A Framework for Applying Data-Driven AI/ML Models in Reliability

Rok Hribar, Margarita Antoniou, and Gregor Papa

Abstract In this chapter, we present a framework for applying artificial intelligence (AI)/machine learning (ML) in reliability, in the context of the iRel40 project. Data-driven models are becoming an increasingly fruitful tool for detecting patterns in complex data and identifying the circumstances in which they occur. Using only data, gathered along the value chain, data-driven methods are now being used to detect indications of potential early failures, signs of wear-out or degradation, and other unwanted events within the development, fabrication, or service phases of the electronic components and systems. We present general considerations that were found to be important during iRel40 project, when designing pipelines that combine data processing with the AI/ML models for predicting or detecting reliability issues. This chapter serves as an introduction to the definitions and concepts used within the specific use cases that rely on the AI/ML methodology within the iRel40 project.

1 Overview

Reliability is an important topic in electronics because it affects a number of efficiency performance metrics, such as production costs, maintenance costs, and spare parts management costs, which ultimately affect productivity and revenue. Low reliability implies that machines have frequent breakdowns and thus frequent interruptions in manufacturing processes. Additionally, reliability is a crucial factor in modern industries, as it directly affects customer satisfaction and business strategy. The occurrence of unexpected product failures can result in customer dismay and negatively impact a company's reputation. Therefore, ensuring the reliability of electronic components and systems is essential for maintaining customer satisfaction

Rok Hribar, Margarita Antoniou, and Gregor Papa
Jožef Stefan Institute, e-mail: rok.hribar@ijs.si, margarita.antoniou@ijs.si, gregor.papa@ijs.si

Rok Hribar, Margarita Antoniou, and Gregor Papa
Jožef Stefan International Postgraduate School

and competitiveness in the market. Furthermore, reliability is becoming a key enabler for successful product designs, and big data can be used to understand failure mechanisms, usage scenarios, and optimal designs.

Artificial intelligence (AI)/machine learning (ML) models play an important role in improving reliability by enabling real-time monitoring and predictive maintenance of electronic components and systems. By analyzing large amounts of data generated by sensors and other sources, AI/ML models can identify patterns and even predict potential failures before they occur. This allows for proactive maintenance and repair, reducing downtime and improving overall system reliability. Additionally, AI/ML models can be used to optimize manufacturing processes, improve product quality, and reduce costs. This is why the development of the AI / ML methodology was one of the main building blocks of the iRel40 project. The goal of the iRel40 project was to improve reliability by reducing the failure rates of electronic components and systems throughout the value chain of Electronic Components and Systems (ECS) in Europe. The main objective of the project was to improve the reliability of ECS by developing new materials, designs, and testing methods, as well as implementing real-time feedback on production lines. This is why the project aimed to leverage AI/ML models to enable real-time monitoring and predictive maintenance. The project also aimed to contribute to the enhancement of knowledge in the field of reliability of ECS by aligning research results with real production insights and data.

This chapter serves as the foundational introduction to the iRel40 framework, a key outcome of our project aimed at improving industrial reliability through data-driven AI/ML methodologies. In the following pages, we will define the key concepts and components of the framework, enabling readers to grasp the subsequent practical use cases.

2 Reliability Challenges and AI/ML

In recent times, AI and ML techniques have been embraced by industry, advertising, healthcare, banking, security, and others due to their improved accuracy and efficiency. This is in addition to growing confidence in these approaches. Data-driven modeling is useful when the problem cannot be explained theoretically, but there is a lot of data that contains rules and patterns related to the phenomena that control the system. AI and ML can be used to uncover rules and patterns that can be used to approximately solve a problem in a more or less understandable manner.

> Machine Learning (ML) is a collection of techniques used to identify patterns in data and use those patterns to make predictions or decisions. The data are analyzed to uncover trends and relationships, which can then be used to make inferences and predictions, or to make decisions in uncertain situations.

The objective of ML is to make computers gain knowledge from experience, most notably by mining the available data, in order to solve practical problems. Instead of directly programming how a computer should solve a given problem, the structure of the desired program is found using an optimization algorithm. For this to work, a database is required that is used to train a ML model alongside with its expected behavior when exposed with a given data. With such a database, the ML algorithm can be trained to find and recognize patterns in the input data that are relevant for a given task in mind. For example, if a model is trained to estimate the remaining useful life of an electronic device, it learns to find patterns in the sensor data that were observed to happen in a given time span before failure. As more data become available, the algorithms perform better and are more accurate. ML is being applied in many different sectors, and each of them have its own particularities and specialized techniques that go along with it.

In current industrial practice, there are several tasks where ML has found its place and was proven to be efficient and reliable. ML allows companies to automate a wide variety of tasks that are labor intensive and error-prone when performed by humans [1]. Another ideal task for ML is the situation where the data volume required to make an adequate decision is beyond human capacity [12], due to a very high number of features or due to the speed with which the relevant data are generated. Usually used to forecast results or identify and comprehend patterns, ML is most helpful with structured or labeled data. In particular, machine learning is effective in organizing and analyzing data such as videos [9], photos [14], and audio [10] due to its simplicity in classifying and evaluating this type of data. When a goal value cannot be predicted using straightforward calculations or rules, or when dealing with very large data sets, ML becomes especially useful [3].

Data-driven modeling is just one part of the effort to improve reliability. As any other methodology ML has its advantages and drawbacks and it is not uncommon to augment ML with more classical approaches, such as expert systems or physics-based models, in order to join the strengths of different methodologies. An important aspect of data-driven modeling is its dependence on data quality. Specifically, in reliability-related problems, the gathered data on which the ML is trained can be incomplete and unbalanced. For example, in failure detection, the data are usually not comprehensive enough to cover all types of failure and the intricate physical processes that are pertinent. We usually operate with data that include only some modes of failure (incomplete data), and the frequency of anomalous operation is orders of magnitude less likely compared to normal operation (unbalanced data). In such cases, sole data-driven modeling can provide only limited insight. However, in the iRel40 project, many approaches have been applied and developed to overcome these problems, as well as being practically tested in real-world use cases. Examples of these approaches range from smarter data acquisition (tailored for AI applications) to more intricate model development with customized regularizations and training procedures. The ultimate goal of our future work is to find a unified way to combine the physics of failure models with data-driven ones. Experience from iRel40 shows that, when this is possible, the data-driven models require much less data to train since its knowledge does not come only from patterns extracted from data, but also from our knowledge of the underlying physics. Because data acquisition in reliability is a very expensive part of ML pipeline, finding a way to reduce data quantity would allow application of data-driven modeling in ways that are just not possible in the current setting.

3 The iRel40 Framework

Throughout the iRel40 project, various approaches that incorporated AI in some way or another were developed. These techniques range from general ones that were applied to several use cases to specialized methods specific to a unique problem setting. In this section, we present the general characteristics of these methods and how they relate to the properties of the underlying problem and the available training data. It is essential to keep in mind that it is not possible to provide a precise formula for the use of AI in reliability, as the effectiveness of the methods and techniques used also depends on more subtle features of each specific use case. To provide a universal formula for how to use AI would, in essence, solve the problem of general artificial intelligence, which is still unresolved. However, we can present some guidelines for the AI/ML usage framework given the experience acquired during the iRel40 project.

To better understand how AI/ML can be applied in reliability, it is essential to first recognize what are the different kinds of settings in which AI/ML is usually used in reliability. Figure 1 provides a visual representation of different settings encountered in iRel40 use cases in terms of the characteristics of the problem to be solved, available data, and the AI methods used to address it. The upper part of the figure can be seen as a taxonomy for the use cases, which can be used to arrange them in a hierarchical structure. The top part of the figure illustrates the relationship among use cases in terms of the underlying problem and their data properties, while the bottom part focuses on the methodology used in a given setting and how they can be grouped into major ML approaches. The definitions of terms found in Figure 1 are summarized in Table 1. A more comprehensive discussion of each layer of the categorization is given in the following paragraphs.

3.1 Reliability Aspects

3.1.1 Problem Type: Prognostics vs Diagnostics

The complexity of modern engineering systems, such as industrial processes, aircraft, road vehicles, manufacturing systems, electrical and electronic equipment, presents challenges in terms of their reliability, safety, and performance. To address these issues, **diagnostics and prognostics methods** must be developed to be used in real-world conditions. Diagnostics tasks focus on fault detection, fault isolation, and fault identification [6]. Fault detection is the process of recognizing when something

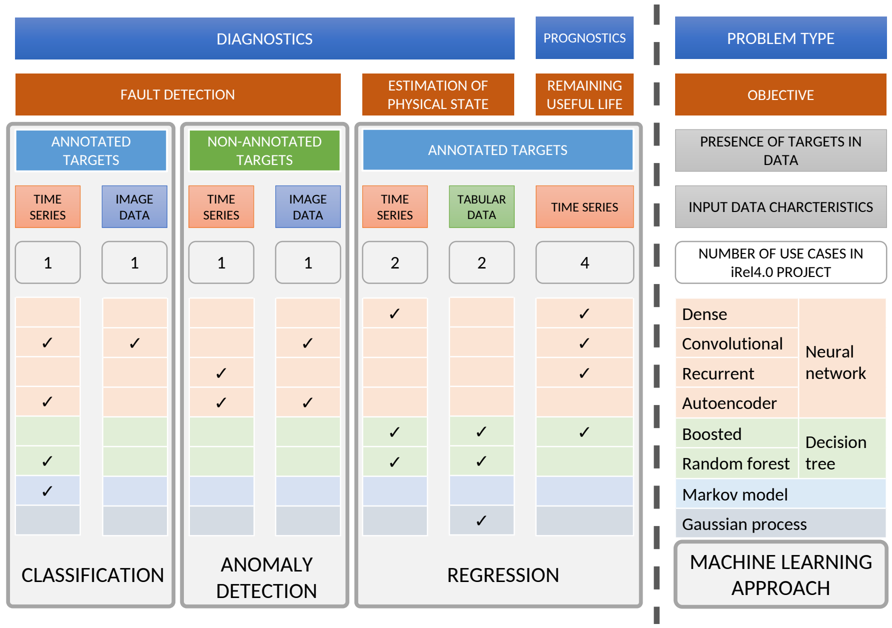

Fig. 1 Schematic representation of how use cases (that employed AI/ML as a base methodology) in iRel40 project relate to one-another with respect to the characteristics of the problem being solved, the available data and to the applied methodology in the scope of AI algorithms.

is wrong in the monitored system, fault isolation determines which fault candidate is causing abnormal behavior, and fault identification determines the magnitude and type of the fault [2]. Prognostics, on the other hand, attempts to predict the likelihood of a system failing based on its current state and diagnostics. The remaining useful life (RUL) is the main type of prediction in prognostics, which is the time left before the end of useful life or before the health state of a system crosses a failure threshold [7]. Additionally, prognostics can also be used to estimate the probability that a system will operate without failure until a specific future time, given its current health condition. The core difference between the two problem types is that in diagnostics we estimate a current state while in prognostics we estimate a future state. This distinction is not just philosophical in nature but has implications to what methods are used, especially regarding data gathering and processing, as well as the methodology for model evaluation and training. The advantages of diagnostics and prognostics for complex systems are numerous, including increased system safety and operational reliability, improved and optimized system design and development, and reduced life-cycle cost and increased revenue. A comprehensive review of diagnostic and prognostic approaches for complex engineered systems, discussing their strengths, weaknesses, and applicability, and highlighting unaddressed challenges using automotive systems as an example, can be found in [13].

Table 1 Definitions of terms found in Figure 1.

Problem Type	Diagnostics	Models are estimating the current state of a device or process.
	Prognostics	Models are estimating a future state of a device or a process.
Objective	Fault Detection	Models predict whether a device or a process is faulty in some manner and possibly pinpoint the type of fault and its location.
	Estimation of Physical State	Models estimate some properties of a device or a process that are usually difficult to measure in real-world situation.
	Remaining Useful Life	Models are predicting the length of time a machine is likely to operate before it requires repair or replacement.
Presence of Targets in Data	Annotated Targets	Model predictions are known, i.e., targets are annotated in the data set.
	Non-Annotated Targets	Model predictions are not known, i.e., targets are non-annotated in the data set.
Input Data Characteristics	Tabular Data	Input data has no (known) topological structure.
	Time Series	Input data instance is composed of a meaningfully ordered sequence of vectors.
	Image Data	Input data instance is a set of vectors with rectangular two-dimensional topology.

3.1.2 Reliability Assessment Methods

The second level addresses a broad objective of what we are trying to achieve with AI/ML. Here, we recognize three general categories for the use cases in the iRel40 project. Those are Fault Detection, Estimation of Physical State, and Remaining Useful Life Estimation. The use cases in iRel40 were equally distributed among these objectives, each of them containing four use cases. Clearly identifying the objective of our desired AI/ML system will, in effect, determine a general form our AI/ML framework will have.

Fault Detection

Fault detection is a process to detect whether a device or a process is faulty in some way, that is, whether its state is abnormal, anomalous, or in some way different compared to normal operation. For example, finding out if a device meets the minimum standards of quality for it to be sold falls under this objective. Fault detection can also include the task of pinpointing the type of fault and its location in case the fault was recognized by the model. In the scope of fault detection, we require data, such as sensor measurements, that include enough information for determining health of the system. We should have many data instances of the system under different circumstances, and the primary assumption is that these data instances can be clustered based on some criteria (that AI/ML finds) into disjoint classes. In its basic form we would have two classes, one for normal operation and one for faulty one. However, you might be able to separate the data instances into even more classes which should indicate different kinds of state the system is in.

Remaining Useful Life Estimation

Remaining useful life estimation is a process of predicting the length of time a machine is likely to operate before it requires repair or replacement. For example, estimating the number of cycles a device will endure before some type of failure occurs falls under this objective. In this case, we require data for the system in various stages of its life. AI/ML algorithm would then be used to find patterns in the data that are characteristic for different stages of operation (and types of degradation that may occur) and connect it to the chronological order that these stages follow each other. From this knowledge, AI/ML system can deduce an estimation for the operation time, number of cycles before a failure occurs. This AI/ML pipeline is usually designed so that a data-driven model gets some (sensor) data as input and returns a single number, an estimation of "time" before a defect. The training data for the model are prepared so that different past measurements of a system with an observed defect are paired with the "time" before that defect occurred. In this way, a data-driven model is trained on such pairs of data instances and fitted to generate remaining useful life estimation so that its output on a past measurement is as close as possible to the observed "time" before failure.

Estimation of Physical State

Estimation of physical state is the process of approximating some properties of a device or a process that is usually difficult to measure in a real-world situation. In the scope of iRel40 this objective is closely connected to the other two objectives, since the physical state being estimated is in all cases used to predict faults or degradation. For example, using expert knowledge, we know that when a device reaches a given physical state, which is estimated, it has failed or will likely fail in the near future.

However, the physical state we are interested in can be very hard to measure and might require special equipment. To circumvent these problems we can use AI/ML to estimate this physical state using other, easily available measurements that should have some correlations with the actually desired physical state. Methodologically speaking, this type is very similar to remaining useful life estimation, since we predict some values that are known beforehand. However, the data acquisition procedure is quite different. For this type, we do not actually require measurements for the system for all stages of its life until failure. We can work with data that only include some stages of the system life and we generalize the learned behavior to later stages of life using expert knowledge. This methodology is convenient if the system has a very long lifetime and gathering data for all stages would be too expensive. However, the drawback is that our understanding of the failure process might not be comprehensive enough to generalize the behavior of the system using measurements only from one part of its lifetime. In iRel40 we found that in most cases using this type of pipeline results in a good approximation of the physical state, even outside of the regime the system was measured, and by proxy failure prediction.

3.2 Data and Machine Learning Aspects

While the problem specification defines a general shape for an AI/ML framework, it is the properties of the available data that affect how the AI/ML pipeline will be structured and which data-driven models can be used inside it.

3.2.1 Model Output Data Characteristics

The output of the AI/ML model, i.e., the information that it returns, can be known or unknown during training of the data-driven model. In both cases, AI/ML methods can be applied; however, the methodology is quite different. Suppose that we are developing an AI/ML framework that should return probability that a failure has occurred, then we can train the model on historical data in which failures are recorded or not. In ML jargon we say that the targets (output of the ML model) are either annotated or non-annotated. This refers to the process of annotation that is a part of data preparation. For example, in manufacturing, we usually have a testing phase in which a product is checked for defects. In this stage, some measurements are performed and an expert decides whether to discard the product based on the measurement readings. In some cases, the expert would annotate that the product for which the measurements were made is faulty or not and possibly what type of error was observed. In other cases, for whatever reason, such annotations are not made, which results in a data set that includes measurements of different products but does not include information about faults that were observed during testing. The reasons for not possessing annotations are diverse and include non-ideal data acquisition procedures (e.g., due to cost limitations), targets can not be measured directly (e.g., due to high noise levels), the volume of meaningful events is negligible (e.g., because larger sample sizes are unfeasible) and so on.

From the perspective of ML methodology we distinguish two classes of techniques that correspond to settings with annotated targets, called supervised learning, and non-annotated targets, called unsupervised learning. Supervised learning is a slightly more mature field where the task is to fit the model so that it returns the correct known output when given a known input. This usually means that we define a model with tunable parameters along with a loss function that tells us how close the output of the model is to the desired target. In the training phase, we then find values for the tunable parameters using an optimization algorithm so that loss function is minimal when our model is applied to our historical data set of desired input, output pairs. In unsupervised learning we do not have this luxury, and training of the data-driven model needs to be performed by other means. The easiest approach is to define some type of similarity measure for our data instances and use this measure to cluster the data and find out how more typical data instances look like and how less typical ones (the outliers) differ from them. This allows us to detect anomalies without actually knowing which exact instances are faulty. However, for this method to work, we require a similarity measure that is meaningful for the problem at hand. In recent years, we have seen great progress in methods to learn such similarity measures from the data itself. This approach is connected to Dimensionality reduction, Manifold learning and Representation learning. In all cases, the model is trained to find a smaller representation for the original data that still retains all the information contained in it. This allows to find new data representations on which it is very easy to define a meaningful similarity measure.

3.2.2 Model Input Data Characteristics

The properties of the data on which the model bases its prediction, or in other words, the properties of the data that are used as an input to the AI model, are another important factor that determines the structure of the AI/ML framework. There are many important characteristics that data sets possess that affect the choice of AI methods. However, here we restrict ourselves to three broad types that affect the methods most profoundly. These are Tabular Data, Time Series and Image Data. Respectively, they can be topologically thought of as zero-, one-, and two-dimensional data sets. Before defining these categories, it is important to make a distinction between a data set and a data instance. A data set is usually composed of many data instances. An atomic entity that an AI model processes is a data instance. When the model makes a prediction, it processes a single data instance. However, when the model is being trained, it requires many such instances, a data set, in order to learn to act intelligently/rationally. In tabular data, a data instance is composed of a set of features that possesses no natural ordering, i.e., one can permute the features without making the data set less meaningful. A simple example of a tabular data set is a table of measurements performed under different conditions that are not easily related among themselves. This is the most general form for a data set. In time-series data, a data instance is an ordered set of vectors. This ordering induces a topology on a data instance that is meaningful and relevant for model training. A simple example of a time series data instance is a table of consecutive measurements of some quantities. The distinguishing property here is continuity, that a measurement in a given time is more naturally related to those performed in close time proximity than to those that are more distant in time. For image data, a data instance is a table of vectors where ordering is meaningful in two directions, both for rows and columns. A simple example of image data instance is a photograph with a clear two-dimensional topology. There RGB values of the neighboring pixels are related among themselves more than those that are farther apart. In other words, continuity is present in two directions. Some more specialized AI models are able to exploit these continuity properties and can be trained with fewer data and can generalize to broader domains more easily.

A data-driven model has to be structured so that it can take the required shape of the data as input. This is a necessity. However, the model can be internally structured in a way that it searches for correlations and patterns on groups of features where they are, in fact, expected to emerge while ignoring the ones that are not expected to be useful. For example, when detecting faults from images, a data-driven model would try to find patterns in pixel values that are correlated with the presence of a fault. A general data-driven model would search for these patterns from a set of all possible pixel groups, even considering values from pixels that are scattered across the image. In most cases, this is not necessary since patterns that we are interested in are localized to a small part of the image, such as a crack in a material (when doing failure detection). For this reason, we use specialized model architectures that focus their pattern searching on feature groups that are topologically close, as are neighboring pixels on an image or neighboring time steps in a time series. Examples of such special architectures from deep learning models include convolutional networks, used for both images and time series, and recurrent neural networks and transformers, used for time series. There is another useful consideration when our input data have special structure and those are symmetries. In many cases, the underlying problem that AI/ML framework is designed for has some symmetries, meaning that if we transform the input data in some special way, it should not change the output of a model. If we return to the crack detection example, it may not matter where on the image the crack is positioned and how it is oriented to be detected by the framework. Mathematically speaking, the model should process data in the same way if we shift the image in space and/or rotate it. Such symmetries can be explicitly encoded in the model structure [4] and by this we restrict pattern searching to patterns that are meaningful for our problem.

3.2.3 Machine Learning Approaches

The bottom part of Figure 1 lists the AI / ML models that were used in each group of use cases and how these methodologies are grouped into broad ML approaches. This is a different view from the top-hierarchical part that specifies the problem. We see that for reliability the most useful broad classes of approaches in machine learning are: Classification, Regression, and Anomaly Detection. They encapsulate use cases into areas that align well with the problem specification hierarchy. Regression is an approach in which a machine learning model is trained to return numerical value(s), for example, to predict the time until critical failure. Classification is an approach where a machine learning model is trained to predict a discrete object known as a class which belongs to a set of predetermined classes, for example, predicting mode of failure. The distinguishing property of classification is that the model return values do not possess any natural ordering such as is the case with regression. Anomaly Detection (also referred to as outlier or novelty detection) is an approach where the machine learning model is trained to recognize whether the input data significantly deviates from the majority of the data and do not conform to a well-defined notion of normal behavior. These three approaches to ML are a subset of what we usually refer to as types of ML, as depicted in Figure 2.

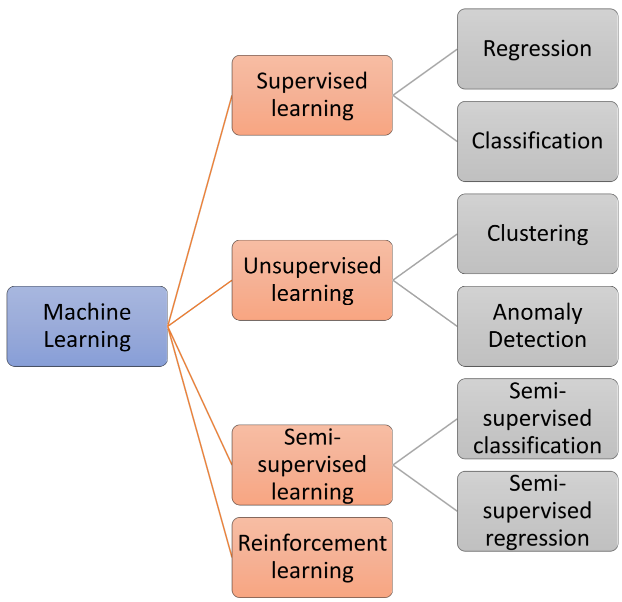

Fig. 2 Types of broad ML approaches known from the literature. For reliability problems the most pronounced are Regression, Classification and Anomaly detection. Extracted from [15].

In the middle part of the Figure 1, we can observe the types of machine learning model that were used for a given leaf node in the categorization hierarchy. However, there is no definite rule that would indicate which class of models is most suitable for a problem with the given characteristics. This is expected, as it is known from the current AI literature that the selection of the most suitable model is highly dependent on the data properties that are outside the scope of the simple categorization that we present [8]. Finding a rule that would tell which model is the most appropriate for the given data set is ongoing work [5] and depends on the data distribution (skewness, correlations, asymptotic behavior of outliers, topology of the support manifold). Nevertheless, we can recognize that artificial neural networks and decision trees are the most commonly used models for reliability problems. A good guideline for applying ML in general is the decision workflow depicted in Figure 3. This flowchart is quite rudimentary and considers only the most basic considerations about the properties of the problem and data, however, it is a good starting tool to choose the appropriate methodologies and models. It is important to note that during the framework development, the choice of architecture should change based on an iterative re-test of the framework.

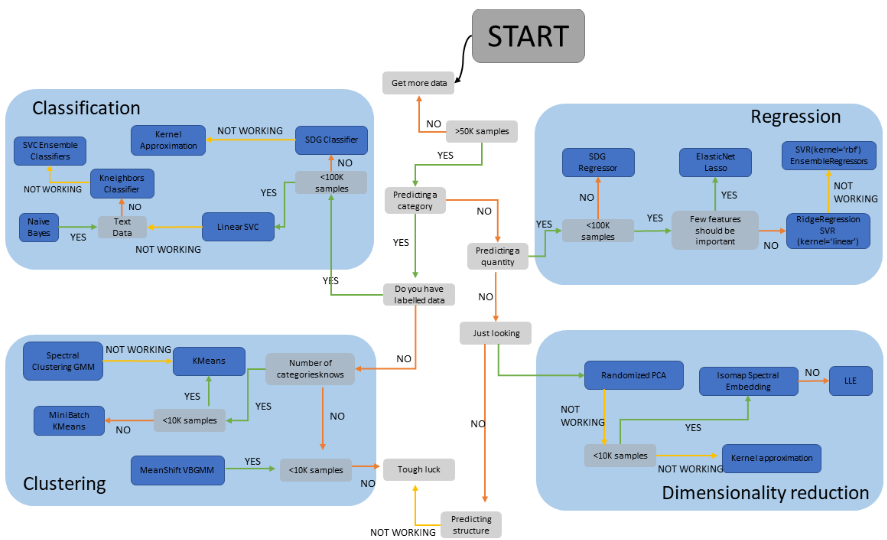

Fig. 3 Approximate guideline for choosing an ML methodology and a specific data-driven model for a given task based on the properties of the problem and the properties of the data. Extracted from [11].

We can observe that some classes of ML methodologies depicted in Figure 2 were not used during the iRel40 project. Clustering is one example of the type that was not used as a base method, however, it was used as a data preprocessing or postprocessing step. Clustering helps us to better understand the structure of our data set, and from its results, there are no direct means to estimate traits connected to reliability. Another important category is semi-supervised learning, which is a methodology for working with data sets that are partially annotated (some instances in the data set are annotated, while usually many are non-annotated). At first glance, there is no reason this methodology would not apply to reliability; however, in iRel40 the data sets used for training the data-driven models each came from a unique source, which means data instances were either all annotated or none of them were. In reliability, semi-supervised learning would be useful in cases where data acquisition is done by at least two different sources, with one having the means to annotate targets while the other whiteout those means. Reinforcement learning, on the other hand, is a more niche methodology that can work in the absence of data but requires an interactive environment for the model to learn on. Instead of looking for patterns in the data, model training is performed using on-line experience with that environment. An example of problems that require this methodology is robotics where it is impossible to directly define what is the desired behavior of the model, but there is a way to assess whether the model is doing well so far or not (such as measuring the time a robot is able to stand before falling).

iRel40 project was very successful in having very diverse use cases related to reliability in terms of the type of problem and data characteristics. This means that the AI techniques used to address them were also distinct and pose unique challenges for each group of use cases. For instance, even though we had two use cases of fault detection, both involving image data, they were fundamentally different due to the fact that they belong to two different classes of approaches in machine learning, one to classification and the other to anomaly detection. While they share similarities in methodology such as the use of data augmentation and convolutional operations that are specific to image data, the architecture of the AI models and algorithms to train them are distinct. Additionally, the obstacles encountered are different, with class imbalance being a major issue for classification, while in anomaly detection the most important difficulties were training the model to learn representations that scale well. Therefore, even though convolutional neural networks are the base model for both cases, the main effort of the use case lies in making these AI models perform useful tasks in their specific situation using techniques that are tailored to the problem and data characteristics.

4 Discussion

The categorization of frameworks presented here is, of course, not comprehensive. It can be challenging to assign a given use case to a single category. This is because real-world problems are often heterogeneous, involve constraints, and involve multiple modalities. If all of these modalities were included in the categorization, it would be too complex and less useful to the reader. This is also true for AI methodology, where methods are interconnected and hard to separate without compromising the readability and understandability of the categorization. Here are some examples of real use cases that demonstrate these facts.

In some cases, it is difficult to clearly define the characteristics of the input data because transformations can be applied to the data set that converts it from one category to another. Employing such transformations can be beneficial for modeling, as it can put the data in a form that is more natural for the given model to train on. For example, in one use case, they showed the benefits of transforming vibrational input data that are originally of a data series type to both a tabular and an image data type. The transformation from time series to image data type was performed using a shorttime Fourier transform, which is a lossless transformation and expands the topology of the data instance from only time to both time and frequency domains. It was shown that the reliability-relevant patterns can be more easily learned by the AI model when the data instances are in such two-dimensional (image) form. Their transformation to tabular data type (by calculating several time-series statistical features) is not lossless; however, the chosen feature set was found to be comprehensive enough for the task at hand. In fact, the approach of transforming time series data to tabular one was found to be the most accurate fault detection system out of the considered ones. So, we see that, at least from the modeling perspective, it is difficult to clearly pinpoint the input data type, since it can be transformed into even all three types considered in our categorization interchangeably. Due to this fact, the data types considered in Figure 1 are derived from the properties of the raw data available without considering the data processing transformations used by the partners.

It is also not uncommon that the data set includes both annotated and non-annotated targets at the same time. Annotating targets in real-world data sets is usually an expensive and also error-prone task. The resulting data set can, therefore, include data instances where targets are annotated and instances where they are not. To fully benefit from a data set of such a mixed type, semi-supervised learning can be employed. This is an approach to deploy AI models in such a way that they perform both supervised learning (classification or regression) and anomaly detection at the same time. In this case, both the annotated and non-annotated parts of the data set are used for training, which usually results in superior data mining and pattern recognition compared to training on each part separately. In addition to better generalization of the AI models trained in that way, we also gain the ability to detect important events that were not considered when data set annotation was performed, but are nevertheless relevant for reliability. For example, in one use case, a similar methodology was deployed to better pinpoint the type of defect in failure detection. There are many modes of failures of which we have limited knowledge, and understanding them is crucial for improving the production process to result in as low number of faults as possible. Their AI model is able not only to classify types of defects that are known to exist, but it can also detect new classes of defects alongside their subclasses. In Figure 1 we do not consider mixed cases and state whether the majority of targets in the data set are annotated or not.

It is important to note that the fact that a given use case belongs to more than one category can be beneficial. Such mixed properties allow the use of several (opposing) methodologies at the same time and possibly bring together their advantages. However, such endeavors are labor intensive because for several techniques to work in cohesion on a real-world problem requires custom model development along with all the peripheral techniques associated with it and usually involves a lot of trial and error with no guarantee. Fortunately, this project culminated several successful examples of such AI algorithm customization tailored for a very specific problem and data. The taxonomy in Figure 1 cannot possibly cover all the techniques used in iRel40, although they are more important than a simple choice of the AI model, which we cover in Figure 1. The categorization presented in this chapter also provides a clearer view of the diversity of use cases in this project and in what way they diverge among themselves when considering them as problems that are being solved by machine learning.

Acknowledgements The authors acknowledge the financial support of the Slovenian Research Agency (research core funding No. P2-0098 and young researcher grants). The work is also part of a project that has received funding from the ECSEL Joint Undertaking under grant agreement No 876659 (iRel40).

References

1. Azeem, M., Haleem, A., Javaid, M.: Symbiotic relationship between machine learning and industry 4.0: A review. Journal of Industrial Integration and Management **7**(03), 401–433 (2022)
2. Belkacem, L., Simeu-Abazi, Z., Dhouibi, H., Gascard, E., Messaoud, H.: Diagnostic and prognostic of hybrid dynamic systems: Modeling and rul evaluation for two maintenance policies. Reliability Engineering & System Safety **164**, 98–109 (2017)
3. Cao, D., Hu, W., Zhao, J., Zhang, G., Zhang, B., Liu, Z., Chen, Z., Blaabjerg, F.: Reinforcement learning and its applications in modern power and energy systems: A review. Journal of modern power systems and clean energy **8**(6), 1029–1042 (2020)
4. Finzi, M., Stanton, S., Izmailov, P., Wilson, A.G.: Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In: International Conference on Machine Learning. pp. 3165–3176. PMLR (2020)
5. He, X., Zhao, K., Chu, X.: Automl: A survey of the state-of-the-art. Knowledge-Based Systems **212**, 106622 (2021)
6. Jardine, A.K., Lin, D., Banjevic, D.: A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mechanical systems and signal processing **20**(7), 1483–1510 (2006)
7. Kan, M.S., Tan, A.C., Mathew, J.: A review on prognostic techniques for non-stationary and non-linear rotating systems. Mechanical Systems and Signal Processing **62**, 1–20 (2015)
8. Kerschke, P., Hoos, H.H., Neumann, F., Trautmann, H.: Automated algorithm selection: Survey and perspectives. Evolutionary computation **27**(1), 3–45 (2019)
9. Mahadevkar, S.V., Khemani, B., Patil, S., Kotecha, K., Vora, D.R., Abraham, A., Gabralla, L.A.: A review on machine learning styles in computer vision—techniques and future directions. IEEE Access **10**, 107293–107329 (2022)
10. Purwins, H., Li, B., Virtanen, T., Schlüter, J., Chang, S.Y., Sainath, T.: Deep learning for audio signal processing. IEEE Journal of Selected Topics in Signal Processing **13**(2), 206–219 (2019)
11. Samson, M.: Mineral Resource Estimates with Machine Learning and Geostatistics. Master's thesis, University of Alberta (2020). https://doi.org/10.13140/RG.2.2.24972.05764
12. Singh, V., Chen, S.S., Singhania, M., Nanavati, B., Gupta, A., et al.: How are reinforcement learning and deep learning algorithms used for big data based decision making in financial industries–a review and research agenda. International Journal of Information Management Data Insights **2**(2), 100094 (2022)
13. Soleimani, M., Campean, F., Neagu, D.: Diagnostics and prognostics for complex systems: A review of methods and challenges. Quality and Reliability Engineering International **37**(8), 3746–3778 (2021)
14. Thakur, R.S., Chatterjee, S., Yadav, R.N., Gupta, L.: Image de-noising with machine learning: A review. IEEE Access **9**, 93338–93363 (2021)
15. Xu, Z., Saleh, J.H.: Machine learning for reliability engineering and safety applications: Review of current status and future opportunities. Reliability Engineering & System Safety **211**, 107530 (2021). https://doi.org/https://doi.org/10.1016/j.ress.2021.107530, https://www.sciencedirect.com/science/article/pii/S0951832021000892