industrie 4.0, Plattform Industrie 4.0, 2019

AUTHORS

Markus Heintel, Siemens AG | Dr. Detlef Houdeau, Infineon Technologies AG | Dr. Wolfgang Klasen, Siemens AG | Dr. Bernd Kosch (Leitung), Industrie-KI GmbH | Markus Ruppert, KOBIL Systems GmbH | Dr. Michael Schmitt, SAP SE | Thomas Walloschke, Fujitsu Technology Solutions GmbH | Dr. Thomas Wille, NXP Semiconductors Germany GmbH

This publication is a result of the sub-working group ‘Artificial Intelligence’ of the working group ‘Security of Networked Systems’ (Platform Industrie 4.0).

www.plattform-i40.de