ASSURANCE FOR MACHINE LEARNING SYSTEMS

Abstract

Machine Learning (ML) has become one of the fastest growing fields of science in recent years. The theories and technologies underpinning ML are used in a vast number of industrial and scientific applications, including banking, security, automotive and healthcare. Although it is becoming more prevalent to use ML technologies in safety critical settings such as healthcare and aerospace, there is still a significant need to assure safety of such systems. In this report, we will first take a look at the foundations of ML and safety. Next we review some of the methods currently developed for safety assurance in ML. We present these methods using a conceptual model based on four stages of ML: Data Management, Model Training, Model Verification and Model Deployment. We identify some of the open research areas associated with each stage.

Contents

List of Figures

List of Tables

Abbreviations

AI	Artificial Intelligence
ML	Machine Learning
AV	Autonomous Vehicle
ASIL	Automotive Safety Integrity Level
MDE	Model Driven Engineering
PDF	Probability Distribution Function
RL	Reinforcement Learning
GSN	Goal Structuring Notation
EDA	Exploratory Data Analysis
GAN	Generative Adversarial Network
iCM	Intuitive Certainty Measure

Chapter 1

Introduction

With new developments in Artificial Intelligence (AI) and ML, a growing number of research and development projects have started utilizing these methods. ML methods are also used in many safety critical applications such as Autonomous Vehicles (AVs) and healthcare applications. Therefore, it is very important to have a clear perspective of the safety of such methods in these applications, and the challenges or difficulties associated with assuring them.

In some applications, an erroneous outcome of the ML model can have a harmful impact, for example in medical diagnosis [27], loan approval [39], autonomous vehicles driving [36], and prison sentencing [9]. Despite the numerous research papers on this subject, there are still open questions and challenges, and still a need to delve deeper and understand the behavior of ML systems when applied in safety critical applications. There is also a need to better understand the risks associated with using ML in safety critical applications.

One major drawback in using ML algorithms is that they are often treated as a

black box and hence, using safety procedures for these methods is sometimes inapplicable or very difficult [48]. In a review of automotive software safety methods [47], an analysis of ISO-26262 part-6 methods was performed with respect to safety of ML models. This assessment shows that about 40% of typical software safety methods do not apply to ML models [47].

Safety specifications often assume that behavior of a component is fully specified. Since the training sets used in ML methods are not necessarily complete, they violate this assumption, and some parts of the specification becomes not applicable to the ML components [47]. Most widely used ML frameworks such as Tensorflow [42], Caffe [31], Pytorch [23] and Theano [54] employ a model driven approach in problem solving. Although model driven engineering approach has been successful in safety critical applications such as Automotive industry, the ML models cannot be guaranteed to operate in a safe manner.

There are two approaches with respect to ML and safety; the first is to study safety of ML methods, algorithms, and processes and the second is to use ML methods to improve pre-existing safety assurance procedures. We will initially follow the first approach and review the literature for the methods applied to standardize and measure the safety of ML methods.

There are inherent performance metrics related to ML methods, such as accuracy and robustness, which can affect their applicability in safety critical applications. ML models can also be dependent to the domain they are trained [28]. In addition, other perturbations such as noise, natural and imaging artifacts can cause ML models to function less accurately [30].

In this report we will first explore the basics of ML in Chapter 2. Then in Chapter

3 we review a definition of safety and how assurance cases are structured. Finally in Chapter 4 we survey the literature on ML assurance and identify some of the open challenges in this area.

Chapter 2

Machine Learning

In this chapter we will start by reviewing definitions of ML in the literature, and continue with definitions of learning categories such as supervised, unsupervised and reinforcement learning. Next, we explore how data is managed in a given ML problem, as this is relevant to safety, certification and assurance (for example, data might form some of the evidence used in an assurance case). Finally, we review how performance of ML methods is measured.

2.1 Definition

Machine learning algorithms can extract patterns and learn from data [18]. A brief definition of learning can be given as [43]

> "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

A task is the main objective of using an ML algorithm. For example, in an autonomous vehicle, driving the car is the task. A task is not the process of learning. Learning is used as a means to achieve an ability to accomplish a task [18]. With developments in ML methods, they have been applied to different tasks, some examples of tasks are classification, regression, transcription, machine translation, denoising [18].

The performance measure is used to quantify how successfully a task is accomplished, equivalently, number of erroneous outputs could be used as a way of indicating a method's performance.

Based on the above-stated definition, the ML algorithm undergoes an experience in the process of learning. This experience is generally classified into **unsupervised**, **supervised** and **reinforcement** learning.

2.2 Learning types

Unsupervised learning finds the properties of the overall structure of the dataset. Clustering as an example of unsupervised learning, finds clusters within a dataset and assigns each data-point to one of them.

In supervised learning, on the other hand, data-points that the learning algorithm experiences have a label. This label acts as a guide for the ML algorithm. The term supervised arises from the fact that the labels instruct the algorithm what to do. Labels are unavailable in unsupervised learning and the ML system is responsible to make sense of the data independently [18].

Reinforcement learning (RL) algorithms experience an environment instead of a fixed dataset. The algorithm should learn how to maximize a reward function by

taking an appropriate action [51]. The learner discovers this appropriate action by trying different actions and observing the value of the reward function. Actions not only affect the immediate reward, but can also change next actions' rewards. Trial and error search and delayed reward are two main characteristics of RL.
The learner, also known as the agent in RL terms, should have the capability to sense the state of the environment, take actions that can alter the state and also have a goal to reach by taking actions. These three aspects are included in the reward function used by the agent [51].

2.3 Data in ML

Evidently, ML algorithms need data to learn and function. A dataset can be described as a **design matrix**. Every row in the matrix contains an example, also known as data-point, and each column is a feature. Iris dataset is one of the first ones used in statistics and ML [24]. This dataset is comprised of 150 examples which have 4 features each. One example corresponds to one individual plant. Sepal length, sepal width, petal length and petal width are recorded as features of each plant [24]. This means that if X is the matrix, we can say $\mathbf{X} \in R^{150\times 4}$

The ML model will ultimately be deployed and used in a real world situation, hence, we are interested in how well an ML model performs on the data it has not seen before, this is also known as **generalization**. A portion of dataset is therefore not used in the training process and reserved as a **test set**. The data used in the training process is accordingly referred to as the **training set** [18].

In some cases, the training and test datasets might be limited in size and to have a better generalization, it is necessary to use as much of the data for training as

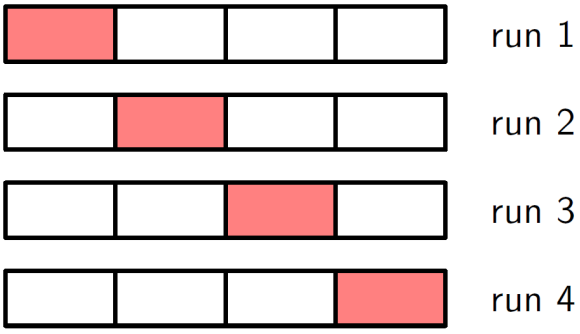

Figure 2.1: Cross validation for S=4 [12].

		Predicted Class		
		C-	C+	
True class	C-	Tn	Fp	Cn
	C+	Fn	Tp	Cp
		Rn	Rp	N

Table 2.1: A confusion matrix

possible. In other words, there will be less data available to estimate the performance of the model. One solution for this situation is **cross-validation**. The entire dataset is split into S subsets. In each run, $S-1$ subsets are used for training and one remaining subset is the test set. For the next run, a different test set is selected [12]. Figure 2.1 shows selection of subset for $S = 4$.

2.4 Performance measures

In classification tasks, *confusion matrix* is a way to demonstrate differences between predicted and true classes [15]. Table 2.1 shows the structure of a confusion matrix.

In Table 2.1 Tp and Tn represent true positives and true negatives respectively. Fp and Fn are in the same manner the count of false positives and false negatives

respectively. *Cp* and *Cn*, therefore, are the total number of positive and negative examples. Finally, *Rp* and *Rn* denote total number of predicted positives and negatives, respectively [15].

A variety of performance measures can be calculated from the confusion matrix, e.g., accuracy, precision, sensitivity and specificity [15]. Accuracy is often considered as a performance criteria which is simply the fraction of correctly classified samples to total samples, i.e., $\frac{Tp+Tn}{N}$. It is also possible to obtain the same information by calculating the *error rate*.

The Receiver Operating Characteristic (ROC) curve helps to find a pareto-optimal point between true and false positive rates as the decision threshold changes [15]. Each point on the curve represents the Tp (vertical axis) and Fp (horizontal axis) for a decision threshold. The area under ROC curve (AUC) is, therefore, a measure of the sensitivity of the model to changes in operating conditions. If AUC value is at maximum, i.e., one, it can be concluded that the $P(Fp) = 0$ and $P(Tp) = 1$ even when the operating conditions change.

According to the mandates of the ML application, various performance metrics can be calculated from the confusion matrix, [50] introduces twenty four of these metrics and in which applications they can be used.

2.5 Neural networks

In this section we briefly review the structure of neural networks and their building blocks. As the name suggests, neural networks are fundamentally a collection of entities called *neurons*, which can hold small units of data. Here we consider a simple neural network, depicted in Figure 2.2, consisting of a hidden layer and an output

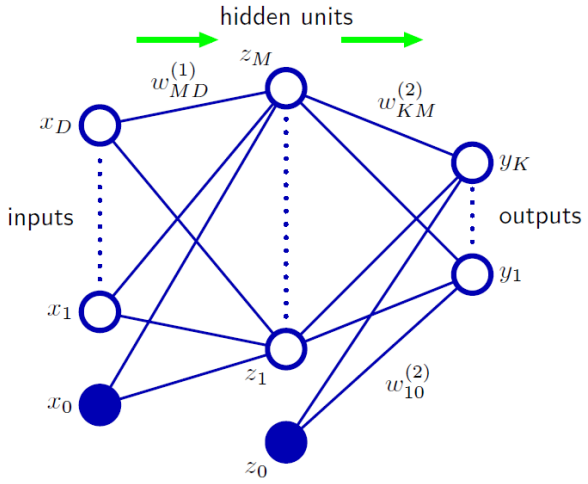

Figure 2.2: Structure of a two layer neural network. [12].

layer. Hidden layers are the ones not directly accessible from the outside world, i.e., not the input or output layers.

Using the notation in [12], if we have D input variables, x_i, we can calculate linear combinations a_j such that

$$a_j = \sum_{i=1}^{D} w^{(1)}_{ji} x_i + w^{(1)}_{j0}$$

Here, $1 \leq j \leq M$ where M represents number of neurons (nodes) in the hidden layer. The superscript (1) corresponds to the first layer of the network, i.e., the hidden layer in this example. $w^{(1)}_{ji}$ are called weights and $w^{(1)}_{j0}$ are biases for the hidden layer. The a_j values are referred to as *activations*. Next, we use activation function, h, to calculate z_j, such that

$$z_j = h(a_j)$$

The activation function is required to be differentiable due to the differentiation in the learning process. z_j values are then used to compute the linear combinations a_k

in a similar manner to the previous layer, i.e.,

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)}$$

a_k is the output layer activation, therefore, similar to the hidden layer, an activation function, σ, will be used to reach final output values of the network, y_k.

$$y_k = \sigma(a_k)$$

Substituting values from all layers the final y_k will be

$$y_k(x, w) = \sigma\Big(\sum_{j=0}^{M} w_{kj}^{(2)} h\Big(\sum_{i=0}^{D} w_{ji}^{(1)} x_i \Big) \Big)$$

2.6 Trends in ML research

With the rapid development of frameworks and libraries in ML, it is increasingly straightforward to apply ML algorithms and tools in different applications. It is possible to develop smaller models which in turn makes applications such as IoT [32] more feasible. Prevalence of ML raises important questions in its fairness [17], privacy [22], and safety [55]. The term *responsible ML* covers the social impacts of using ML in everyday life. Moreover, the desire to have faster, better performing ML systems has lead to areas such as quantum ML [10]. As a matter of fact, quantum ML has been one of the initial motivations for experiments which lead to the famous quantum supremacy [21].

Chapter 3

Safety and assurance

In this chapter, we start with a basic definition of safety to set the context. Next, we delve into (safety) assurance cases and how they are structured. Finally, we review a representation method for assurance cases called Goal Structuring Notation (GSN).

3.1 Definitions

3.1.1 Safety

Safety is defined in ISO 26262 [26] as:

> Absence of unreasonable risk.

An unreasonable risk is a [26]:

> Risk judged to be unacceptable in a certain context according to valid societal moral concepts.

Various safety standards have been developed for different industries and activities. Some examples are ISO 26262 for functional safety of road vehicles, DO-178C for

aerospace industry, ISO 8124 for safety of toys, ISO 7164 for healthcare organization management.

3.1.2 Assurance

Assurance is defined in ISO 15026 to be [3]

> Grounds for justified confidence that a claim has been or will be achieved.

Assurance is, therefore, the grounds on which the users of a system can rely on its functionality. It is specially important for systems with complexity, such as ML, to give assurance to the users before they start utilization. The level of this assurance is closely related to the level of dependence or trust needed from the users' side. Adequate evidence and arguments need to be present to justify the safety and reliability of the system. The basis for this justification is achieved with reducing uncertainty in measurements, observations, estimations, predictions, information, inferences or effects of unknowns [3].

3.2 Safety Assurance Case

Assurance cases have been successfully used in various industries to specify an argument as to why a system can be safely used for a specific application in a specific context [7]. A recent definition of safety assurance case is described in [13] as

> "A structured argument, supported by a body of evidence, that provides a compelling, comprehensible and valid case that a system is safe for a given application in a given environment"

A structured argument is a [44]

> "connected series of statements or reasons intended to establish a position...; a process of reasoning."

Reasons used in a structured argument can be considered as premises in logical terms and a conclusion can be drawn based on them [44]. The purpose of using an assurance case is to communicate a clear, comprehensive, defensible argument that a system is safe to be used in a particular context [34]. Assurance cases are comprised of five basic components: claims, arguments, evidence, justifications and assumptions. The most common use of assurance cases is to give assurance about system's functionality and properties to the parties which were not involved in the process of developing the system [3].

Assurance cases are used to explicate and support reasoning, albeit in a subjective manner, especially when compared to the logical proofs which consider an absolute truth. In other words, assurance cases are useful because the full range of a system's properties are not always representable in a logical formalization. Also, assurance cases may sometimes be disproved because the underlying logical theory used in them is not relevant [3].

Since assurance cases are considered artefacts, they inherit quality related properties of them such as: the structure of its content, semantic features such as completeness, creation and maintenance. The conclusions of the assurance case should also be stated clearly with clear level of uncertainty [3].

> For hazards associated with warnings, the assumptions of [7] Section 3.4 associated with the requirement to present a warning when no equipment failure has occurred are carried forward. In particular, with respect to hazard 17 in section 5.7 [4] that for test operation, operating limits will need to be introduced to protect against the hazard, whilst further data is gathered to determine the extent of the problem.

Figure 3.1: Problems associated with textual representation [34].

3.3 Goal Structuring Notation

When the safety assurance case is more complex in nature, textual representation suffers to express the case in a clear and understandable way. Figure 3.1 shows an example of such problem where the English structure of the argument is hard to understand. Having multiple cross references is specially difficult to capture in text [34].

The Goal Structuring Notation (GSN) is a graphical notation for safety argumentation. A GSN specification explicitly represents elements of a safety argument and the relationships among these components. For example, how requirements are supported by claims or how claims are supported by evidence or how the case has a defined context [34]. Figure 3.2 depicts basic building blocks of a GSN with example instances of each element.

3.4 An example of a GSN

The goal structure is used to show how goals (claims about the system) can be split into sub-goals successively until the sub-goal can be directly supported by available evidence. Figure 3.3 represents an example of a GSN.

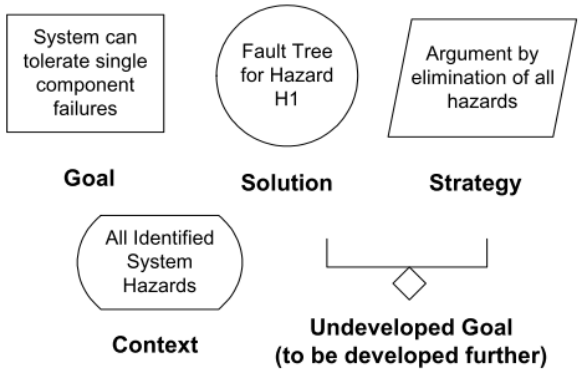

Figure 3.2: Basic elements of a GSN [34].

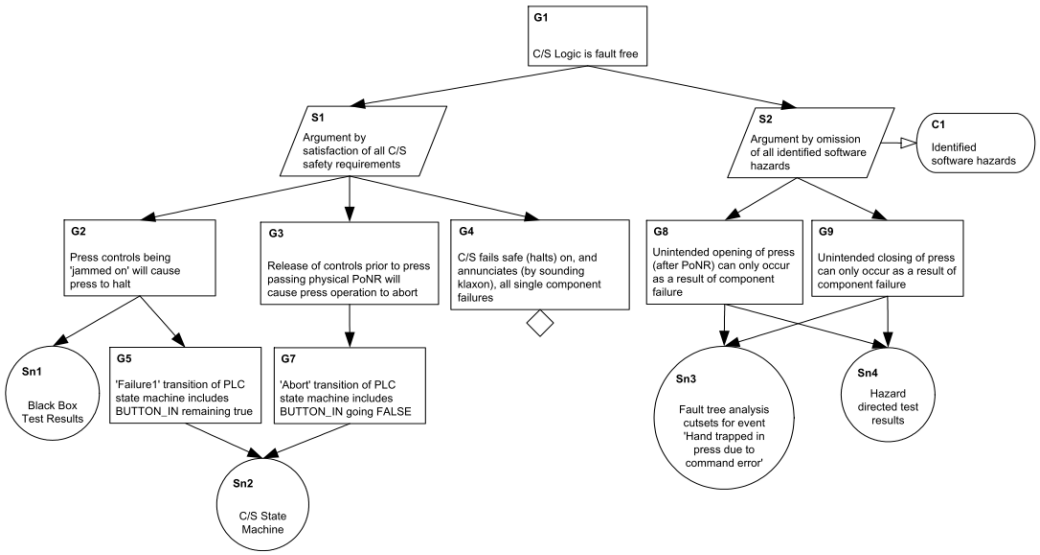

Figure 3.3: An example of a goal structure [34].

In this example, "Control System (C/S) logic is fault free." is one single top level goal. The main goal is then divided to two sub-goals through strategies $S1$ and $S2$. These two strategies are then supported by five sub-goals $G2 - G4$ and $G8 - G9$. In a goal structure, there will be a stage where the sub-goals can be directly supported by solutions. In this example, sub-goals $G8 - G9$ are supported by $Sn3 - Sn4$ and there is no need to break down the goals further in this branch [34].

3.5 Trends in safety research

Safety assurance should constantly evolve and adapt to the new paradigms in industry. With advent of Industry 4.0, new risks and challengers arise in workspace and occupational safety. Safety 4.0 is a response to these new challenges [38]. There is a growing number of research projects on using the recent advances in ML technologies to enhance safety, some of which is referred to as "safety informatics [56]". In addition, 5G technology has raised concerns about the long term health consequences. Critiques are collecting evidence that 5G may result in skin cancer and the millimeter wave radiation can ultimately affect the nervous system [46]. Unmanned Aerial Vehicles (UAV), or commonly known as drones, are rapidly spreading in industrial usage and thus their safety and privacy challenges are of interest [8].

Chapter 4

Literature Review

In this chapter we will briefly review some of the literature about the safety of ML methods and identify major research questions in this area. Today, ML is used in variety of applications with varying safety requirements. This diverse portfolio includes smart phones [5], cars [40], surgical equipment [20], construction industry [11] and many more. A fault in an ML system, e.g. a misclassification, has different repercussions in each application. An out of focus image taken by a camera can be easily remedied, but a malfunction in a surgical equipment could be fatal and result in an irreversible situation. Therefore, assuring safety is an essential part of the design process for these applications.

One major issue in safety assurance for ML is guaranteeing that the training data is complete and relevant. Data used in the operational stage is by definition relevant, however, training data may not reflect all possible situations that the learning algorithm needs to be exposed to. On the other hand, the operational environment may change and the training data may diminish in relevancy. As discussed in detail later in this chapter, assuring completeness and relevancy is challenging and may not

always be feasible.

In [19], five major research problems associated with unsafe behavior of ML models is presented. They can be summarized as

1. Avoiding Negative Side Effects: How to ensure that the model will not disturb the environment while pursuing its goals, e.g. can a cleaning robot knock over a vase because it can clean faster by doing so? Can we do this without manually specifying everything the robot should not disturb.

2. Avoiding Reward Hacking: How to ensure that the model does not avoid situations to achieve a higher reward. For example, if we reward the robot for achieving an environment free of messes, it might disable its vision so that it won't find any messes, or cover over messes with materials it can't see through, or simply hide when humans are around so they can't tell it about new types of messes.

3. Scalable Oversight: How to ensure the model respects the parts of the objective function that are expensive to evaluate and makes a safe approximation of these parts. For example, in the cleaning robot example, if the user is happy with the cleaning quality is an expensive objective function, but it can be approximated to presence of any dirt on the floor when the user arrives.

4. Safe Exploration: How to ensure that the ML model explorations are safe. For example, the robot should experiment with mopping strategies, but putting a wet mop in an electrical outlet is a very bad idea.

5. Robustness to Distributional Shift: How to ensure that the model performs robustly if the environment shifts from the training environment. For example,

strategies a cleaning robot learns for cleaning an office might be dangerous on a factory work-floor.

4.1 Machine Learning lifecycle

To obtain assurance for ML systems it is essential to understand the ML lifecycle and how to analyze safety in each step. In this section we will first introduce these steps and review some of the safety measures for each step. This lifecycle follows a spiral process model, i.e., the stages are iteratively repeated to actively reduce risk [14]. ML lifecycle is comprised of four stages [7]

4.1.1 Data Management

This stage involves collecting, preprocessing, augmenting and initial analysis of data. The training and validation datasets are also prepared in this step. From assurance perspective, the data collected in this step should be

- Relevant: The dataset should be relevant to the desired functionality of the final model. For example a dataset of handwritten letters in Japanese language cannot be used for English language. In many cases, a pre-existing dataset will be used to train the model. This dataset should be obtained from trusted sources with an encrypted transmission medium. An attacker can add irrelevant samples into the training dataset to make the final model behave in a specified way for particular inputs [16]. The irrelevant data injected in the dataset is called a backdoor [16] Despite early efforts in detecting backdoors in face recognition [56] finding a general solution is still an open challenge.

- Complete: The features of a dataset should not have unintended correlations that can confuse the classifier. For example, if a classifier is trained on pictures of wolves and huskies, and all wolves have snow in the background, it may be concluded that snow in the background means a wolf [45]. In this case the dataset is not complete because it does not include pictures of wolves with different backgrounds. Exploratory Data Analysis (EDA), is an important step in examining completeness of a dataset. It is also important to identify how the data points are distributed in the input sample space, measures such as gap ratio [53] can be used to evaluate uniformity. To further increase completeness of the dataset, Generative Adversarial Networks (GAN)s can be used to learn the distribution of the classes and generate more data [6].

- Balanced: For classification problems, it can happen that one class has significantly more data-points in the training set than the others and thus, classifier has more exposure to that class. In addition, imbalance can also happen in features, for example, if age is one of the features and data is collected from primary schools in a city, an imbalance in age feature is expected. The impact of feature imbalance on final models' behavior is an open challenge.

- Accurate: This property considers factors like sensor accuracy, correctness of data collection and processing method. In the case of supervised learning, labels' accuracy is important. In many applications it is needed to manually label data. Various crowdsourcing platforms is developed for this purpose however, accuracy of the labels is debatable and should be investigated subsequently [49]. The data collection process should be documented to identify potential inaccuracies [7].

4.1.2 Model Learning

In this stage of the ML lifecycle, the type of the model and its hyper-parameters are selected. For some ML applications, the dataset is very large or the model structure is complex and therefore, the learning process needs considerable amount of computational power. In these cases, it is reasonable to take advantage of a previously trained model and adapt it to our needs by re-training only the parts that are different from the other application. This process is called *transfer learning* [18]. If there is a need to do transfer learning, it will be decided at this stage and finally the learning process starts using the train dataset obtained in previous stage. In order to have a clear view of the model in safety related aspects, the final model should be [7]

- Performant: As a requirement for a safer model, it should have a justifiable performance according to the measures introduced in Chapter 2. Judging model's performance solely on one composite metric, is sometimes inaccurate and not contextually relevant [25]. While measures such as accuracy represent a general idea of model's performance by averaging over all outputs, they lack information on outcomes for a specific example. Intuitive Certainty Measure (ICM) provides an estimate of model's performance for a specific example, based on the errors it made in the past [56].

- Robust: The model should be able to perform as well on the unseen data as the training data, i.e., generalizes well to be considered robust. One solution to increase robustness of the model is to enhance completeness of the dataset, therefore, data augmentation data augmentation is one of the techniques to increase generalization [35]. GANs as discussed in Data Management section help in completeness of the dataset and, consequently, increasing models' robustness

[6].

- Reusable: Using transfer learning can help to use the assurance evidence of the original model, provided that the transfer learning is performed in the right context for the source and destination models. However, reusing models comes with the risk that safety issues propagates to the destination model too [29]. Models trained on popular standard datasets can be found in model zoos such as [2][1].

- Interpretable: This property shows how much the decisions made by the model are explainable and thus helps to analyze the safety of such decisions. Natural language explanations [37], model visualization [41], explaining by example [4]

4.1.3 Model Verification and Validation

The black swan problem expounds one of the major challenges in validating ML models. A system or a person could incorrectly conclude from abundance of training data samples that common observations are true [36]. A model which has only seen white swans, may infer that all swans are white and ignore the fact that there are black swans [52]. One major challenge is thus making sure that the model works well, i.e., satisfies its requirements, on the data it has not seen before which is also known as generalization. If the model fails in this stage, the process will go back to Data Management or Model Learning steps. Model verification involves requirements encoding, test-based verification and formal verification. The verification stage should be [7]

- Comprehensive: Model verification should ensure that all the requirements of

the system and also intended goals of the previous stages of ML lifecycle, i.e., data management and model learning, are covered.

- Contextually relevant: Verification process should be relevant to the intended use of the ML model. For example an ML model used in autonomous vehicles, we are more concerned about how changes in the environment will affect model's performance and thus, how robust is the model with changes in weather rather than the changes in image quality.

- Comprehensible: Verification results should be understandable for the users. Requirement violations should be clearly expressed in such a way that the cause of it can be identified and fixed [7]. Ideally, the results should also include any black swan biases present in the model [36].

4.1.4 Model Deployment

Preparing the ML model to be used in the final application. Activities in stage includes integration, monitoring and updating. To assure safety of this stage of ML lifecycle, the ML model should have the following properties

- Fit-for-Purpose: The difference in hardware can cause performance differences between ML stages. Also, each distinct hardware setting where a model is deployed can affect model's performance. For a model to be fit for purpose, the performance seen in the previous stages should be carried over to the deployment phase.

- Tolerable: The system should be able to tolerate occasional incorrect outputs of the ML model. To accommodate this, the host system should be able to

identify the incorrect outputs and to replace them with a safe value so that the system continues the normal processing activities.

- Adaptable: Deployed models are in many cases needed to be updated due to variety of reasons including operational, legislative or environmental changes. This property indicates how safe is the process of updating.

4.2 Open challenges in ML assurance

Using an ML component in a system poses several challenges in each step of the ML lifecycle. In the data management step, further research is needed to guarantee security of data and its fitness for the purpose. Although a vast amount of research has been conducted in the model learning stage, there is still a need to further study hyper-parameter selection. In addition, with recent successes in transfer learning, there is still need for more research in assuring safety in this area. Furthermore, safety assurance requires ML models to be reusable and interpretable. Model verification assurance is mainly accomplished using test-based and verifications. However, there is still a need to develop methods to encode model requirements into proper and formal tests. In model deployment stage, there is no explicit equivalent for updating models in software engineering world, therefore, there is a need to devise assurance methods for adaptable safety-critical systems [7].

In some applications requirements for a safe ML system reinforce each other. For example, accuracy in data management stage will most likely result in more performant model. However, in some cases, there is a trade-off between requirements, an explainable model is probably more exposable to cyberattack [7]. In spite of

attempts to address this issue [33], more research is required to adapt these concepts to ML.

Chapter 5

Conclusion

In this report we first started with the basidc definitions and principals of machine learning and safety assurance concept and eplored some of the fundamentals in both areas in Chapter 2 and Chapter 3 we also glanced currently trending research in these areas. Finally, in Chapter 4 we reviewed some of the literature in assurance of ML systems and some of the open challenges and research questions in each step of ML lifecycle. Despite significant research into some stages of ML lifecycle, more research is needed in other parts. For example, explainable models, there are no global methods providing insights into complex ML models.

Bibliography

[1] Caffe — Model Zoo. http://caffe.berkeleyvision.org/model_zoo.html.

[2] Collection: Model Zoos of machine and deep learning technologies. https://github.com/collections/ai-model-zoos.

[3] 15026-1-2019 - ISO/IEC/IEEE International Standard - Systems and software engineering–Systems and software assurance –Part 1:Concepts and vocabulary. *ISO/IEC/IEEE 15026-1:2019(E)*, pages 1–38, 2019.

[4] Ajaya Adhikari, D. M. J Tax, Riccardo Satta, and Matthias Fath. LEAFAGE: Example-based and Feature importance-based Explanationsfor Black-box ML models. *IEEE Int. Conf. Fuzzy Syst.*, 2019-June, dec 2018.

[5] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L. Reyes-Ortiz. Human Activity Recognition on Smartphones Using a Multiclass Hardware-Friendly Support Vector Machine. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 7657 LNCS:216–223, 2012.

[6] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data Augmentation Generative Adversarial Networks. nov 2017.

[7] Rob Ashmore, Radu Calinescu, and Colin Paterson. Assuring the Machine Learning Lifecycle. *ACM Comput. Surv.*, 54(5):1–39, may 2021.

[8] Burchan Aydin. Public acceptance of drones: Knowledge, attitudes, and practice. *Technol. Soc.*, 59:101180, nov 2019.

[9] Richard Berk and Jordan Hyatt. Machine Learning Forecasts of Risk to Inform Sentencing Decisions. *Source Fed. Sentencing Report.*, 27(4):222–228, 2015.

[10] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, sep 2017.

[11] Muhammad Bilal and Lukumon O. Oyedele. Guidelines for applied machine learning in construction industry—A case of profit margins estimation. *Adv. Eng. Informatics*, 43:101013, jan 2020.

[12] C M Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.

[13] Robin Bloomfield and Peter Bishop. Safety and assurance cases: Past, present and possible future - An adelard perspective. In *Mak. Syst. Safer - Proc. 18th Safety-Critical Syst. Symp. SSS 2010*, pages 51–67. Springer London, 2010.

[14] Barry Boehm and Wilfred J Hansen. Spiral Development: Experience, Principles, and Refinements Spiral Development Workshop February 9, 2000. Technical report, 2000.

[15] Andrew E Bradley. The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms. *Pattern Recognit.*, 30(7):1145–1159, 1997.

[16] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning. Technical report.

[17] Sam Corbett-Davies and Sharad Goel. The Measure and Mismeasure of Fairness: A Critical Review of Fair Machine Learning. jul 2018.

[18] Ian Goodfellow Courville, Yoshua Bengio, and Aaron. *Deep learning*, volume 29. MIT Press, 2016.

[19] Dario Amodei et al. Concrete Problems in AI Safety. jun 2016.

[20] Melissa Egert, James E. Steward, and Chandru P. Sundaram. Machine Learning and Artificial Intelligence in Surgical Fields. *Indian J. Surg. Oncol. 2020 114*, 11(4):573–577, jul 2020.

[21] Frank et al. Arute. Quantum supremacy using a programmable superconducting processor. *Nat. 2019 5747779*, 574(7779):505–510, oct 2019.

[22] Peter et al. Kairouz. Advances and Open Problems in Federated Learning. *Found. Trends® Mach. Learn.*, 14(1–2):1–210, dec 2019.

[23] Adam et al. Paszke. PyTorch: An Imperative Style, High-Performance Deep Learning Library. Technical report, 2019.

[24] R. A. Fisher. The Use of Multiple Measurments in Taxonomic Problems. *Ann. Eugen.*, 7(2):179–188, sep 1936.

[25] Peter Flach. Performance Evaluation in Machine Learning: The Good, the Bad, the Ugly, and the Way Forward. *Proc. AAAI Conf. Artif. Intell.*, 33(01):9808–9814, jul 2019.

[26] International Organization for Standardization. *ISO 26262: Road Vehicles : Functional Safety*. ISO, 2018.

[27] Kenneth R Foster, Robert Koprowski, and Joseph D Skufca. Machine learning, medical diagnosis, and biomedical engineering research-commentary. Technical report, 2014.

[28] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *32nd Int. Conf. Mach. Learn. ICML 2015*, 2:1180–1189, sep 2015.

[29] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. aug 2017.

[30] Dan Hendrycks and Thomas Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. *arXiv*, mar 2019.

[31] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. *MM 2014 - Proc. 2014 ACM Conf. Multimed.*, pages 675–678, jun 2014.

[32] Mansi Jindal, Jatin Gupta, and Bharat Bhushan. Machine learning methods for IoT and their Future Applications. *Proc. - 2019 Int. Conf. Comput. Commun. Intell. Syst. ICCCIS 2019*, 2019-January:430–434, oct 2019.

[33] Nikita Johnson and Tim Kelly. Devil's in the Detail: Through-Life Safety and Security Co-assurance Using SSAF. In *Comput. Safety, Reliab. Secur.*, pages 299–314, Cham, 2019. Springer International Publishing.

[34] Tim Kelly and Rob Weaver. The Goal Structuring Notation-A Safety Argument Notation.

[35] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. Audio Augmentation for Speech Recognition.

[36] Philip Koopman and Michael Wagner. Challenges in Autonomous Vehicle Testing and Validation. Technical report, 2016.

[37] Samantha Krening, Brent Harrison, Karen M Feigh, Charles Isbell, Mark Riedl, and Andrea Thomaz. Learning from Explanations using Sentiment and Advice in RL.

[38] Vendula Laciok, Katerina Sikorova, Bruno Fabiano, and Ales Bernatik. Trends and Opportunities of Tertiary Education in Safety Engineering Moving towards Safety 4.0. 2021.

[39] Stefan Lessmann, Bart Baesens, Hsin-Vonn Seow, and Lyn C Thomas. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *Eur. J. Oper. Res.*, 247:124–136, 2015.

[40] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J. Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, Michael Sokolsky, Ganymed Stanek, David Stavens, Alex Teichman, Moritz Werling, and Sebastian Thrun. Towards fully autonomous driving: Systems and algorithms. In *IEEE Intell. Veh. Symp. Proc.*, pages 163–168, 2011.

[41] Aravindh Mahendran and Andrea Vedaldi. Understanding Deep Image Representations by Inverting Them. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 07-12-June-2015:5188–5196, nov 2014.

[42] Martín Abadi et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. www.tensorflow.org, 2015.

[43] T M Mitchell. *Machine Learning*. McGraw-Hill international editions - computer science series. McGraw-Hill Education, 1997.

[44] Object Management Group. Structured Assurance Case Metamodel (SACM). https://www.omg.org/spec/SACM/2.1/PDF, 2010.

[45] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?" Explaining the Predictions of Any Classifier. In *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, New York, NY, USA. ACM.

[46] Cindy L. Russell. 5 G wireless telecommunications expansion: Public health and environmental implications. *Environ. Res.*, 165:484–495, aug 2018.

[47] Rick Salay, Rodrigo Queiroz, and Krzysztof Czarnecki. An Analysis of ISO 26262: Using Machine Learning Safely in Automotive Software, sep 2017.

[48] Gesina Schwalbe and Martin Schels. A Survey on Methods for the Safety Assurance of Machine Learning Based Systems. 2020.

[49] Victor S. Sheng and Jing Zhang. Machine Learning with Crowdsourcing: A Brief Summary of the Past Research and Future Directions. *Proc. AAAI Conf. Artif. Intell.*, 33(01):9837–9843, jul 2019.

[50] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.*, 45:427–437, 2009.

[51] Richard S Sutton. Introduction: The challenge of reinforcement learning. In *Reinf. Learn.*, pages 1–3. Springer, 1992.

[52] Nassim Taleb. *The black swan : the impact of the highly improbable*. Random House, New York, 2007.

[53] Sachio Teramoto, Tetsuo Asano, Naoki Katoh, and Benjamin Doerr. Inserting Points Uniformly at Every Instance.

[54] The Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. https://github.com/Theano/Theano, 2016.

[55] Kush R. Varshney. Engineering safety in machine learning. In *2016 Inf. Theory Appl. Work. ITA 2016*. Institute of Electrical and Electronics Engineers Inc., mar 2017.

[56] Bing Wang and Chao Wu. Safety informatics as a new, promising and sustainable area of safety science in the information age. 2019.