Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 118 (2023) 32–37

www.elsevier.com/locate/procedia

16th CIRP Conference on Intelligent Computation in Manufacturing Engineering, CIRP ICME '22, Italy

Assessment Framework for Deployability of Machine Learning Models in Production

Henrik Heymann[a,*], Hendrik Mende[a], Maik Frye[a], Robert H. Schmitt[a,b]

[a]*Fraunhofer Institute for Production Technology IPT, Steinbachstr. 17, Aachen 52074, Germany*

[b]*Laboratory for Machine Tools and Production Engineering WZL of RWTH Aachen University, Campus-Boulevard 30, Aachen 52074, Germany*

* Corresponding author. Tel.: +49-241-8904-478; *E-mail address:* henrik.heymann@ipt.fraunhofer.de

Abstract

Deploying machine learning (ML) models in production environments comes with challenges such as the model's integration into live production and the missing trust of process experts in new technologies. These challenges must be addressed already in phases ahead of the deployment. Therefore, this paper aims to clarify how to ensure the deployability of methods used during model development. For this purpose, criteria for measuring and evaluating deployability in manufacturing environments are defined. A subsequent analysis of existing data preprocessing methods and ML algorithms regarding deployability as well as deployment options serves to counteract deployment issues early on in an ML project.

Keywords: Artifical Intelligence; Machine Learning; Deployment; Deployability; Production; Manufacturing

1. Introduction

Machine learning (ML) as a subset of artificial intelligence (AI) finds application in production in the form of many different use cases, all of which have domain-specific challenges [1]. There is a major expectation towards the performance and broad deployment of ML across all areas and industries [2]. However, ML-based systems are not immune to failure. Especially in critical decisions, where reliability is a crucial concern. Hence, there is a necessity for robust and reliable ML systems, which may only fail in a predictable and contained fashion [3].

ML models are developed under carefully controlled conditions to obtain the best possible performance. In real, productive environments, where models are ultimately deployed, conditions are rarely perfect, resulting in a discrepancy between development and deployment [4]. This is due to gaps in the ML workflow during the transition from the development to the deployment of models, which we call development-deployment-discrepancy. Many deployments are realized unsuccessfully due to diverse challenges, which may be unique to the respective domain [5]. Consequently, the phase of deployment in a project is crucial and requires careful planning.

First and foremost, the expectations and challenges of deployment need to be addressed early on. In project phases before the deployment, activities and methods need to be assessed regarding their effect on the deployability of the ML system, which is nowadays often neglected, and which requires the consideration of domain specific conditions from production. This leads to the following research questions (RQ):

- RQ1: What is deployability of ML in production?
- RQ2: How can deployability be assessed?

Therefore, the goal of this work is to develop an assessment framework for deployability of ML models in production in

2212-8271
10.1016/j.procir.2023.06.007

order to close the development-deployment-discrepancy. The framework supports users, equipped with a use case and an initial model to evaluate the set-up regarding its suitability for deployment.

The remainder of this document is structured as follows. Section 2 contains the state of the art focusing on deployability in the context of software and ML. The methodology is introduced in section 3, while the results are presented in section 4. Section 5 concludes by providing a summary of the results and an outlook on future research.

2. State of the art

In this chapter, a review of existing approaches regarding deployability is conducted. Deployability as a concept is located at the verge between software and ML. It can be applied to the process of deploying software and ML model. Therefore, each perspective is handled separately. The chapter ends with an interim conclusion.

2.1. Deployability in the context of software

Pillai defines the deployability of a software system from a practical perspective as "the ease with which it can be taken from development to production. It can be measured in terms of the effort in terms of man-hours, or complexity in terms of the number of disparate steps required for deploying code from a development to production environment." [6] In the context of deployment time, as the time to get code into production, deployability can be understood as "a new quality attribute for complex systems" [7]. *Bellomo et al.* [8] also define deployability as a quality attribute, which assesses how well requirements are achieved. They focus on design decisions regarding the software architecture to enable deployability. Goals of deployability are to enable the build and continuous integration, test automation, deployment and robust operations, as well as synchronized and flexible environments. *Schaefer et al.* [9] underline the importance of environment consistency for deployability.

2.2. Deployability in the context of ML

Deployability can also be analyzed in the context of general AI applications or with a more technical focus on ML. As for the former aspect of a more general view, the *MIT Center for Deployable Machine Learning* [2] proposes to create AI systems that are robust to a variety of random and adversarial corruptions, safe for real-world deployment, and enable reliable decision-making. Furthermore, the systems need to be understandable and easy to work with for humans with or without ML expertise. *Liao et al.* [10] state that the actual deployability of a model depends on four aspects: (1) the prediction performance of the model; (2) the robustness of the model; (3) the dependence of the model on external data; and (4) the storage size of the model. Further technical aspects of deployability include the limitation of the deployment by device type [11] and measuring the deployability of models by measuring their run time performance [12].

Deloitte's AI Qualify offering [3] aims at creating robust AI by putting AI reliability to the test. ML models are subjected to a series of tests, examining the resilience, reliability, stability and attack vector vulnerability. Evaluations in all dimensions are aggregated to an overall measure of robustness, while tracking robustness versus predictive power. The *Fraunhofer IAIS* [13] presents priorities for the certification and trustworthy use of AI. In addition to ethics and law, the audit areas include autonomy and control, fairness, transparency, reliability, security and data protection.

Deployability is closely linked to challenges and obstacles, which appear in context of the deployment. *Baier et al.* [14] present challenges in the deployment and operation of ML in practice by conducting a literature review including interviews. Identified challenges are grouped by pre-deployment, deployment, and non-technical. Similar challenges are then clustered into data structure, implementation, infrastructure, governance, customer relation, and economic implications. *Paleyes et al.* [15] identify challenges in deploying ML through a survey of case studies. The ML workflow is divided into deployment stages (data management, model learning, model verification, model deployment, and cross-cutting aspects). For each stage, definitions of deployment steps are provided in combination with the respective considerations, issues and concerns. *Ployzotis et al.* [16] present challenges in production. Given the focus on building ML models, the lessons learned during data management challenges (data understanding; data validation and cleaning; data preparation) include realistic assumptions about the availability and non-centralized storage of data, the diverse needs of different users who interact with ML systems, and the smooth integration of data management tools into development workflow. *Nestor et al.* [17] present caveats to deployable model performance in common clinical ML tasks. They address the goal of identifying and overcoming obstacles – mainly the sustainment of performance over time – to effective model deployment. AI development and deployment in general can be accelerated by using pre-trained models in form of transfer learning. Transfer learning is defined as the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned. As transfer learning holds the advantage of performance stability by achieving better results in comparison to train-from-scratch models, organizations can save efforts during an ML project [18].

2.3. Interim conclusion

Deployability from a software perspective is understood in the context of continuous integration and continuous delivery for complex software systems. When it comes to deployability of ML, different aspects are associated with deployability, and either from a more general or a more technical perspective. Similar concepts may be treated without any reference to deployability. Existing approaches analyze the challenges for deployment through case studies and literature review, but do not allow a direct evaluation of existing methods. Especially domain-specific considerations, e.g., from production, are not covered sufficiently.

3. Methodology

Based on RQ1 and RQ2 as well as the findings from literature, the methodology to develop the assessment framework for deployability of ML models in production consists of two steps.

First, existing concepts are aggregated into one unified definition. Then, dimensions of deployability are defined including the specification of how each dimension is evaluated. The dimensions are associated to one of the four steps – data integration, data preparation (DPP), modeling, deployment – in the ML pipeline in production [19].

The proposed deployability assessment with the help of the framework is conducted after finishing modeling and before starting the deployment to close the development-deployment-discrepancy. This enables the evaluation of methods used in previous steps of the ML pipeline with regard to their suitability for the planned deployment. Besides that, the framework also provides criteria to evaluate the deployment itself. It allows to make a statement about the success of the realized deployment. The framework's development was supported by participating partners in a publicly funded research project, who provided data and input for the textual elaboration.

4. Results

Following the methodology, deployability is defined in a first step. Subsequently, the different dimensions of deployability along the ML pipeline in production are introduced including an individual assessment approach for each dimension.

4.1. Definition of deployability

For the purpose of the development of the framework, we define deployability of ML as a quality criterion indicating how well the requirements towards the deployment are met. In comparison to software deployment, the deployment of ML is more complex. Instead of bringing code to a production environment, deploying ML models has the goal of making a resulting model available in a specific environment in order to make the results usable where they are needed. Therefore, deployability of ML is limited neither to the time from development to production nor to finding the best performing model. Rather, deployability is about achieving an overall goal of ML deployment as defined before. Consequently, deployability is understood as an umbrella term for dimensions of successful deployment, which allow the evaluation of methods regarding their suitability. Given the individual requirements for each use case, the evaluation is use case dependent. However, the dimensions of the evaluation are valid in general. Therefore, an assessment framework is developed, which can be applied in multiple scenarios. The framework aims at bringing the dimensions into the deciders' awareness and to develop a deployment mindset, which helps with achieving the ultimate goal of any ML application in the industry, which is to create the highest business value. By this definition, RQ1 can positively be answered.

4.2. Dimensions of deployability

In the following, the dimensions in each of the areas from data integration to deployment are provided including a description and the evaluation schema. The evaluation schema highly depends on the dimension. Some dimensions may be measured directly, others are assessed qualitatively.

Data integration: Deployability dimensions associated with data integration are summarized in Table 1. *Environment consistency* describes the reproducibility between training and deployment infrastructure [9]. It is measured by the degree of overlap of data sources and the used software. Looking at the *volume of data*, the incoming data flow in training and serving are measured with regard to their data size.

Table 1. Data Integration.

Dimension	Description	Evaluation
Environment consistency	Reproducibility between training and deployment infrastructure	Degree of overlap between environments
Volume of data	Incoming data for training and serving	Size of incoming data flows

Data preparation: Table 2 subsumes the dimensions with regard to DPP. First, the *performance* of the DPP method plays a crucial role in any ML project and thus for deployment. Based on data quality checks such as the number of missing values or outliers, the performance of DPP methods can be measured. In addition, the performance can also be determined by assessing the performance of ML algorithms (see modeling). Another dimension is *robustness*, which is characterized by the repeatability, stability, and dependence on the randomness of the DPP method. It is measured through changes in performance or based on sensitivity analyses. The ability to understand the functionality of the chosen DPP method is further of great importance. *Explainability* can be measured based on the underlying complexity of the DPP method.

Table 2. Data Preparation.

Dimension	Description	Evaluation
Performance	Performance of DPP method	Performance metrics
Robustness	Repeatability and stability of DPP method	Performance changes
Explainability	Ability to understand the DPP method's functionality	Explainability metrics
Impact	Degree of data set's manipulation	Data set characteristics
Computing time	Execution time for running DPP method	Latencies

The degree of how the data set is manipulated is covered by the *impact*. Data set characteristics can be measured before and after applying DPP methods providing insights to the degree of manipulation. The manipulation directly affects the noise being present in the data and is interlinked with performance and explainability. Lastly, the computing of DPP methods is considered, which represents the execution time for running the DPP method and is measured via latencies.

Modeling: Deployability dimensions for modeling are given in Table 3. The first dimension in modeling is *performance*, which describes the predictive power of a model. Suitable metrics depend on the underlying ML task, e.g., accuracy, precision, recall, F1-score for classification and mean squared error, root mean squared error, mean absolute error for regression. [20]

Robustness (alternatively reliability, adaptivity or stability) describes a model's behavior towards outliers and noise in the input data. It allows to make a statement of the robustness regarding corruption and drift by indicating how perturbed an ML algorithm is by small changes to its inputs. In case of classification, robustness can be measured by observing the mean shift of the accuracy and the variation of the accuracy when juxtaposing clean and noisy data [21].

Safety, also called adversarial robustness, resilience, or attack vector vulnerability, is related to *robustness* and analyzes the behavior of a model in case of adversarial input. In malware detection applications, perturbations in the input are crafted by an adversary in order to reduce the model's performance [22]. In contrast to *robustness*, the *safety* dimension focuses on intentionally manipulated inputs. However, it is measured in the same way as *robustness* by means of performance metrics. In industrial environments, there can be adversarial attacks on industrial control systems, which make ML cybersecurity defenses necessary [23].

As experts need to trust and understand the model's actions, the *explainability* dimension evaluates how well the ability of comprehending and understanding the decision process within the model, i.e., the trustworthiness, is given. There are user-focused metrics which aim to capture *explainability* aspects such as the goodness of explanations, user satisfaction or trust in a human-AI system [24]. From a technical perspective, the SHAP (SHapley Additive exPlanations) framework, for instance, assigns each feature an importance value for a particular prediction so that experts can interpret them [25].

From a software-related perspective, the dimension of *model run time* and *storage size* refer to the computing time for executing model training and serving, respectively the storage space of the model artifact. Time is measured by latencies, and storage space in usage of local or cloud storage. [10, 12]

From an organizational point of view, the dimension of *modeling effort* captures the cost and time spent in the model building task. The dimension can be evaluated by looking at six levels of AutoML ranging from no automation to full automation. [26]

Another form of reducing cost and time is to re-use existing models for other domains or tasks. This is referred to as *transfer learning*. Deployability in the context of AI-scaling – the organization-wide deployment of AI / ML solutions – benefits heavily from using pre-trained models. To measure the success of transfer learning, the performance graph over training time can be analyzed regarding a higher start, higher slope and higher asymptote in comparison to training from scratch. [18]

How the dimensions are weighted against each other depends on the scenario and individual requirements. There are multiple trade-offs between the dimensions. Mainly between *performance* and *robustness* as well as between *performance* and *explainability*. A robust model, which still performs adequately in a worst-case scenario, will not achieve the absolute overall best *performance*. How *robustness* is weighted depends on the area of application with stronger requirements towards robustness in life-critical applications. Similarly, the highest accuracy is often achieved by complex models, which are very difficult to interpret even for experts. Some dimensions, such as *safety*, are often overlooked in production scenarios. Manufacturing companies experience less sabotage of production data as in comparison to intentional corruption of social media algorithms for example. Aspects regarding hardware limitations or *modeling effort* are more important to smaller companies, which may lack the resources of a large enterprise.

Table 3. Dimensions for Modeling.

Dimension	Description	Evaluation
Performance	Predictive power of model	Performance metrics
Robustness	Behavior towards outliers and noise	Performance changes
Safety	Behavior of model in case of adversarial input	Performance changes
Explainability	Ability to understand model actions	Explainability metrics
Model run time	Computing time for running model training and serving	Latencies
Model storage size	Storage space of model artifact	Storage space
Modeling effort	Cost and time of modeling task	Levels of AutoML
Transfer learning	Re-use of model for other domains or tasks	Performance gains

Deployment: The dimensions summarized in Table 4 serve to analyze the deployability of an ML model, focusing on the deployment itself.

Time restrictions refer to the integration of the deployed ML model into the production processes without causing disturbances to the operational procedures. In some use cases, results may be necessary within a time frame of a few hours, e.g., for personnel checking anomalies provided by ML models on previous day data. However, specifically for real time applications results may need to be available within a few milliseconds. Measuring the process time in production into which the deployment will be made reveals time restrictions to which the model serving will need to adhere to.

Added value for user is created if the information content is sufficient so that the task can be executed in the desired manner. Here, the quality and quantity of the provided information and its interpretability by the user is in the focus. *Value for user* can be measured in two ways. One would be to measure the impact on overall objectives since deployment of the model in changes to the productivity, quality, ecological and economic efficiency, or workload. The second way is to measure the conditional entropy or amount of information provided [27].

The *deployment cost* of the solution refers to the expenditures made through the use of personnel, software, hardware, training, maintenance and upkeep, as well as certification procedures. A review of cost and time expenditure

throughout all stages of the deployment and all personnel involved is necessary.

An assessment of the technical *complexity* of the solution is necessary for determining its deployability, since it has effects on the maintainability, scalability, and the deployment cost. We propose to measure two indicators to determine the complexity. First, the processing frequency of the solution depends on the frequency with which incoming data needs to be handled. Second, the dimensionality of the model is to be measured consisting of both the type of components needed, like data ingestion, data preparation, models for inference, or post-processing, and in the number of modules for those components.

The *maturity* of the solution has an impact on the deployability due to different implications on the effort necessary for deployment and upkeep. A prototype may only run in a closed-off, controlled environment, whereas a complete system may be integrated into the productive environment. An assessment of the maturity can be made with the technology readiness level (TRL) method [28].

The deployment and its upkeep must be executed in a quick, reliable manner. Continuous integration and continuous delivery as well as respective tools offer certain levels of automation to the process. Measuring the CI/CD *pipeline automation* levels can be done with maturity models. Examples are the Continuous Delivery 3.0 Maturity Model [29], the MLOps levels [30] or the Machine Learning operations maturity model [31].

Table 4. Dimensions for Deployment.

Dimension	Description	Evaluation
Time restrictions	Integration into the production processes	Process time
Value for user	Information content available to user	Impact and conditional entropy
Deployment cost	Overall cost of solution	Cost and time expenditures
Complexity	Technical complexity of solution	Processing frequency and dimensionality of model
Maturity	Technical maturity of deployment	TRL
Pipeline automation	Efficiency of continued deployment	CI/CD maturity models
Scalability	Scaling of users, models, requests, and other criteria	Measures of cohesion and coupling
Cyber security	Security of data and application	PSRC model

Scalability refers to the possibility of transferring and duplicating the solution to other production sites and processes, as well as multiplying the solution itself. Regarding the first aspect of scalability, the models themselves but also the preceding DPP can be transferred to other processes, for example. Regarding the second aspect, models can be duplicated and parallelized within the solution. To measure scalability, the modularity of the solution plays an important role. According to *Dorbecker et al.* [32], it can be measured by means of cohesion "how closely related the components of a module are" [33] and coupling "how connected/disconnected each module is from each of the other modules" [34].

Another important aspect is *cyber security*. Attacks on the solution can have serious consequences for production. The Fraunhofer IPT's PSRC model can be used to assess cyber security [35]. The model is based on established models such as the Cybersecurity Capability Maturity Model (C2M2) and important cybersecurity standards known from industry.

4.3. Application of framework

Fig. 1 answers RQ2 by summarizing the identified deployability dimensions and by associating them with the respective step in the ML pipeline in production. Deployability is broken down and its assessment can be executed along the given workflow following the given structure.

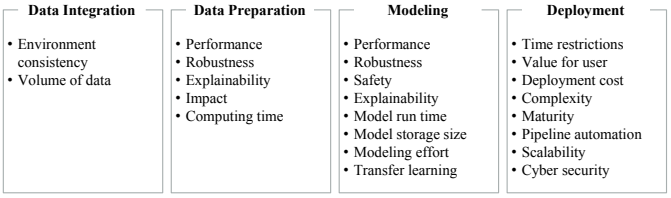

Fig. 1. Deployability dimensions along ML pipeline steps in production.

The framework containing an evaluation method for each dimension can be applied in industrial use cases to assess the deployability. For a holistic deployment approach it can be combined with the guideline for deployment of ML [36]. Use cases for future validation include predictive maintenance, predictive quality and process parameter setting in 3D printing. In each use case, different options are compared and evaluated in the provided dimension with the help of the introduced evaluation techniques. Options as input for the assessment consist of ML algorithms, software tools or further methods that are related to the deployment of ML. Based on the deployability assessment results, more suitable options can be selected and implemented in the organization to ensure an effective as well as efficient deployment of ML.

5. Conclusion

In this work, an assessment framework was developed, which enables an evaluation of the deployability of ML models. For this purpose, deployability in the context of production was defined and relevant dimensions for deployment along the ML life cycle were introduced. The framework can be used to assess and compare possible options by means of an individual evaluation method for each dimension.

Limiting factors include the fact that only an evaluation of the current situation without any hints on possible actions to improve the evaluation can be realized within the defined dimensions. Furthermore, a validation of the derived framework with a real-life use case is pending.

There are two lines of future investigation. First, it is necessary to work towards a more complete approach covering possible counteractions. Second, it is to be investigated, how deployability can be analyzed on a higher level, early on in a

project. Already during the use case selection, deployment-related aspects such as hardware limitations in the existing infrastructure as well as the organizational processes, maturity and expertise need to be considered.

Acknowledgements

The research was conducted within the “AI-gent3D – AI-supported, generative 3D-Printing” project. The German partners of this research and development project are funded by the German Federal Ministry of Education and Research (BMBF) within the “Innovations for Tomorrow's Production, Service and Work” Program with the funding reference 02P20A500 and implemented by the Project Management Agency Karlsruhe (PTKA). The author is responsible for the content of this publication.

References

[1] Wuest T, Weimer D, Irgens C, Thoben K-D. Machine learning in manufacturing: advantages, challenges, and applications. Production & Manufacturing Research 2016; 4(1): 23–45 [https://doi.org/10.1080/21693277.2016.1192517]
[2] MIT Center for Deployable Machine Learning. CDML. Available from: URL: https://cdml.mit.edu/.
[3] Deloitte. Putting AI Reliability to the Test: Deloitte’s AI Qualify Offering for Robust AI; 2021.
[4] Chen P-Y. Securing AI systems with adversarial robustness: IBM. Available from: URL: https://research.ibm.com/blog/securing-ai-workflows-with-adversarial-robustness.
[5] Weiner J. Why AI/Data science projects fail: How to avoid project pitfalls. San Rafael, California: Morgan & Claypool Publishers 2021.
[6] Pillai AB. Software architecture with Python: Design and architect highly scalable, robust, clean, and high performance applications in Python. Birmingham: Packt April 2017.
[7] Mistrik I, Soley RM, Ali N. Software Quality Assurance: In Large Scale and Complex Software-intensive Systems. 1. Aufl. s.l.: Elsevier Reference Monographs 2015.
[8] Bellomo S, Ernst N, Nord R, Kazman R. Toward Design Decisions to Enable Deployability Empirical Study of Three Projects Reaching for the Continuous Delivery Holy Grail. In: Toward Design Decisions to Enable Deployability Empirical Study of Three Projects Reaching for the Continuous Delivery Holy Grail; 2014. New York: IEEE; 702–7.
[9] Schaefer A, Reichenbach M, Fey D. Continuous Integration and Automation for Devops. In: Kim, Haeng Kon, Ao S-I, Rieger, Burghard B., editors. IAENG Transactions on Engineering Technologies. Springer Netherlands 2013; 345–58.
[10] Liao Z, Pan H, Fan X, Zhang Y, Kuang L. Multiple Wavelet Convolutional Neural Network for Short-Term Load Forecasting. IEEE Internet Things J. 2021; 8(12): 9730–9 [https://doi.org/10.1109/JIOT.2020.3026733]
[11] Perego R, Candelieri A, Archetti F, Pau D. Tuning Deep Neural Network’s Hyperparameters Constrained to Deployability on Tiny Systems. In: Farkaš I, Masulli P, Wermter S, editors. Artificial Neural Networks and Machine Learning – ICANN 2020. Cham: Springer International Publishing 2020; 92–103.
[12] Adam G, Chitalia V, Simha N, *et al.* Robustness and Deployability of Deep Object Detectors in Autonomous Driving. In: Robustness and Deployability of Deep Object Detectors in Autonomous Driving; 2019. New York: IEEE; 4128–33.
[13] Cremers AB, Englander A, Gabriel M, *et al.* Trustworthy Use of Artificial Intelligence: Priorities from a Philosophical, Ethical, Legal, and Technological Viewpoint as a Basis for Certification of Artificial Intelligence. Sankt Augustin; 2019.
[14] Baier L, Jöhren F, Seebacher S. Challenges in the Deployment and Operation of Machine Learning in Practice. In: Johannesson P, Ågerfalk P, Helms R, editors. Challenges in the Deployment and Operation of Machine Learning in Practice; 2019.
[15] Paleyes A, Urma R-G, Lawrence ND. Challenges in Deploying Machine Learning: a Survey of Case Studies. arXiv; 2020.
[16] Polyzotis N, Roy S, Whang SE, Zinkevich M. Data Lifecycle Challenges in Production Machine Learning: A Survey. SIGMOD Record 2018; Vol. 47(No. 2).
[17] Nestor B, McDermott MBA, Boag W, *et al.* Feature Robustness in Non-stationary Health Records: Caveats to Deployable Model Performance in Common Clinical Machine Learning Tasks. arXiv; 2019.
[18] Torrey L, Shavlik J. Transfer Learning. In: Soria Olivas E, editor. Handbook of research on machine learning applications and trends: Algorithms, methods, and techniques. Hershey, Pa: IGI Global (701 E. Chocolate Avenue Hershey Pennsylvania 17033 USA) 2010; 265–76.
[19] Frye M, Krauß J, Schmitt RH. Expert System for the Machine Learning Pipeline in Manufacturing. IFAC-PapersOnLine 2021; 54(1): 128–33 [https://doi.org/10.1016/j.ifacol.2021.08.014]
[20] Sarkar D, Bali R, Sharma T. Practical Machine Learning with Python: A Problem-Solver's Guide to Building Real-World Intelligent Systems. Berkeley, CA: Apress 2018.
[21] Buzhinsky I, Nerinovsky A, Tripakis S. Metrics and methods for robustness evaluation of neural networks with generative models; 2020 Mar 4.
[22] Grosse K, Papernot N, Manoharan P, Backes M, McDaniel P. Adversarial Examples for Malware Detection. In: Foley SN, Gollmann D, Snekkenes E, editors. Computer security - ESORICS 2017: 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017 : proceedings. Cham: Springer 2017; 62–79.
[23] Anthi E, Williams L, Rhode M, Burnap P, Wedgbury A. Adversarial attacks on machine learning cybersecurity defences in Industrial Control Systems. Journal of Information Security and Applications 2021; 58: 102717 [https://doi.org/10.1016/j.jisa.2020.102717]
[24] Hoffman RR, Mueller ST, Klein G, Litman J. Metrics for Explainable AI: Challenges and Prospects. arXiv; 2018.
[25] Lundberg S, Lee S-I. A Unified Approach to Interpreting Model Predictions; 2017.
[26] Tunguz B. Six Levels of Auto ML; 2020 [cited 2022 April 8] Available from: URL: https://medium.com/@tunguz/six-levels-of-auto-ml-a277aa1f0f38.
[27] MacKay DJC, Mac Kay DJC. Information theory, inference and learning algorithms. Cambridge university press 2003.
[28] Mankins JC. Technology readiness levels. White Paper, April 1995; 6(1995): 1995.
[29] Jan Vlietland. Continuous Delivery 3.0 Maturity Model (CD3M); 2019 [cited 2022 April 23] Available from: URL: https://nisi.nl/continuousdelivery/articles/maturity-model.
[30] Garg S, Pundir P, Rathee G, Gupta PK, Garg S, Ahlawat S. On Continuous Integration/Continuous Delivery for Automated Deployment of Machine Learning Models using MLOps. In: On Continuous Integration/Continuous Delivery for Automated Deployment of Machine Learning Models using MLOps. IEEE; 25–8.
[31] Microsoft. Machine Learning operations maturity model [cited 2022 April 23] Available from: URL: https://docs.microsoft.com/en-us/azure/architecture/example-scenario/mlops/mlops-maturity-model.
[32] Dorbecker R, Bohm D, Bohmann T. Measuring Modularity and Related Effects for Services, Products, Networks, and Software -- A Comparative Literature Review and a Research Agenda for Service Modularity. In: Measuring Modularity and Related Effects for Services, Products, Networks, and Software -- A Comparative Literature Review and a Research Agenda for Service Modularity; 12015. IEEE; 1360–9.
[33] Al-Dallal J. Qualitative Analysis for the Impact of Accounting for Special Methods in Object-Oriented Class Cohesion Measurement. J. Softw. 2013; 8(2): 327–36.
[34] Stryker AC. Development of measures to assess product modularity and reconfigurability. Air Force Institute of Technology 2010.
[35] Kiesel R. Cybersecurity in Networked Production. Fraunhofer-Gesellschaft; 2021.
[36] Heymann H, Kies AD, Frye M, Schmitt RH, Boza A. Guideline for Deployment of Machine Learning Models for Predictive Quality in Production. Procedia CIRP 2022; 107: 815–20 [https://doi.org/10.1016/j.procir.2022.05.068]