SYSTEM RELIABILITY ENGINEERING IN THE AGE OF INDUSTRY 4.0: CHALLENGES AND INNOVATIONS

ANTOINE TORDEUX*, TIM M. JULITZ, ISABELLE MÜLLER, ZIKAI ZHANG, JANNIS PIETRUSCHKA, NICOLA FRICKE, NADINE SCHLÜTER, STEFAN BRACKE, AND MANUEL LÖWER

UNIVERSITY OF WUPPERTAL, SCHOOL OF MECHANICAL ENGINEERING AND SAFETY ENGINEERING
GAUSSSTRASSE 19, 42119, WUPPERTAL, GERMANY

ABSTRACT. In the era of Industry 4.0, system reliability engineering faces both challenges and opportunities. On the one hand, the complexity of cyber-physical systems, the integration of novel numerical technologies, and the handling of large amounts of data pose new difficulties for ensuring system reliability. On the other hand, innovations such as AI-driven prognostics, digital twins, and IoT-enabled systems enable the implementation of new methodologies that are transforming reliability engineering. Condition-based monitoring and predictive maintenance are examples of key advancements, leveraging real-time sensor data collection and AI to predict and prevent equipment failures. These approaches reduce failures and downtime, lower costs, and extend equipment lifespan and sustainability. However, it also brings challenges such as data management, integrating complexity, and the need for fast and accurate models and algorithms. Overall, the convergence of advanced technologies in Industry 4.0 requires a rethinking of reliability tasks, emphasising adaptability and real-time data processing. In this chapter, we propose to review recent innovations in the field, related methods and applications, as well as challenges and barriers that remain to be explored. In the red lane, we focus on smart manufacturing and automotive engineering applications with sensor-based monitoring and driver assistance systems.

1. Introduction

The recent advent of computational learning algorithms has revolutionised many disciplines [76], whether in engineering [72, 149, 3], the natural sciences [23, 108, 58, 129] or the social sciences [150, 40, 160]. This revolution, which began during the 1990s, has also profoundly impacted reliability engineering [39, 161, 163, 2]. Rather than the emergence of new methods, the rise of Internet-of-Things (IoT), cyber-physical systems and Industry 4.0, along with the growing use of (intelligent) sensors, connected and autonomous systems, data storage, and advanced high-performance computational prognostics, are the key drivers of this change [137, 31, 75, 47]. In fact, learning approaches have reached maturity and meet the need to address the reliability of increasingly complex systems.

Condition monitoring, predictive maintenance, and remote support stand out as remarkable advancements, using real-time data and AI to predict and prevent equipment failures [119, 132, 74]. These approaches reduce downtime, lower costs, and extend equipment lifespan, thereby improving system sustainability. Other recent progresses concern system design, e.g. fault-tolerant, fail-safe, fail-operational system structures and dynamic reconfiguration, especially for safety-critical systems [114, 24]. In addition, computer-aided reliability analysis based on simulation, digital twins and real-time data assimilation allows, for example, the effects of degradation and other dynamic factors in mechanical and cyber-physical systems to be taken into account in real time [100, 145, 146]. Applications cover various fields, including (smart) manufacturing and industrial production [31, 4, 52], energy [89, 39], robotics, transport and logistics [105], (smart) cities and healthcare [99]. The increasing complexity of systems and algorithms, the integration of new digital technologies and the management of large amounts of data generated in real time remains major challenges. Overall, the convergence of advanced technologies in Industry 4.0 requires rethinking reliability tasks with a focus on adaptability and real-time data processing.

The aim of this chapter is to review the latest innovations, applications and relevant methodologies in the field, as well as the challenges and barriers that remain to be addressed. A red lane in the chapter is dedicated to smart manufacturing, automotive and traffic engineering applications, in particular sensorbased monitoring and driver assistance systems. The chapter is structured as follows. We identify and

^{*} Corresponding author – E-mail address: tordeux@uni-wuppertal.de

review in section 2 three recent innovation domains in advanced system reliability engineering, namely real-time IoT-enabled sensor-based reliability analysis in section 2.1, improved reliability through system design in section 2.2, and simulation-aided reliability analysis in section 2.3. Then we present some challenges and discuss barriers that remain to be explored in section 3, covering difficulties related to safeguarding AI and real-time data processing in section 3.1; system complexity and component interdependence in section 3.2; and human-machine collaboration and sociotechnical systems in section 3.3. Finally, in section 4, we review related statistical methods and machine learning techniques for multivariate datasets. The methods include unsupervised learning approaches for data visualisation, dimension reduction and clustering in section 4.1, and supervised learning methods for advanced predictive analytics, e.g., multivariate regression, classification, and machine learning techniques in section 4.2.

2. Innovations

In this section, we review the most important recent advances in the three innovation areas of prognostics, system design and simulation-based reliability analysis (see Figs. 1 and 2). These research areas of reliability engineering have benefited from important innovations in recent decades. First, sensor-based monitoring based on machine learning techniques, supported by the Internet of Things and Industry 4.0,

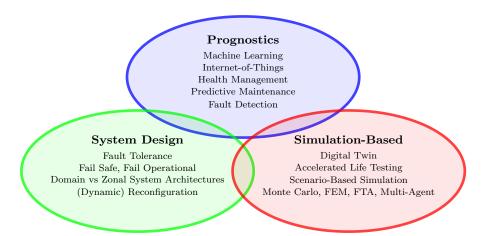


FIGURE 1. Main sectors that contribute to advanced reliability engineering techniques during the last decades.

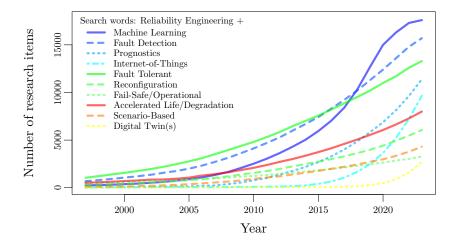


FIGURE 2. Cumulative number of research articles from 1997 to 2023 with the tags 'Reliability Engineering' and related keywords in Google Scholar (query made on 01.10.2024).

has shown significant progress in prognostics and health management, real-time condition monitoring, fault detection, self diagnosis and predictive maintenance. Then system design, which showed how advanced architectures can improve the system reliability through redundant configurations, fault-tolerance, fail-safe, fail-operational architectures, dynamic reconfiguration and automation. Finally important innovations have also been provided by simulation-based reliability analysis, e.g. using digital twins, accelerated life testing, or scenario-based simulation. Figure 2 shows the number of research articles over the last twenty years with related keywords in the Google Scholar engine. The trends are clearly more than linear for all the selected keywords. We can observe that the keywords Digital Twin(s), Internet-of-Things and, to a lesser extent, Prognostics and Fault Detection are very topical. The keyword Machine Learning peaked about five years ago, although it is still the most cited keyword. Fault Tolerant, Accelerated Life/Degradation, Reconfiguration, Fail-Safe/Operational, and Scenario-Based are keywords that were established earlier in the 2000s.

2.1. **Prognostics.** Sensor-based monitoring has emerged as an innovation in reliability engineering, driven by advances in data acquisition, real-time analysis, and integration with machine learning (ML) and AI technologies. IoT integration in the context of Industry 4.0 enables continuous or periodic monitoring of operating conditions in systems by using for example **Wireless Sensor Networks** (WSN). WSN collect vast amounts of data (i.e. vibration, temperature, pressure, etc.) remotely and decentralised. For (real-time) analysis purposes, IoT platforms provide access to this transmitted data, where it is processed to assess system health. The data acquisition is a crucial step in the process. Indeed, the labelled data enables machine learning algorithms to be trained and validated. Numerous open-access online data repositories exist for this purpose, such as Kaggle with specif keywords, NASA Prognostics Data Repository, UC Irvine Machine Learning Repository, etc.

A comprehensive framework that incorporates these innovations is **Prognostics and Health Management** (PHM). PHM consists of four main components:

- (1) Data Acquisition and Condition Monitoring,
- (2) Diagnostics,
- (3) Prognostics.
- (4) Health Management.

Within the PHM framework, data acquisition and condition monitoring integrates sensor-based monitoring with data analytics and machine learning techniques to monitor system health. Such a area relies on advanced sensor technology (e.g., WSN). Diagnostics aims to detect abnormal system behaviour, identify and isolate faults, and analyse the root causes of system failures, while Prognostics goes a step further by estimate the remaining useful life (RUL) and failure prediction based on current operational conditions. System maintenance strategy (i.e. predictive maintenance) and performance optimisation are enabled through the use of diagnostics and prognostics. Optimising maintenance schedules by using diagnostics and prognostics minimises unplanned downtime and maximises the operational life of (industrial) systems and performance optimisation, which is referred to Health Management. Many reviews exist on the topics, see, e.g., [137, 31, 75, 47, 119, 132, 74] and references therein.

2.1.1. Fault detection and diagnosis. In modern industrial applications where sensor networks are implemented, a large amount of process data is available. They contain important information about process dynamics and system health. In general, the diagnostics process integral to the Fault Detection and Diagnosis (FDD) framework is a series of transformations on system measurements [153], shown in figure 3. The FDD system uses these measurements from the process as an input. Then, in the next step features are derived from measurements through feature engineering or residual generation. Features act as health indicators of faulty (abnormal) process behaviour and provide a basis for detecting changes, which refer to fault symptoms. For example [170] provides an overview of the health indicator (HI) construction for rotating machinery. In the decision space, changes in the operating conditions are detected with threshold or discriminant functions. Subsequently, the fault diagnosis constitutes a mapping between the set of fault symptoms and the set of fault classes. In order to ascertain the fault classes, including the normal operating condition, a variety of methods may be employed, such as pattern recognition, classifiers, inference methods

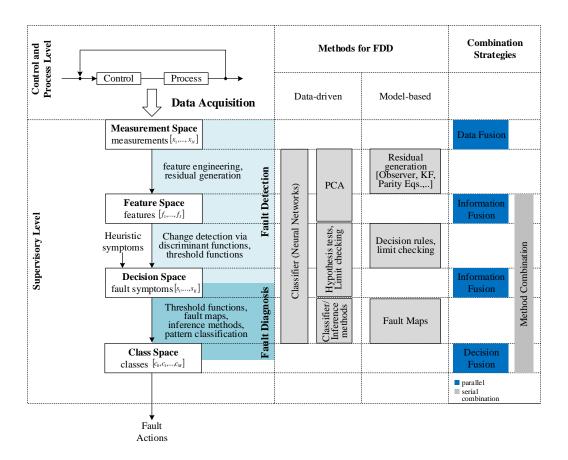


FIGURE 3. Transformations within the FDD system, used data-driven and model-based methods and combination strategies for hybrid approaches [158]

or threshold functions. Afterwards the output of the diagnostics process, i.e. features and fault classes, may be used as an input for prognostics and health management [172, 144].

The methods employed in diagnostics can be broadly classified into two categories: data-driven and model-based, shown in figure 3. In addition, the work of [158] considered knowledge-based approaches. Data-driven methods extract information from historical process data and are generally divided into supervised (i.e., classification and regression techniques) and unsupervised methods (i.e., statistical models and clustering techniques). Supervised methods train algorithms to historical data with the objective of learning and distinguishing patterns within the high-dimensional space of process measurements or features. The patterns may then be used to perform fault detection and fault diagnosis, specifically to predict the fault symptoms or classes for new data. However, such methods require labelled training data, whereby for each input feature vector, the corresponding class label must be known [158].

Unsupervised methods do not require class labels and are therefore suitable for a wider range of applications where these are not available. Most unsupervised methods focus only on the first or second transformation step to extract features or fault symptoms [158]. Principle component analysis (PCA) is the most common approach to dimensionality reduction and feature extraction. Typical data-driven methods especially used for classification purposes are artificial neural networks (ANN), support vector machines (SVM), k-nearest-neighbour (kNN), decision trees, and Random Forest, see e.g. [123, 69, 5].

Model-based approaches rely on numerical reproduction in the form of mathematical equations of the real physical process. The mathematical equations are based on the knowledge of the input-output relationships of the process [67]. Due to these known input-output relationships, the model requires less process

data compared to data-driven approaches. The operating state of the process is determined by comparing measurements of the real process state with the state predicted by the physical model. An inconsistent comparison indicates a faulty condition. Specific inconsistencies are referred to as residuals. The residuals correspond to the characteristics shown in the workflow fig.3. The subsequent fault diagnosis can be carried out using threshold logic functions, for example. Typical methods for FDD based on physical models are bond graphs, parity equations, state estimation (e.g., using Kalman filter), state observers or simple limit and trend checking, see e.g. [127, 66, 33, 50, 67].

Hybrid FDD approaches combine two or more established methodologies. Additionally, some approaches consider direct integration of a priori knowledge. Hybrid FDD methods may be combined parallel or serial through a variety of algorithms, see [158]. In the case of a data set comprising a significant amount of unlabelled data, the following hybrid methods may be employed to automatically expand the data set, for instance:

- Semi-supervised: A model is trained using a small, manually labelled data set, which is then applied to unseen, unlabelled data. Subsequently, the previously unlabelled data is incorporated into the training data set, with a label selected by the model itself. It is crucial that the base data set with the labelled data is selected to be representative of the use case, in order to facilitate the optimal training of the model. In an iterative process, the data labelled by the model is then added to the training data set, with the objective of optimising the model [21, 92].
- Activ Learning: The model is trained in an iterative manner, whereby the most "uncertain" or informative data points are selected for human labelling. This approach entails that only the most necessary data is labelled manually, while the model assumes responsibility for the remaining labels automatically. This reduces the labelling effort, but with the potential for incorrect labelling of data points in which the model is certain [26, 30].

2.1.2. Remaining Useful Life Prediction and Sampling Methods. Predictive maintenance strategies are generally based on real-time prediction of the **Remaining Useful Life** (RUL) of a system. These strategies allow for proactive maintenance planning, which helps avoid unexpected downtime and optimizes operational efficiency. As RUL predictions reach critical thresholds, maintenance interventions are initiated, enabling actions to be taken before failures occur. RUL prediction methods are widely applicable, particularly in industries that rely on battery and bearing maintenance, as reviewed in [65, 55, 60, 74].

Common predictive maintenance algorithms can be categorised into *shallow* and *deep learning* methods, each providing unique advantages. In shallow learning, methods like *Logistic Regression* are computationally efficient and simple to implement, though they assume linear relationships, which limits their suitability in complex environments [65, 55]. *Artificial Neural Networks* (ANNs) are also common in shallow learning; they store complex patterns and are robust to noise but may yield unpredictable results under certain conditions [90].

Other popular shallow learning techniques include Support Vector Machines (SVM), which perform well in high-dimensional data spaces, though they are less effective with large datasets. Deep Belief Networks (DBNs) require minimal labeled data and have quick training times but pose challenges in determining optimal network structures [60]. Deep learning approaches, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), are highly effective for processing high-dimensional or sequential data, albeit with substantial computational requirements [74].

Convolutional Neural Networks excel in tasks like image recognition due to their automatic feature detection capabilities, although they require considerable computational resources [123]. Recurrent Neural Networks are particularly well-suited for time-series data because of their capacity to store sequential information; however, they are prone to the vanishing gradient problem, which affects prediction accuracy over long sequences [67]. *Hidden Markov Models* (HMMs) are also effective for capturing sequential dependencies, although they are unsupervised and necessitate large datasets for training [66, 50].

Signal sampling and acquisition are foundational in digital technology, with applications from control systems to digital communication. Traditional sampling employs a uniform time-based cycle, known as *Riemann Sampling* (RS), which integrates along the time axis. This method is widely used for its simplicity in analysis and implementation [164, 122]. In contrast, *Lebesgue Sampling* (LS) is an event-driven approach where data points are recorded only when a measurement transitions between states. This approach, which

integrates along the y-axis (i.e., measurement value), reduces the number of data points and is ideal for systems with limited computational resources [122].

In RUL prediction, physical quantities indicating equipment health gradually change over time due to wear or aging. In early operational phases, signals typically remain stable with minimal variation, making continuous monitoring through RS both redundant and resource-intensive. In such cases, Lebesgue-based sampling techniques offer a novel solution, especially when RUL prediction is constrained by computational limits [122]. By focusing on state changes rather than continuous time-based measurements, LS-based sampling in hybrid models can reduce data volume while maintaining predictive accuracy, making it well-suited for edge devices with limited processing capacity.

Hybrid models combining Lebesgue Sampling with traditional data-driven or model-based methods benefit from event-driven data acquisition, allowing for efficient resource allocation without compromising RUL prediction quality. This sampling adaptation enhances the hybrid model's ability to manage large datasets more efficiently, particularly under constrained computational conditions, contributing significantly to predictive maintenance solutions.

Algorithm	Advantages	Limitations
Logistic Regression	Simple, efficient	Assumes linearity
Artificial Neural Network (ANN)	Robust to noise, stores patterns	Unpredictable behavior
Support Vector Machine (SVM)	Effective in high dimensions	Not suitable for large datasets
Deep Belief Network (DBN)	Quick training on small datasets	Network structure complexity
Convolutional Neural Network (CNN)	Excels in feature detection	Computationally demanding
Recurrent Neural Network (RNN)	Effective for time-series data	Suffers from gradient vanishing
Hidden Markov Model (HMM)	Captures sequential dependencies	Requires large datasets
Cluster Visualization	No need for labeled data	Lacks standardised parameter tuning

Table 1. Comparison of Predictive Maintenance Algorithms based on shallow and deep learning approaches [65, 55, 60, 74].

As shown in Table 1, each algorithm presents distinct strengths and limitations, and their selection depends on the system's specific data and operational requirements. These predictive maintenance approaches, particularly with the integration of Lebesgue Sampling in hybrid models, contribute significantly to enabling real-time, data-driven maintenance solutions across various industrial applications.

- 2.2. **System design.** The chapter discusses fault-tolerant system design in safety-critical and highly automated environments. It provides an overview of key strategies and architectural principles needed to ensure system reliability, availability and safety under failure conditions. Section 2.2.1 introduces core fault tolerance concepts, including *fail-safe*, *fail-degradation*, and *fail-operational strategies*. Section 2.2.2 explores redundancy techniques and their impact on system reliability and availability. Section 2.2.3 outlines different system architectures, highlighting *domain* and *zonal approaches*. Finally, Section 2.2.4 focuses on *dynamic reconfiguration*, emphasizing real-time adaptation to maintain system functionality.
- 2.2.1. Fail-safe, fail-degradation and fail-operational concepts. Fault tolerance is a critical aspect of modern system design, especially in safety-critical and highly automated environments [84, 68, 131, 130]. As systems become more complex, integrating numerous electronic components and software subsystems, the

likelihood of failures increases. Faults in one component can cascade through interconnected systems, leading to catastrophic consequences if not properly managed. For example, in autonomous vehicles, the failure of a single sensor or control module could potentially lead to dangerous driving conditions if not mitigated. Therefore, fault tolerance ensures that the system can continue to operate safely and effectively, even when individual components fail. To achieve fault tolerance, three key concepts: Fail-safe, fail-operational, and fail-degradation provide distinct approaches to managing failures and ensuring system reliability and availability [140]. To further clarify these strategies, Fig. 4 is illustrating how fail-safe, fail-operational, and fail-degradation approaches differ from each other and how they work together to create a fault-tolerant system.

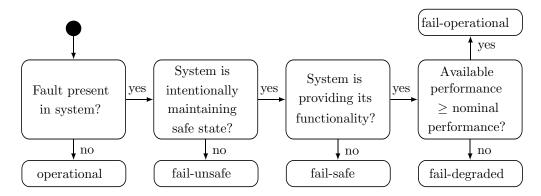


FIGURE 4. Comparison of Fail-Safe, Fail-Operational, and Fail-Degradation strategies [140].

Fail-Safe systems transition into a predefined safe state when a failure occurs [154]. This safe state is designed to minimize risks and prevent any further harm. In automotive systems, this often means ceasing functionality to avoid dangerous situations, such as in braking systems where mechanical fallback mechanisms may be activated upon failure of electronic components. Although the system stops functioning, fail-safe strategies ensure that safety is prioritised over continued operation. The focus is on bringing the system to a state that poses no unreasonable risk, even if this means halting operations entirely.

Fail-Operational systems maintain their full functionality even after a fault has occurred [157]. This is crucial in applications where uninterrupted operation is essential, such as in autonomous driving or aerospace systems. Fail-operational systems are designed to continue providing their specified functionality without performance degradation. For instance, in autonomous vehicles, even if one sensor fails, the vehicle must continue to operate using redundant sensors to complete its mission or reach a safe state. Fail-operational strategies ensure that critical systems can continue functioning without a loss in performance, enhancing both system availability and safety.

Fail-Degradation provides a middle ground, allowing systems to continue operating with reduced performance in the event of a failure [166]. Unlike fail-operational systems, which maintain full functionality, fail-degradation allows for impaired functionality while still ensuring safety. In the automotive context, a fail-degradation strategy could involve reducing the speed of an autonomous vehicle if its primary navigation system fails, allowing it to safely continue driving at a reduced capability or move to a safer location. This concept is especially useful when full operational redundancy is too expensive or unnecessary. Fail-degradation increases both reliability and availability by ensuring that the system can still function, albeit at a lower capacity, instead of shutting down entirely. Fail-degradation can also be part of fail operational concepts [157].

2.2.2. Redundancy and its role in fault tolerant systems. Redundancy is fundamental to achieve fault tolerant capabilities in safety-critical systems [71]. By introducing multiple components or subsystems that can take over in case of failure, redundancy ensures that the system remains operational or enters a safe state when faults occur. There are different strategies for implementing redundancy, often through \mathbf{m} -out-of- \mathbf{n} systems, where n components are in operation, and the system continues functioning as long as at least m components are working [70, 152].

In redundant systems, **NooN** represents a *serial configuration*, where all N components must function correctly for the system to operate. This configuration offers high reliability but is vulnerable, as any single

failure causes the entire system to fail. On the other hand, $\mathbf{100N}$ represents a parallel configuration, where the system can function as long as at least one of the N components works.

Reliability focuses on the probability that the system will operate without failure over a specified time. For a redundant system with n identical components, each with reliability $R_i(t)$, the overall system reliability $R_s(t)$ for an m-out-of-n configuration is given by:

(1)
$$R_s(t) = \sum_{k=m}^{n} \binom{n}{k} R_i^k(t) \left(1 - R_i(t)\right)^{n-k}$$

This formula calculates the probability that at least m components are functioning correctly, ensuring system operation. Redundancy in fail-safe systems guarantees that, upon failure of one or more components, the system transitions into a safe state. The goal in such systems is not to continue operation, but rather to prevent unsafe conditions by shutting down or moving to a controlled, safe condition.

Availability, on the other hand, is crucial for *fail-operational* systems, where the system must remain functional even after failures. Availability is a measure of the system's ability to perform its intended function at any given time, factoring in maintenance and repair periods. In *fail-operational* systems, maintaining availability is often more critical than ensuring perfect reliability, as the system needs to continue functioning despite component failures. The *availability* of a redundant system is influenced by both its reliability and its ability to be repaired or reconfigured, and is asymptotically calculated as:

$$A_s = \frac{MTTF}{MTTF + MTTR},$$

where MTTF is the mean time to failure, and MTTR is the mean time to repair. In this context, redundancy allows the system to continue operating while repairs are made, ensuring minimal downtime. The effectiveness of repair or reconfiguration in fault-tolerant systems is directly influenced by the ability to separate or diversify system elements [126, 138].

Separation refers to the physical isolation of redundant subsystems, ensuring that a failure in one part of the system does not affect others. For example, in aircraft systems, placing critical sensors or control units in different parts of the vehicle prevents damage in one area, such as from fire or impact, from compromising the entire system. This isolation allows faulty components to be repaired or replaced while the rest of the system remains operational, thus maintaining availability [126, 138].

Diversification, on the other hand, involves using different technologies, designs, or suppliers for redundant components to reduce the risk of common-cause failures. In a fail-operational system, diversity ensures that if one component fails due to a specific failure mode or environmental condition, another component with a different design or tolerance can continue functioning. For instance, in autonomous vehicles, using a mix of sensors like lidar, radar, and cameras ensures that the system can maintain functionality even if one type of sensor is compromised by conditions like fog or rain [126, 138].

In fail-degradation systems, redundancy enables the system to degrade gracefully in response to failures, continuing to function at a reduced capacity rather than completely ceasing operation [140]. For instance, if several sensors fail in an autonomous vehicle, the remaining sensors may allow the vehicle to continue driving, although with reduced accuracy or slower speeds. This kind of redundancy ensures that the system can maintain essential functions and prevent complete shutdown, balancing reliability and availability.

2.2.3. System architectures. The concepts of redundancy, separation, and diversification are foundational to designing robust electronic control units (ECUs), particularly in safety-critical systems such as automotive and aerospace applications [138]. In practice, ECUs use redundancy strategies to maintain operation in the face of failures. For instance, a 100N configuration may be employed for non-critical systems where basic functionality must continue despite component failures, while a 2002 or 2003 configuration could be used in safety-critical systems that require more stringent fault-tolerance measures [84]. By deploying multiple ECUs in parallel, the system can isolate faults or switch to a backup ECU if necessary [87]. Two different fail-operational ECU architectures are shown in Fig. 11 in Appendix 1.

ECU redundancy schemes can be integrated into broader architectures at the system level, which are commonly organised into either domain architectures or zonal architectures [87, 125]. These system architectures dictate how ECUs are distributed, communicate, and handle redundancy across the entire System. The two approaches are illustrated in Fig. 5.

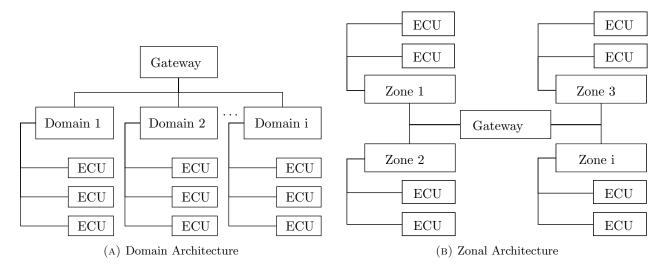


FIGURE 5. Electronic Control Unit Architectures at the System Level

In a domain architecture, ECUs are organised based on their functional roles, with each domain representing a specific set of functions. For example vehicle, the powertrain domain manages functions related to the engine, transmission and drivetrain. The body control domain oversees lighting, doors, windows, and climate control. The infotainment domain handles multimedia, navigation, and user interfaces and the safety domain manages systems such as airbags, ABS (anti-lock braking system), and electronic stability control. In this architecture, each domain operates somewhat independently, with ECUs dedicated to specific tasks within that domain. Redundancy strategies like 2003 TMR or 2002 DFS can be applied within each domain to ensure critical functions like powertrain or safety can continue to operate even in the event of a failure. The advantage of a domain architecture is that it allows for easier management and development, as each domain is functionally isolated, allowing for specialization and optimization of ECUs within that domain. However, a significant drawback is that communication between different domains, such as powertrain and safety, requires more complex wiring and can introduce latency. Physical separation between ECUs in different domains may also limit fault tolerance across domains, as sharing resources in failure conditions is more difficult.

A zonal architecture, on the other hand, organizes ECUs based on their physical location in the vehicle, dividing the system into zones. For instance, the front zone of a vehicle would include ECUs handling headlights, front sensors, and climate control for the front passengers, while the rear zone would include ECUs for tail lights, rearview cameras, and rear passenger controls. Each zone houses ECUs that manage multiple functions from different domains. For example, an ECU in the front zone could handle both body control functions like headlights and safety functions like front sensors. In this approach, redundancy strategies like 2002 DFS or 2003 TMR can be applied within or across zones to improve fault tolerance and ensure continuous operation. One of the key advantages of a zonal architecture is that it reduces wiring complexity, as ECUs in the same zone can communicate and share resources locally, improving efficiency. Real-time communication between different systems within a zone is also faster, as there is less distance and latency in the connections. Moreover, zonal architectures allow for better scalability, as new zones or systems can be added without redesigning the entire vehicle architecture. However, this approach makes it more challenging to specialize ECUs for specific functions, since each ECU within a zone may need to handle multiple tasks across different domains, increasing the complexity of the system. Additionally, ensuring redundancy and fault tolerance for critical systems that span across zones can introduce challenges in managing inter-zone communication and resource sharing.

2.2.4. Dynamic reconfiguration in automated systems. As systems become increasingly automated, dynamic reconfiguration—the ability to adapt the configuration of electronic control units (ECUs) in real time—plays a critical role in ensuring continuous operation [139, 114]. This capability is particularly important in fail-operational systems, where safety and availability are paramount. Dynamic reconfiguration allows the system to respond to faults or failures by redistributing tasks or switching over to backup resources.

In highly automated systems, where there is no human fallback option, dynamic reconfiguration becomes crucial. As, e.g. autonomous vehicles, takes full control over critical functions, the system must have the ability to adapt in real time to any failures or faults [62]. This is especially important in fail-operational systems, where the vehicle must continue functioning even after a component failure, without relying on human intervention. The shift towards fail-operational systems requires transitioning from static E/E-architectures to more dynamic solutions that can handle real-time reconfiguration [114]. Current Electric/Electronic (E/E) architectures typically implement a fail-safe mode, where the driver must intervene in case of system failure. However, as the industry moves towards higher levels of automation (SAE levels 4 and 5), necessitating the development of fail-operational architectures that can dynamically reconfigure to ensure the system remains operational without human intervention. Dynamic reconfiguration is realised through both hardware and software adaptations, allowing ECUs to switch tasks or functions between redundant components. For example, in a fail-operational system for brake and steering control, multiple ECUs work together, and if one ECU fails, another ECU—previously in a hot standby mode—takes over the failed ECU's tasks [114].

Dynamic reconfiguration can be effectively modeled using **Markov processes**, which are suited to describe probabilistic transitions between system states based on failures, reconfigurations, or repairs [35, 59]. An example of a Markov transition diagram of a 2-out-of-3 System with failure rates λ_i and repair rates μ_i can be found in Appendix 2, Fig. 12.

In such a system, the state of each ECU evolves based on transition probabilities that account for different operational conditions. The probability of transitioning from one state i (e.g., fully operational) to another state j (e.g., reconfiguration) at a time t is given by:

(3)
$$p_{ij}(t) = P(\text{transition from state } i \text{ to state } j \text{ at time } t)$$

This transition probability captures how likely the system is to reconfigure itself in response to detected failures

The **reconfiguration time** is a critical factor, defined as the time it takes to detect a fault and switch control to the standby ECU. According to [114], the **fault reaction time** (FRT), which includes both fault detection and reconfiguration time, must be shorter than the **failure tolerance time interval** (FTTI) for the system to maintain safe operation. For example, in a braking system, if the FTTI is 160 ms, the reconfiguration process must occur within that time to prevent hazardous events, as outlined by ISO 26262 standards.

To further quantify this, the system's **transition rate matrix** Q describes the rate of transitioning between states, where q_{ij} represents the rate at which the system moves from state i to state j. For instance, the transition from normal operation to a reconfigured state (following ECU failure) can be described as:

$$q_{ij} = \lambda_{ij} e^{-\lambda_{ij}t}$$

where λ_{ij} is the rate of failure or reconfiguration. The system must then balance between fail-operational and fail-safe states based on real-time conditions.

In practice, dynamic reconfiguration ensures that critical systems like braking and steering remain operational, even in highly automated vehicles. [114] describes a three-ECU system for braking and steering, where the third ECU serves as a backup and is dynamically activated in the event of a failure in one of the primary ECUs. This approach provides both redundancy and flexibility, allowing the system to reconfigure in real time while adhering to strict timing requirements, as outlined in the failure detection Time and reconfiguration time tables provided in [114]. For example, if a nominal steering function is lost, the system must detect the failure within 30 ms and reconfigure within 100 ms to maintain safety and operational integrity.

By leveraging real-time monitoring, predictive maintenance, and Markov-based probabilistic modeling, the system can anticipate and mitigate failures before they escalate into critical issues. This aligns with the fail-operational concepts we have previously discussed, ensuring that the system continues to function even when individual components experience faults.

2.3. Simulation-based reliability analysis. In reliability engineering, simulation techniques are critical tools for modelling, analysing, and improving system reliability [121, 117]. They allow engineers to predict the performance and failure behaviour of complex systems under different operating conditions, without the need for time-consuming or expensive physical testing. Simulation analysis is particularly relevant for safety-critical systems involving humans, where experimentation and testing is not possible for safety, ethical or cost reasons. A typical example is automated driving. In fact, the probability of a traffic accident per

kilometre driven is extremely low, while testing and experimenting new systems in real traffic situations is difficult and expensive [81]. Generally speaking, simulation-aided analysis can help optimise maintenance schedules, component designs, and system architectures for maximum reliability or test different scenarios in a what-if analysis. Simulation techniques are especially relevant for analysing the reliability of dynamic mechanical systems that exhibit wear and fatigue effects, or more generally systems where the failure rates are time-dependent [121, 117, 37, 48]. Indeed, such systems cannot be modelled using Markovian processes and are in general not analytically tractable. Simulations can also handle the complexity of real world systems with multiple interacting components, for instance by identifying critical components, where redesigns can enhance reliability [106, 52]. Simulation techniques are also very useful for scheduling of complex maintenance strategies for repairable systems [34, 151, 42]. They can also incorporate sensor data and be used as digital twins for real-time monitoring of the system health [64, 100, 9, 98, 14].

Simulation techniques in reliability engineering showing particular innovations during the last decades include

- Monte Carlo Simulation (MCS), one of the most widely used simulation techniques in reliability engineering (see the reviews [116, 2, 136]). Monte Carlo techniques are particularly well suited to discrete-event simulation. MCSs use random sampling to simulate the behaviour (typically failure times) of complex systems, especially when analytical solutions are difficult or impossible. For example, they can address the structural reliability of multi-component systems including degradation processes [96, 101] or with interdependent failure rates [13, 79].
- Fault Tree Analysis (FTA) Simulation, a top-down, deductive failure analysis tool used to identify potential causes of system-level failures. FTA methods can be used to quantify the likelihood of different failure scenarios and to identify critical components [133]. FTA is particularly useful in safety-critical systems, such as aerospace or nuclear power, where understanding the root cause of potential failures is important [1, 106, 104], see also the bibliographic review [165].
- Finite Element Method (FEM), a numerical technique widely used in reliability engineering to analyse and predict the behaviour of physical systems described by (nonlinear) partial differential equations. FEM is especially relevant when dealing with complex structures, materials, and loading conditions [48, 41, 88]. FEM allows engineers through numerical schemes to break down complex systems into smaller, simpler parts (finite elements) to study stress, strain, heat distribution, vibration, and other physical phenomena that can affect the reliability of a system.
- Multi-Agent Systems (MAS), to analyse the reliability of complex systems where multiple autonomous components (or subsystems) must work together to maintain system reliability. MAS can dynamically adjust to changes, detect failures, and coordinate repairs or optimisations in a distributed and autonomous manner. They are generally based on a reinforcement learning processes for which model parameters are adjusted through feedback reward mechanisms [168]. Typical applications concern robotics and automated (smart) manufacturing [115, 141, 112]. MAS are especially suited for modelling emergent collective behaviours such as cascading failures [97, 162].
- 2.3.1. Digital twin. In reliability engineering, a digital twin is a dynamic, virtual model of a physical system that uses real-time data, simulation, and analytics to predict system performance and behaviour. The digital twin concept is becoming increasingly important as it enables more accurate, detailed, and real-time analysis of system reliability, including failure prediction, maintenance scheduling, and optimisation of system performance, see the reviews [9, 145, 146]. A digital twin is created using data from sensors, historical information, and models of the physical system, often differential models. This virtual model is continuously updated with real-time data from the physical counterpart, allowing engineers to monitor, analyse, and optimise the physical system in real time. In the context of reliability engineering, digital twins are particularly useful for:
 - Monitoring system health and performance in real time [98],
 - Predicting failures using predictive analytics [100, 14],
 - Optimising maintenance strategies (e.g., predictive maintenance) [42, 151],
 - Improving reliability by simulating and testing different configurations, conditions, scenarios or repair strategies [124, 64, 9].

Applications of digital twins in reliability engineering cover many areas, including aerospace and aviation, manufacturing, energy and power systems, automotive industry, smart cities and infrastructure, etc., see the

reviews [16, 100, 64]. Typical examples are in automotive engineering for accident analysis and reconstruction of traffic accidents. The modelling of the vehicle structure and the deformation due to impacts and collisions can be performed at low cost using continuous physical models and FEM simulation analysis [44, 113]. Another example of applications is the field of automated vehicles and advanced driver assistance systems. Here, the extremely low probability of accidents in traffic and the obvious safety concerns prevent testing and experimentation under real conditions and are generally carried out using numerical simulation, see, e.g., the review [36].

2.3.2. Accelerated life testing. For safety-critical technical products or systems, the term reliability is defined as a characteristic or behavioural trait that indicates the extent to which a specific function is performed in a reliable manner within a given time interval [17]. To determine the exact lifespan for purposes such as maintenance intervals, service policy or optimal operating conditions, it is essential to conduct targeted tests that accurately reflect actual failure behavior. Nevertheless, given that the anticipated product lifespan can span several years, accelerated life testing is employed to facilitate the assessment of the planned service life within a shorter time frame.

As part of the Design of Experiments (DoE), targeted stresses and loads are simulated by adjusting individual or multiple parameters in combination, thereby accelerating the ageing process or degradation. The challenge in simulating degradation or failure behavior lies in replicating or representing the causative factors of damage responsible in reality at an earlier point in time, despite the accelerating factors [156, 135]. A basic distinction is made here between qualitative High Accelerated Testing (HALT) and quantitative Accelerated Testing (ALT). While HALT aims to provoke failure causes as quickly as possible through overloading, ALT is conducted strictly within specified product specification limits, making it transferable to real-world conditions. Although sensor data can be collected during HALT, it only allows for qualitative assessment, whereas ALT can be used for quantitative modelling.

While [61] categorises the possible forms of accelerated life testing in more detail, [156] and [17] provides a methodological overview of the quantitative modelling types depending on the degradation or damage event. Specific applications include, for example, modelling the complex degradation behaviour of corroding lead anodes (see reference [29]), or deriving an acceleration factor (scuffing factor) relevant for the maintenance forecast of products from the construction industry that are designed to last longer than 19 years [45].

2.3.3. Scenario-based simulation. Scenario-based simulation is a widely used approach for evaluating the reliability of complex systems, such as those encountered in safety-critical domains [121, 143]. These systems are typically characterised by numerous interacting components and dynamic behaviour, leading to high-dimensional problems that are often NP-hard. The complexity of such systems can escalate rapidly, making it infeasible to derive general solutions analytically. Consequently, it is essential to constrain the analysis space and focus on identifying specific safety-critical scenarios. This process requires domain expertise to determine which scenarios pose significant risks to the system. For example, in the context of autonomous vehicles, understanding critical traffic situations such as lane changes or interactions at intersections is fundamental for ensuring system safety. Identifying these scenarios enables a targeted evaluation, facilitating more effective safety assessments [142, 143].

In [57], the authors categorise scenario methods into three main groups. Intuitive Logics is a qualitative approach, introduced by [80], relies on expert judgment to develop scenarios without using mathematical models. It focuses on generating diverse future environments to explore strategic alternatives and assess potential impacts, making it suitable for exploratory long-term planning. Cross-Impact Analysis uses probabilistic modeling to analyze interdependencies between influencing factors. Starting from initial probabilities for each factor, it computes how changes in one factor affect the probabilities of others using a cross-impact matrix. Scenarios are then generated based on the highest combined probabilities, making it a mathematically rigorous approach for identifying the most likely future developments. Consistency-Based Approaches evaluate the pairwise consistency between different projections to create scenarios that are logically coherent. By scoring how well different assumptions fit together, it identifies consistent sets of scenarios, ensuring internal alignment and plausibility, but without emphasizing likelihood of occurrence.

3. Challenges and development perspectives

Although sensor-based monitoring, system design and simulation-based reliability analysis have provided important advances and innovations in reliability engineering over the last decades, many challenges remain to be addressed. In this section we suggest some of the more important ones.

- 3.1. Trustworthy AI and real-time data processing. Ensuring that machine learning systems perform as expected, minimising risks and operating reliably under different conditions is crucial in advanced reliability methods. This aspect is commonly referred to as Trustworthy or Safeguarding AI in engineering [82, 25, 118]. With the increasing integration of machine learning algorithms into safety-critical systems (e.g., transportation or healthcare) and industrial automation, failures can lead to catastrophic outcomes. Therefore, it is necessary to quantify confidence and uncertainty in AI predictions and to explore the robustness of the algorithms in different situations, and especially rare ones. Safeguarding AI in reliability engineering includes various strategies and frameworks to ensure safe and consistent performance [146]. The following are some recently developed methods and concepts.
 - Robust and Redundant AI Key considerations for safeguarding AI are robustness, ensuring that AI systems can handle unexpected inputs or disturbances without failing, and redundancy, when incorporating multiple AI algorithms or fallback mechanisms to ensure that if one system fails, another can take over. These aspects remain challenging. They can be addressed through Monte Carlo simulation, fuzzy logic and ensemble learning methods [103, 136], and in system design through redundant, fail-safe, fail-operational or fault-tolerant architectures that drive AI systems to safe or operational states in case of failure.
 - Explainable and Transparent AI The explainability and transparency are also key considerations, by building AI models that are interpretable so that engineers can understand why a certain decision was made, and by ensuring that the development, training, and operation of AI systems are documented [118]. This is especially important in safety-critical applications. Indeed, as AI models become more complex, particularly with the use of ensemble learning or deep learning, their decisions become harder to explain and trust in safety-critical applications. A typical application concern the driving automation where AI algorithms for vehicle perception and motion planning quickly turn out to be black boxes whose operation is difficult to interpret. More research is needed into explainable AI (XAI) techniques to ensure that human operators can understand and trust AI-based systems in high-stakes environments [8, 110, 25].
 - AI Model Validation and Verification Model validation is an important task of AI modules, by ensuring that the AI model performs well under various operational conditions and uncertainties [73]. Model verification, by checking that the AI system adheres to its specified requirements and does not deviate from expected behavior, even in dynamic or unpredictable environments, is also fundamental for trustworthy AI [36, 94].
 - Handling of Rare Events Another challenge is the handling of rare events. Indeed, AI systems often struggle with rare or outlier events, especially in reliability engineering where catastrophic failures are rare but highly impactful [2, 134]. Developing models that can reliably predict rare events is still a challenge. Methods like extreme value theory, enhanced Monte Carlo simulations, or specialised outlier detection algorithms are needed to better handle these edge cases.
 - Scalability and real-time processing of AI Models The scalability of AI models is also especially challenging. In fact, many AI models, such as networks or ensemble methods, are difficult to scale for large, complex systems with many interacting components [171, 11, 118]. There is an important need for more efficient algorithms and computational techniques to handle the large-scale, high-dimensional systems found in industries like aerospace, healthcare, and energy. Another challenge is to make AI model processing operational in real time. This challenge is particularly relevant for smart manufacturing, autonomous vehicles and other safety-critical dynamic systems.
- 3.2. Addressing system complexity. Dealing with system complexity is an important task in reliability engineering. In fact, the operation of complex systems often consists of combination of different modules and components interacting with each other. The system architecture may also be complex and include state and time dependent mechanisms, for instance for redundant, safe tolerant or fail safe designs. To address the reliability of these systems, the complexity of the system must be assessed, taking into account the system

architecture. This task is not straightforward, especially when the system modules and components are interconnected. The interdependence of the component failures, such as common cause failures and cascading failures makes the reliability analysis much more complex. Furthermore, the failure interdependency can drastically affect the system reliability [85], for example for the perception or motion planning of autonomous vehicles [56, 86, 79]. Indeed, dependent components require the system to be considered as a whole (i.e., in a multidimensional framework), since the components cannot be regarded individually. Several models have been developed for this purpose, including lifetime distribution models [169, 111, 83], system state models [109, 93], failure interaction models [22, 46], failure propagation models [32], see also the review [167] and references therein. However, the task remains challenging and is and currently under active development [46, 79].

Another challenge in addressing system complexity relates to the diverse technologies involved in system reliability, particularly for cyber-physical systems. The combination of IoT, AI, robotics and digital control can lead to unforeseen failure modes that traditional reliability methods may not be able to address. Another challenging component of complexity is the collaboration between humans and machines in sociotechnical systems. This aspect is explored below.

- 3.3. Sociotechnical systems. A sociotechnical system can be defined as a system consisting of two interacting components: a technical component and a social component (see, e.g., [148]). The concept emphasises that both components of an overall system have to be addressed when changes are introduced in one part—due to the complex interaction structure (see, e.g., [155]). Two aspects are mainly relevant in sociotechnical theory:
 - (1) Interactions between technical and social subsystems determine the overall system success and
 - (2) Optimising either social or technical component without considering the other part or the interaction can lead to unintended, non-linear relationships diminishing overall system performance [155].

The ideas stem from original coal mining workplace studies, where technological innovations meant to improve working situations and increase performance, but introduced unintended side-effects of changing social structures among the workers [147]. Outcomes were high fluctuation and sick-leaves instead of performance increases [95]. The theory has implications for any change in case of complex systems. Taking a look at the changes of industry 4.0, several aspects related to sociotechnical theory have to be taken into account.

First of all, the increasing digitalisation and interconnectedness through e.g. cyber-physical systems increases the amount of available and relevant information and secondly autonomous and intelligent processes operating in real time, as well as interactions with virtual reality displays create new strain on workers [54]. Automation for example, poses the challenge of changing the role of the workers from active controllers to monitors which in turn leads to degradation in skills that might be necessary when abnormal situations occur [10]. This –along with the notion, that humans are not well suited for prolonged vigilance tasks– has already been formulated in the *ironies of automation* [10]. The idea of compensating this challenge through training and getting the human to adapt to the new system challenges will not produce satisfactory outcomes. Instead, the sociotechnical systems approach shall be applied: before introducing technological innovations simultaneous estimation of social changes and influences on the human and possible countermeasures shall be addressed.

4. Statistical learning methods

In this section we review statistical, learning and simulation methods for advanced reliability techniques of complex systems discussed in Sections. 2 and 3. The methods include unsupervised techniques for data visualisation and clustering in section 4.1. They also cover supervised techniques for prognostics, such as classification, regression and machine learning algorithms in section 4.2. See also [17] for in-depth reliability engineering basics.

4.1. **Unsupervised learning methods.** We first consider a multivariate dataset X consisting of $n \in \{1, 2, ...\}$ observations of $p \in \{1, 2, ...\}$ variables (features)

(5)
$$X = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^p \\ x_2^1 & x_2^2 & \dots & x_2^p \\ \vdots & \vdots & & \vdots \\ x_n^1 & x_n^2 & \dots & x_n^p \end{bmatrix} \in \mathbb{R}^{n \times p}, \qquad \begin{vmatrix} x_i = (x_i^1, \dots, x_i^p) \in \mathbb{R}^p, & i = 1, \dots, n \\ x^j = (x_1^j, \dots, x_n^j)^\top \in \mathbb{R}^n, & j = 1, \dots, p \end{vmatrix}$$

We assume that the variables $(x^1, ..., x^p)$ are dependent, i.e., the multivariate sample X has a structure that we aim to explore. In general, the data are sensor measurements of various physical quantities such as temperature, pressure, vibration, speed (e.g. rotation), noise, etc. The values describe the state of the system, e.g., a machine.

4.1.1. Principal component analysis. There exist many different graphical methods to visualise multivariate datasets, including parallel plots, radar charts, Andrews' plot [6] or again Chernoff's face [28]. However, the dimension of the dataset, i.e. the number of features p observed in the system, can be large, making it difficult to visualise the global structure of the data. In addition, several features may be highly correlated (e.g. noise with rotation speed) and the dimension of the dataset can be reduced without losing a significant amount of information.

Principal component analysis (PCA) can be traced back to the pioneering work of Karl Pearson and, later, Harold Hotelling [63]. The idea is to use a proper orthogonal decomposition of the data, the principal components, that maximises the amount of information. In this way, the first principal component is the best representation of the data in one dimension in terms of variability, the first two components the best representation in two dimensions, and so on. If the original multivariate dataset is highly correlated, the last components contain almost no information and can be neglected. Therefore, the dimensionality of the dataset can be reduced by using only the first components.

Principal components are orthogonal linear combinations of the original variables. Geometrically, we maximise the variability of projections of the data (see Figure 6). Thanks to the Pythagorean theorem, the total variability is conserved and maximising the variability on the axes resumes to minimising the sum of the squared orthogonal distances to the axes. This differs from the technique used in linear regression, which minimises the squared difference between a linear model and the data. For this reason, PCA is sometimes referred to as orthogonal regression.

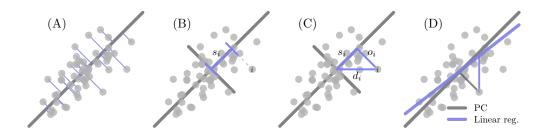


FIGURE 6. Principles of the principal components (example in two dimensions). (A): orthogonal projections of the data. (B): maximisation of the information on the axe $\sum_i s_i^2$. (C): As the total variability $\sum_i d_i^2 = \sum_i s_i^2 + \sum_i o_i^2$ is conserved, maximising $\sum_i s_i^2$ leads to minimising the squared orthogonal distances $\sum_i o_i^2$. (D): Principal component and linear regression generally do not match.

Analytically, the principal components can be determined using basic linear algebra. Let \tilde{X} be the standard score transformation of the data with mean zero and variance one over the p features and $\Gamma = \frac{1}{n}\tilde{X}^{\top}\tilde{X} \in \mathbb{R}^{p \times p}$ the variance/covariance matrix, i.e., $\Gamma_{j,k} = \operatorname{covar}(\tilde{x}^j, \tilde{x}^k)$. Then, thanks to the diagonal formulation $\Gamma = P^{-1}DP$, where $D \in \mathbb{R}^{p \times p}$ is the diagonal matrix of the ordered eigenvalues of Γ and $P \in \mathbb{R}^{p \times p}$ is the

matrix formed by the associated eigenvectors, the principal components are directly the columns of the matrix $\tilde{X}P \in \mathbb{R}^{n \times p}$. An example of application of PCA on measurements of banknotes is given in Appendix 3.

- 4.1.2. Clustering techniques. Clustering techniques are unsupervised data mining methods designed to explore multivariate datasets and to identify patterns and clusters. In principle, the characteristics of the observations in a given cluster are more similar than those of the observations in the other clusters. Clustering techniques can be distinguished according to the following criteria
 - Monitored and unmonitored clustering, whether the number of clusters is known or unknown
 - Strict clustering, where each observation belongs to exactly one cluster
 - Clustering with outliers, where some observations can also belong to no cluster
 - Overlapping clustering, where observations can belong to more than one cluster
 - Fuzzy clustering, where each observation belongs to each cluster to some extent
 - Hierarchical clustering, where observations of a child cluster also belong to the parent cluster
 - Centroid clustering, where clusters are represented by a centroid (mean value)

Generally speaking, non-parametric clustering are empirical methods based on distance metrics between the observations or kernels. Parametric clustering rely on multivariate probabilistic distributions, typically multivariate normal distributions.

K-means clustering. K-means algorithm in a non-parametric monitored strict clustering technique where a cluster is represented by a centroid. The terminology K-means dates from the late 1960s [102] although the method has been used since the 1950s. Consider $(x_1, \ldots, x_n) \in \mathbb{R}^{n,p}$ observations and a partition of clusters $S = \{S_1, \ldots, S_K\}$ where μ_1, \ldots, μ_K are the means on each cluster. Then K-means clustering is the partition that minimises the inter-cluster variability:

(6)
$$S_{K-\text{means}} = \arg\min_{S} \sum_{k=1}^{K} \sum_{i \in S_{k}} ||x_{i} - \mu_{k}||^{2}.$$

In two dimensions, the K-means partition corresponds to a Voronoi diagram. Note that since the total variability is constant for any clustering, minimising the inter-cluster variability is equivalent to maximising the intra-cluster variability.

The K-means problem is computationally NP-hard difficult. However, efficient heuristic algorithms quickly converge to a local optimum. For instance, the **Lloyd-Forgy algorithm** is an iterative process consisting of three steps:

- (1) initialisation: Choose K random mean values $u_1^1, \dots, u_K^1 \in \mathbb{R}^p$.
- (2) Assignment: Each data object is assigned to the cluster with the smallest variance increment.

(7)
$$S_k^t = \left\{ x_i : \left\| x_i - u_k^t \right\|^2 \le \left\| x_i - u_{k^*}^t \right\|^2 \text{ for all } k^* = 1, \dots, K \right\}, \ k = 1, \dots, K.$$

(3) Update: Recalculate the centres of the clusters:

(8)
$$u_k^{t+1} = \frac{1}{|S_k^t|} \sum_{i \in S_k^t} x_i, \qquad k = 1, \dots, K.$$

Steps 2-3 are repeated until the assignments no longer change. Figure 7 shows an illustrative example with n=9 synthetic bivariate observations where the Lloyd-Forgy algorithm converges in three iterations. The number of iterations before convergence depends on the initial mean values chosen. In this example, the achieved local optimum is also the global optimum.

Hierarchical clustering. Hierarchical methods are unmonitored clustering based on tree representations. The top of the tree is a single cluster with all observations while each observation is a cluster in the bottom of the tree (the leaves). Hierarchical methods are based on a distance metric (e.g., Euclidean, Manhattan, Mahalanobis, etc.), and a linkage between two clusters (centroid distance, min/max distance, etc.) from which we can build an objective function. In general, the problem of finding the optimal hierarchical clustering, i.e., the one that minimises a global objective function is NP-hard. This means that as the size of the data grows, finding the exact optimal hierarchical clustering for large datasets becomes computationally infeasible because the number of possible hierarchical clustering grows exponentially with the observation number.

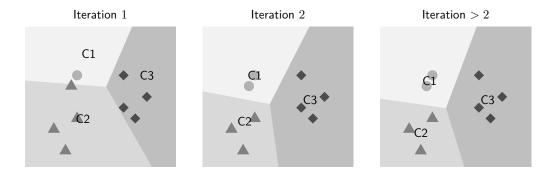


FIGURE 7. Iteration of the Lloyd-Forgy algorithm for the K-means clustering with n=9 synthetic bivariate observations. The algorithm converges in three iterations.

Agglomerative and divisive methods are efficient heuristic algorithms that produce good hierarchical clustering in practice, but they do not guarantee the globally optimal solution due to the NP-hardness of the problem. Agglomerative clustering methods (ACM) are bottom-up approaches, in contrast to top-down divisive methods. In agglomerative methods, each observation is initially a cluster. ACM consists of repeatedly merging the two closest clusters until the number of clusters is greater than one. A dendrogram is a common graphical representation of hierarchical clustering. Dendograms are trees with the linkage distances in the x coordinate (branches) and observations in the y coordinate (leaves). The section of the dendogram allows the number of clusters and the members of each cluster to be determined. A basic ACM with Euclidean distance and centroid linkage for seven synthetic bivariate observations and its dendogram representation is given in Figure 8. Here the shape of the dendogram suggests a configuration with two clusters: $\{A, B, C\}$ and $\{D, E, F, G\}$, or with three: $\{A\}$, $\{B, C\}$ and $\{D, E, F, G\}$, as shown in the figure.

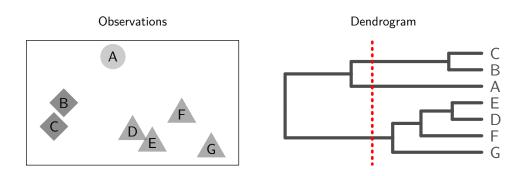


FIGURE 8. Seven bivariate observations (left panel) and their dendogram representation from an agglomerative clustering method with Euclidean distance and centroid linkage, with a section for three clusters: $\{A\}$, $\{B,C\}$ and $\{D,E,F,G\}$.

Density-based clustering. Density-based clustering techniques are non-parametric methods based on empirical data probability density functions, typically using kernel density estimates. These algorithms are designed to identify regions where data points are densely packed and separate them from regions of lower density including noise and outliers. Density-based clustering problems are considered NP-hard in the general case, especially when the goal is to find the globally optimal clustering. However, heuristic methods offer practical solutions without necessarily finding the globally optimal clustering. Popular density-based non-parametric clustering heuristics were developed in the late 1990s, including mean shift [27], DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [43] or OPTICS (Ordering Points to Identify the Clustering Structure) [7] algorithms.

The mean shift algorithm iteratively shifts points towards the mode (peak) of a density function until convergence [27]. Clusters form around these modes. A bandwidth parameter controls the size of the window used to shift points. Similarly, data points are clustered based on the density of points in a neighbourhood with the DBSCAN algorithm [43]. This technique is controlled by two parameters: the neighbourhood search radius and the minimum number of points required to form a dense region (i.e., a cluster). The OPTICS algorithm is an extension of DBSCAN where the radius for neighbourhood radius is adaptive [7]. Oppositely to DBSCAN, OPTICS allows detecting meaningful clusters in data of varying density.

Unlike K-means and, to a lesser extent depending on the metric and linkage used, hierarchical algorithms which mainly identify circular clusters, density-based clustering methods are designed to find clusters of arbitrary shape, including non-convex clusters. In addition, density-based clustering algorithms do not require the number of clusters to be defined. However, density-based models are generally sensitive to parameter values, especially the bandwidth and clustering threshold.

Parametric clustering. Parametric clustering refers to a type of clustering where the data is modelled using a fixed number of parameters, such as the number of clusters or parameters related to the shape of the clusters. Often the data are assumed to follow a particular mixture distribution model, typically multivariate Gaussian distributions. The clusters may have a particular shape, like spherical or ellipsoidal, depending on the parametric model. The main objective is to fit the parameters of the model to the data in a way that best describes the underlying clusters.

For a Gaussian mixture model with multivariate normal probability density function f, the parameters $\theta = (\mu_k, \sigma_k, \pi_k, k = 1, ..., K)$ are, besides the number of clusters K, the expected values μ_k and standard deviations σ_k as well as the proportions of observations per cluster π_k . The likelihood of the model is

(9)
$$\mathcal{L}_{\theta}(X) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k f(x_i, \mu_k, \sigma_k).$$

Local maxima can be obtained using the expectation-maximisation heuristic [107], variational Bayesian methods, or randomised algorithms such as Gibbs sampling. In addition, the number of clusters can be selected using information criteria such as AIC, BIC or likelihood ratio.

4.2. Supervised learning methods. In the context of supervised learning methods the multivariate dataset reads

(10)
$$X = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^p \\ x_2^1 & x_2^2 & \dots & x_2^p \\ \vdots & \vdots & & \vdots \\ x_n^1 & x_n^2 & \dots & x_n^p \end{bmatrix} \in \mathbb{R}^{n \times p}, \qquad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n.$$

The observations X are the set of explanatory variables (sometimes called independent variables, predictor variables, regressors, covariates, features, or input variables), while Y is the variable to be explained (sometimes called the dependent variable, response variable, predicted variable, or output variable), which we assume to be univariate for simplicity. Typically, the data are sensor measurements of various physical quantities such as temperature, pressure, vibration, speed (e.g. rotation), noise, etc. of a machine, and Y is the operational state of the machine or its remaining useful life.

Supervised learning methods are algorithms that aim to predict Y from X. To this end, the parameters of the algorithm are inferred from the data, hence the learning process. Classification methods refer to the case where Y is discrete or categorical, e.g. a class such as the qualitative state of the machine. While regression methods refer to the prediction of continuous numeric quantities, typically remaining useful life. Machine learning methods refer to (nonlinear) algorithms that are flexible enough to be used for both classification and continuous prediction.

A comparison of K-means, AHM and mean-shift clustering algorithms on synthetic bivariate datasets including circular and non-convex clusters can be computed online at

4.2.1. Multivariate regression techniques. We focus first on regression methods where the variable to be predicted is numerical and continuous. The algorithms are models given by

$$(11) y_i = M_{\alpha}(x_i) + \sigma \varepsilon_i,$$

where α is a set of parameters (coefficients) to be calibrated and ε_i is a centred random variable with amplitude σ corresponding to the unexplained part of the models. For instance, a linear model reads

(12)
$$M_{\alpha}(x_i) = \alpha_0 + \alpha_1 x_i^1 + \dots + \alpha_n x_i^n.$$

The objective is to calibrate the parameters in order to fit the data as well as possible, within the limits of the model's shape. In regressive models, the coefficients are estimated by minimising the variability of the residuals $R_i(\alpha) = y_i - M_{\alpha}(x_i)$, i.e. the difference between the model and the observation (see Figure 9). This is

(13)
$$\tilde{\alpha} = \arg\max_{\alpha} \sum_{i=1}^{n} (y_i - M_{\alpha}(x_i))^2.$$

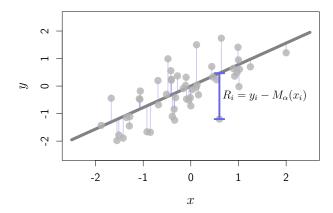


FIGURE 9. Illustrative scheme of a univariate linear regression model. The parameters α are calibrated by minimising the squared difference between the model $M_{\alpha}(x_i)$ and the observation y_i . This corresponds to minimising the variability of the residuals $R_i = y_i - M_{\alpha}(x_i)$.

For linear models, it is easy to check that

$$\tilde{\alpha} = (X^{\top} X)^{-1} X^{\top} Y,$$

while the unbiased estimate of the noise amplitude is given by

(15)
$$\tilde{\sigma}^2 = \frac{1}{n-p-1} \sum_{i=1}^n R_i^2(\tilde{\alpha}).$$

For example, for a univariate linear model M(x) = ax + b, $a, b \in \mathbb{R}$, the least squares estimate is given by $\tilde{a} = \operatorname{covar}(x,y)/\operatorname{var}(x)$ and $\tilde{b} = \bar{y} - \tilde{a}\bar{x}$. This least squares estimate maximises the likelihood if the residuals are independent and identically normally distributed. It is in general a good estimate for compact distributions, but is sensitive to extreme values and outliers. Fisher tests can be used to assess the influence of a single feature, a group of features, or the linear model as a whole. Nonlinear models can be handled by using invertible nonlinear transformations. For instance, linear regression of a log/log transformation $\hat{x} = \log(x)$ and $\hat{y} = \log(y)$ leads to the exponential model

(16)
$$M_{\alpha}(x) = e^{\alpha_0} \cdot (x^1)^{\alpha_1} \dots (x^p)^{\alpha_p}.$$

The logarithmic model

(17)
$$M_{\alpha}(x) = \log(\alpha_0 + \alpha_1 x^1 + \ldots + \alpha_p x^p)$$

can be treated with the exponential transformation $\hat{y} = e^y$. More generally, regressions for differentiable nonlinear models are based on linearisation and iterative heuristics, such as the Gauß-Newton algorithm, which converge to local optima.

The quality of the fit can be assessed by partitioning the total variability SST into the variability explained by the model SSM and the variability of the residuals SSR, where

$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2$$
, $SSM = \sum_{i=1}^{n} (\bar{M} - M_{\tilde{\alpha}}(x_i))^2$, $SSR = \sum_{i=1}^{n} (y_i - M_{\tilde{\alpha}}(x_i))^2$.

Indeed, we have SST = SSM + SSR when the residuals are centred. Therefore, minimising the variability of the residuals leads back to maximising the variability of the model. In addition, the proportion of variability explained

(18)
$$R^2 = \frac{SSM}{SST} = 1 - \frac{SSR}{SST},$$

known as the coefficient of determination, can be used to quantify the quality of the fit. We have $R^2 \in [0, 1]$ for centred residuals. The coefficient of determination is equal to one if the total variance is recovered by the model, while it is equal to zero if the variability of the model is equal to zero. Note that for univariate data, $R^2 = \text{cor}^2(x, y)$.

4.2.2. Classification techniques. In this section, we focus on statistical classification methods where the variable to be predicted is discrete.

Logistic regression. The logistic regression is a binary classification algorithm for which $Y \in \{0, 1\}^n$. The first mention of the logistical model can be traced back to the work of Berkson in 1944 [12]. In the logistic model, the membership of an observation to a class is quantified with the help of the sigmoid (logistic) function

(19)
$$\tilde{y}_i = \pi_i(\alpha) = \frac{1}{1 + \exp\left(-\left(\alpha_0 + \alpha_1 x_i^1 + \dots + \alpha_p x_i^p\right)\right)}.$$

This quantity lies between 0 and 1 with a relatively high transition speed: the logistic function tends quickly to zero if $\alpha_0 + \alpha_1 x_i^1 + \ldots + \alpha_p x_i^p \ll 0$ while it fast tends to one if $\alpha_0 + \alpha_1 x_i^1 + \ldots + \alpha_p x_i^p \gg 0$. The scalars $(\alpha_0, \alpha_1, \ldots, \alpha_p)$ are the p+1 parameters (coefficients) of the logistic model to be estimated and interpreted. Note that binary variables can be generalised to categorical variables when there are more than two possible values, and the binary logistic regression can be generalised to multinomial logistic regression.

The parameters of the binary logistic regression can be estimated by maximising the likelihood

(20)
$$\mathcal{L}(\alpha) = \prod_{i=1}^{n} \pi_i(\alpha)^{y_i} (1 - \pi_i(\alpha))^{1 - y_i}$$

using heuristics, typically gradient methods. In practice, regularisation is often added to the optimisation problem to prevent overfitting, especially when dealing with high dimensional data. The parameters (coefficients) in logistic regression reflect the log-odds of the outcome as a function of the input features. The intercept α_0 represents the log-odds of the outcome when all the input features are zero. If α_0 is large and positive, it indicates that the predicted probability of the outcome is high when all feature values are zero, and vice versa. For a positive coefficient α_j , the feature x^j increases the likelihood of the outcome y=1. The higher the value of the feature, the higher the probability of the positive outcome. The opposite happens for negative coefficients. To make the coefficients easier to interpret, they are often exponentiated to convert them from log-odds to odds ratios.

Linear discriminant analysis. Linear discriminant analysis consists of a linear transformation of the explanatory variables to maximise the inter-class variability or, equivalently, to maximise the intra-class variability. The principle is similar to that of the principal component analysis. The optimal linear transformation can be obtained explicitly by matrix calculus, using the variance/covariance matrix. Linear discriminant analysis deals with binary output data with two classes and can be extended to multi-class analysis [120]. In fine, the binary classification resumes to a linear discriminant

(21)
$$D(\alpha) = \alpha_0 + \alpha_1 x^1 + \dots + \alpha_p x^p.$$

In the standard score transformation of the data, the belonging of an observation to one of the two classes is given by the sign of the discriminant. As in logistic regression, the coefficient estimates allow the influence of a given feature on the class to be interpreted. Note that the linear discriminant analysis is much more sensitive to extreme values and outliers than the logistic regression.

Naive Bayes algorithm. Unlike previous logistic and discriminant classifiers, the naive Bayes algorithm is directly designed for multi-class problems. The classifier is based on the Bayes theorem

(22)
$$\mathbb{P}(Y|X) = \frac{\mathbb{P}(Y)\mathbb{P}(X|Y)}{\mathbb{P}(X)}, \qquad \mathbb{P}(X) > 0,$$

where Y is the (latent) variable to be predicted and X the (observable) explanatory variables. A natural estimation for Y is the value that maximise its probability $\mathbb{P}(Y|X)$ conditionally to the observations. In (22), $\mathbb{P}(X)$ is a constant that does not depend on Y while $\mathbb{P}(Y)$ can be estimated using empirical means \tilde{p}_k . The objective is therefore to estimate the conditional probability $\mathbb{P}(X|Y)$. In the naive Bayes algorithm, the features are assumed to be independent and normally distribution so that the prediction is given by

(23)
$$\tilde{y}_i = \arg\max_k \ \tilde{p}_k \prod_{j=1}^p f(x_i^j, \mu_j^k, \sigma_j^k)$$

with f the normal probability density function. The 2Kp parameters (μ_j^k, σ_j^k) can be estimated by maximum likelihood, i.e., by least square estimates over the classes as the distributions are assumed normal and independent. Similarly to the linear regression, the Bayes classifier is sensitive to extreme values.

Decision tree. Logistic regression, linear discriminant analysis and naive Bayes classifier are linear algorithms providing limited classification for groups whose shape does not allow differentiation by hyperplanes. In contrast, decision trees are based on recursive binary partition rules that can recover nonlinear features. The goal is to split the data in a way that results in the most homogeneous subgroups (in terms of the target variable). As Bayes classifier, they are natural designed for multi-class problem and categorical data. Furthermore, they are easy to interpret and visualise using dendograms, especially for small trees, and don't assume any specific data distribution, making them flexible for various tasks. Decision trees also naturally rank the importance of features based on how often and effectively they are used to split the data. However, they are due to splitting iteration computationally expensive, can lead to overfitting (that can be mitigated by pruning the tree or setting stopping criteria) and can prove unstable (i.e. sensitive to particular observations).

4.2.3. Machine learning techniques. Machine learning techniques are statistical algorithms that can learn from data and generalise to make predictions. However, in contrast to the classical statistical models described above, machine learning algorithms are almost not constrained by the shape of the model [20, 128]. In fact, they contain many more coefficients, giving them great malleability and allowing them to approximate a wide variety of relationships, including complex nonlinear and categorical ones. The learning mechanisms shape the algorithms through the data, empirically identifying the underlying mechanisms at work. Unlike traditional statistical approaches, the predictive power comes from the data rather than the shape of the model. In this sense, machine learning approaches are not based on a theory used to formulate a model. They have no modelling bias and can be applied to a wide range of problems, including classification and regression. However, although their predictive ability can be much more accurate than classical statistical models, they do not allow the influence of the features to be understood directly by interpreting the parameter values, as can be done with models. For this reason, they are often referred to in the literature as black boxes.

The learning processes that allow the data to be fitted and the coefficients to be calibrated are fundamental to machine learning algorithms. This is the model training phase, formulated as a multidimensional optimisation problem. The objective is to minimise a prediction error function, which can be a squared error or an entropy function. The size of the problem, i.e. the number of coefficients to be calibrated, is very large (hundreds, even thousands if not more), so training is performed systematically using iterative heuristics, typically backpropagation methods, which compute the gradient of a loss function with respect to the network weights. Tuning the complexity of the algorithms and their hyperparametrisation is, in the absence of theory, an open problem that needs to be specified according to the input data. Hyperparameters are, for example, the number of layers and neurons in artificial neural networks, the number of support vectors in support vector machines or the number of decision trees in random forests. Although theoretical tools exist for defining the optimal complexity of algorithms, such as the Vapnik-Chervonenkis dimension, they are

often difficult to apply. In practice, determining the optimal complexity of algorithms is usually determined empirically by partitioning the data using cross-validation techniques, where part of the data is used to train the algorithms while the remaining part is used to test the predictions on new data. These operations can be repeated by randomly partitioning the data in training and testing to quantify the uncertainties of the prediction in so-called bootstrapping techniques. If the training error function decreases systematically as the complexity of the algorithms increases, the test error decreases before reaching a minimum and then increases. The minimum corresponds to optimal complexity, the decreasing part to insufficient complexity, known as underfitting, and the increasing part to overparametrisation, known as overfitting, which prevents generalisation.

Artificial neural network. Inspired by the way a brain works, artificial neural networks (ANN) are networks of fully connected neurons organised in layers (see Figure 10, left panel). The first artificial neural network algorithms can be traced back to the pioneering work of Lettvin et al. on the eye and brain of a frog [91]. Each neuron is a activation function (e.g., a sigmoid) on a linear combination of the input, calibrated by weighting coefficients. Since neurons are fully connected, the coefficient number is the dimension of the input plus one times the number of neurons. Using a logistic activation function, a single neuron network is simply a logistic model.

The malleability of the algorithm arises through the superposition of many layers and neurons. In fact, any relationship can approximated using a neural network, provided the network is sufficiently complex. ANNs have found applications in many disciplines, including regression analysis and classification. Different types of architecture are used in practice, depending on the discipline. The simplest networks are feedforward networks in which information propagates in one direction only. Deep neural networks Recursive approaches include loops that can operate at different scales, as in long-short term memory (LSTM) networks. These approaches are well adapted to the analysis of time series. Convolutional neural networks consist of a succession of subsampling of overlapping inputs and convolution, which is particularly well suited to image processing. In addition, networks are said to be deep when the number of layers (and neurons) is large.

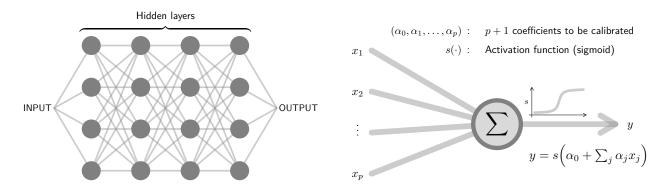


FIGURE 10. Scheme of an artificial feed-forward neural network. Left panel: Layer structure of fully connected neurons. Right panel: Activation operation of a neuron.

Support vector machine. Artifical neural networks consists of a tangle of nonlinear activation functions where the phase of training requires using approximation heuristics. In contrast, support vector machines (SVM) rely on combinations of optimal linear hyperplane in a high dimensional space, making them analytically tractable. The optimisation is carried out by maximising margins between the closest data points of different classes (i.e., the hyperplane and support vectors). It focuses on the points that are hardest to predict. This makes SVMs to be resilient to noisy data, extreme values and outliers unlike classical linear statistical models such as linear discriminant, naive Bayes classifier, or linear regression.

As ANNs, SVMs can be used for both classification and regression [38], although the algorithm was initially designed for classification [15]. Indeed, the used of multiple hyperplane make it possible the fitting of various relationships, whether nonlinear or categorical. In addition, similarly to nonlinear regression models, the input space can be rescaled using nonlinear transformations. This allows the algorithms to fit maximum-margin hyperplane in a transformed feature space that is nonlinear in the original input space and so addressing

complex data structures. Another approach to treat data that is not linearly separable, is to use a technique called the kernel trick projecting the data into higher-dimensional space, where it becomes linearly separable [15]. Common kernels include polynomial, Gaussian and radial basis function kernels. SVMs are especially effective in high-dimensional spaces and when the number of dimensions exceeds the number of samples. They are also memory efficient, as it uses only the support vectors. However, SVMs can be less effective with large datasets because training complexity can become high. In addition choosing the right kernel and hyperparameters (e.g., number of support vectors) can be tricky.

Ensemble learning. Ensemble methods in machine learning are techniques that combine the predictions of multiple algorithms (often referred to as "weak learners" or "base learners") to produce a more accurate and robust prediction than any single algorithm. The key idea behind ensemble methods is that by leveraging the diversity of algorithms, the ensemble can correct for the weaknesses of individual models and improve overall performance. Indeed different models may capture different patterns or aspects of the data, and combining them tends to compensate the weakness and reduce errors. The main types of ensemble methods include bagging methods (bootstrap aggregating), stacking (or stacked generalisation) and boosting.

- Bagging methods [18] reduce variance by training multiple instances of the same algorithm on different random subsets (bootstrapped samples) of the training data, and then averaging their predictions. Random forest is a popular bagging algorithm that trains multiple decision trees and averages their results (for regression) or votes on them (for classification).
- Stacking methods [159, 19] involve training several basic algorithms and then using a *meta-heuristic* to combine their predictions. The meta-heuristic learns how best to aggregate the outputs of the base models. A common setup is to use random forests, SVMs, and neural networks as base models, and then train a linear regression or another model to combine their outputs.
- Boosting methods aim to reduce bias by training algorithms sequentially, focusing on the errors made by the previous algorithms, and progressively improving the prediction. AdaBoost (Adaptive Boosting) [51] and Gradient Boosting [53] are common boosting algorithms. In AdaBoost, misclassified data points are given higher weights in each iteration, while Gradient Boosting iteratively optimises the loss function.

Ensemble learning algorithms are particularly suited to the analysis of complex multivariate datasets. They are widely used in both academic research and industrial applications. Ensemble methods, such as bagging, can help reduce overfitting by averaging out the biases of individual models, leading to a more general and robust prediction algorithm. Boosting techniques can reduce bias by focusing on correcting the errors made by previous algorithms. These two strategies help in balancing the trade-off between bias and variance in machine learning models. However, training multiple algorithms requires more computational resources and time than training a single algorithm. In addition, ensemble methods, combining many algorithms, are difficult to interpret.

Applications of different algorithms for the prediction of pedestrian speeds in single-file movement is given for illustration in Appendix 4.

References

- [1] M. Abdelghany, W. Ahmad, and S. Tahar. Event tree reliability analysis of safety-critical systems using theorem proving. *IEEE Systems Journal*, 16(2):2899–2910, 2021.
- [2] S. S. Afshari, F. Enayatollahi, X. Xu, and X. Liang. Machine learning-based methods in structural reliability analysis: A review. Reliability Engineering & System Safety, 219:108223, 2022.
- [3] S. S. Afshari, C. Zhao, X. Zhuang, and X. Liang. Deep learning-based methods in structural reliability analysis: A review. *Measurement Science and Technology*, 34(7):072001, 2023.
- [4] J. Aldrini, I. Chihi, and L. Sidhom. Fault diagnosis and self-healing for smart manufacturing: a review. *Journal of Intelligent Manufacturing*, 35(6):2441–2473, 2024.
- [5] M. Aly. Survey on multiclass classification methods. Neural Netw, 19(1-9):2, 2005.
- [6] D. F. Andrews. Plots of high-dimensional data. Biometrics, pages 125–136, 1972.

Additionally, multi-algorithm predictions for other datasets including discrete lane-changing maneuvers on highways and RUL aircraft engine forecasting can be computed online at

Multi-algorithm predictions of pedestrian speeds can be computed online at

 $[\]verb|colab.research.google.com/drive/1zGHJQbosu-gZPB8HUafOGVHSJb_hdcbG| \\$

 $[\]verb|colab.research.google.com/drive/1R_p1oJfP_PRke9eYXxRCpknQXTiUuh8W| \\$

 $[\]verb|colab.research.google.com/drive/1KSbi3Lms9hSTx-pnceSOSjltV1XTm7ZU| \\$

- [7] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. OPTICS: Ordering points to identify the clustering structure. ACM Sigmod record, 28(2):49-60, 1999.
- [8] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.
- [9] M. F. Bado, D. Tonelli, F. Poli, D. Zonta, and J. R. Casas. Digital twin for civil engineering systems: An exploratory review for distributed sensing updating. *Sensors*, 22(9):3168, 2022.
- [10] L. Bainbridge. Ironies of automation. In Analysis, design and evaluation of man-machine systems, pages 129–135. Elsevier, 1983.
- [11] M. R. Belgaum, Z. Alansari, S. Musa, M. M. Alam, and M. Mazliham. Role of artificial intelligence in cloud computing, iot and sdn: Reliability and scalability issues. *International Journal of Electrical and Computer Engineering*, 11(5):4458, 2021.
- [12] J. Berkson. Application of the logistic function to bio-assay. Journal of the American statistical association, 39(227):357–365, 1944.
- [13] L. Bian, G. Wang, and P. Liu. Reliability analysis for multi-component systems with interdependent competing failure processes. Applied Mathematical Modelling, 94:446–459, 2021.
- [14] J. Bofill, M. Abisado, J. Villaverde, and G. A. Sampedro. Exploring digital twin-based fault monitoring: Challenges and opportunities. Sensors, 23(16):7087, 2023.
- [15] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.
- [16] D. M. Botín-Sanabria, A.-S. Mihaita, R. E. Peimbert-García, M. A. Ramírez-Moreno, R. A. Ramírez-Mendoza, and J. d. J. Lozoya-Santos. Digital twin technology challenges and applications: A comprehensive review. *Remote Sensing*, 14(6):1335, 2022.
- [17] S. Bracke. Reliability Engineering. Springer, 2024.
- [18] L. Breiman. Bagging predictors. Machine learning, 24:123-140, 1996.
- [19] L. Breiman. Stacked regressions. Machine learning, 24:49–64, 1996.
- [20] L. Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). Statistical science, 16(3):199–231, 2001.
- [21] D. Brüggemann, M. Hinz, and S. Bracke. An application of semi-supervised learning to sparsely labelled data. In M. C. Leva, E. Patelli, L. Podofillini, and S. Wilson, editors, Book of Extended Abstracts for the 32nd European Safety and Reliability Conference, pages 799–806, Singapore, 2022. Research Publishing Services.
- [22] B. Cai, Y. Liu, Z. Liu, X. Tian, X. Dong, and S. Yu. Using bayesian networks in reliability evaluation for subsea blowout preventer control system. *Reliability Engineering & System Safety*, 108:32–41, 2012.
- [23] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová. Machine learning and the physical sciences. Reviews of Modern Physics, 91(4):045002, 2019.
- [24] J. P. Cerrolaza, R. Obermaisser, J. Abella, F. J. Cazorla, K. Grüttner, I. Agirre, H. Ahmadian, and I. Allende. Multi-core devices for safety-critical systems: A survey. ACM Computing Surveys (CSUR), 53(4):1–38, 2020.
- [25] V. Chamola, V. Hassija, A. R. Sulthana, D. Ghosh, D. Dhingra, and B. Sikdar. A review of trustworthy and explainable artificial intelligence (xai). *IEEe Access*, 2023.
- [26] B. Chen. Active Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning. Springer International Publishing and Imprint Springer, Cham, 1st ed. 2012 edition, 2012.
- [27] Y. Cheng. Mean shift, mode seeking, and clustering. IEEE transactions on pattern analysis and machine intelligence, 17(8):790-799, 1995.
- [28] H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American statistical Association*, 68(342):361–368, 1973.
- [29] Christoph Rosebrock. Beitrag zur Prognose der Restnutzungsdauer von komplex beanspruchten Werkzeugen auf der Grundlage einer empirischen Fallstudie zur Korrosion von Bleianoden. Shaker Verlag, 2023.
- [30] Christoph Rosebrock and Stefan Bracke, editors. DETECTION OF DEFECTS ON PRINTED CIRCUIT BOARDS USING INSTANCE SEGMENTATION: Proceedings of the 31st European Safety and Reliability Conference (ESREL 2021). Research Publishing Services, Chennai, 2021.
- [31] Z. M. Çınar, A. Abdussalam Nuhu, Q. Zeeshan, O. Korhan, M. Asmael, and B. Safaei. Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. Sustainability, 12(19):8211, 2020.
- [32] V. da Costa Bueno and I. M. do Carmo. Active redundancy allocation for a k-out-of-n: F system of dependent components. European Journal of Operational Research, 176(2):1041–1051, 2007.
- [33] A. Dardanelli, S. Corbetta, I. Boniolo, S. M. Savaresi, and S. Bittanti. Model-based kalman filtering approaches for frequency tracking. IFAC Proceedings Volumes, 43(10):37–42, 2010.
- [34] B. De Jonge and P. A. Scarf. A review on maintenance optimization. European journal of operational research, 285(3):805–824, 2020.
- [35] A. Djoudi, R. Zitouni, N. Zangar, and L. George. Lora network reconfiguration with markov decision process and fuzzy c-means clustering. *Computer Communications*, 196:129–140, 2022.
- [36] R. Donà and B. Ciuffo. Virtual testing of automated driving systems. a survey on validation methods. IEEE Access, 10:24349-24367, 2022.
- [37] W. Dong, A. R. Nejad, T. Moan, and Z. Gao. Structural reliability analysis of contact fatigue design of gears in wind turbine drivetrains. *Journal of Loss Prevention in the Process Industries*, 65:104115, 2020.

- [38] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. Advances in neural information processing systems, 9, 1996.
- [39] L. Duchesne, E. Karangelos, and L. Wehenkel. Recent developments in machine learning for energy systems reliability management. Proceedings of the IEEE, 108(9):1656–1676, 2020.
- [40] A. Edelmann, T. Wolff, D. Montagne, and C. A. Bail. Computational social science and sociology. Annual Review of Sociology, 46(1):61–81, 2020.
- [41] S. Ereiz, I. Duvnjak, and J. F. Jiménez-Alonso. Review of finite element model updating methods for structural applications. In *Structures*, volume 41, pages 684–723. Elsevier, 2022.
- [42] I. Errandonea, S. Beltrán, and S. Arrizabalaga. Digital twin for maintenance: A literature review. Computers in Industry, 123:103316, 2020.
- [43] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In kdd, volume 96, pages 226–231, 1996.
- [44] S. Evtiukov, E. Golov, and G. Ginzburg. Finite element method for reconstruction of road traffic accidents. Transportation research procedia, 36:157–165, 2018.
- [45] Fabian Hartwig, Marcin Hinz, Severin Dahms, and Stefan Bracke, editors. Auslegung eines Accelerated Life Tests zur Abbildung der Langzeit-Zuverlässigkeit von Produkten im Prototypenstadium: Bericht zur GQW-Jahrestagung 2022 in Chemnitz. Springer Vieweg, Wiesbaden and Heidelberg, 2024.
- [46] R. Fang and X. Li. On allocating one active redundancy to coherent systems with dependent and heterogeneous components' lifetimes. Naval Research Logistics (NRL), 63(4):335–345, 2016.
- [47] C. Ferreira and G. Gonçalves. Remaining useful life prediction and challenges: A literature review on the use of machine learning methods. *Journal of Manufacturing Systems*, 63:550–562, 2022.
- [48] M. Feyzi, K. Fallahnezhad, M. Taylor, and R. Hashemi. A review on the finite element simulation of fretting wear and corrosion in the taper junction of hip replacement implants. Computers in Biology and Medicine, 130:104196, 2021.
- [49] B. Flury. Multivariate statistics a practical approach. Chapman and Hall, London; New York, 1988.
- [50] P. M. Frank and X. Ding. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of process control*, 7(6):403–424, 1997.
- [51] Y. Freund and R. E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.
- [52] J. Friederich and S. Lazarova-Molnar. Reliability assessment of manufacturing systems: A comprehensive overview, challenges and opportunities. *Journal of Manufacturing Systems*, 72:38–58, 2024.
- [53] J. H. Friedman. Greedy function approximation: a gradient boosting machine. Annals of statistics, pages 1189–1232, 2001.
- [54] M. Frost and S. Sandrock. Neue Belastungsarten in der Arbeitswelt 4.0. Zahlen, Daten, Fakten. Technical report, ifaa Institut für angewandte Arbeitswissenschaft, 2018.
- [55] M.-F. Ge, Y. Liu, X. Jiang, and J. Liu. A review on state of health estimations and remaining useful life prognostics of lithium-ion batteries. *Measurement*, 174:109057, 2021.
- [56] H. Gottschalk, M. Rottmann, and M. Saltagic. Does redundancy in ai perception systems help to test for super-human automated driving performance? Deep Neural Networks and Data for Automated Driving: Robustness, Uncertainty Quantification, and Insights Towards Safety, pages 81–106, 2022.
- [57] I. Gräßler, P. Scholle, and H. Thiele. Scenario technique. In S. Vajna, editor, Integrated Design Engineering, pages 615–645. Springer International Publishing, Cham, 2020.
- [58] J. G. Greener, S. M. Kandathil, L. Moffat, and D. T. Jones. A guide to machine learning for biologists. *Nature Reviews Molecular Cell Biology*, 23(1):40-55, 2022.
- [59] I. Häring, Y. Satsrisakul, J. Finger, G. Vogelbacher, C. Köpke, F. Höflinger, and P. Gelhausen. Advanced markov modeling and simulation for safety analysis of autonomous driving functions up to sae 5 for development, approval and main inspection. In M. C. Leva, E. Patelli, L. Podofillini, and S. Wilson, editors, *Proceedings of the 32nd European Safety and Reliability Conference*, pages 104–111. Research Publishing Services, 2022.
- [60] S. A. Hasib, S. Islam, R. K. Chakrabortty, M. J. Ryan, D. K. Saha, M. H. Ahamed, S. I. Moyeen, S. K. Das, M. F. Ali, M. R. Islam, et al. A comprehensive review of available battery datasets, rul prediction approaches, and advanced battery management. *Ieee Access*, 9:86166–86193, 2021.
- [61] M. Hinz, P. Temminghoff, and S. Bracke, editors. Beitrag zur multivariaten Analyse von Qualitäts- und Zuverlässigkeitsdaten in der Produkt-Erprobungsphase am Beispiel der Fahrzeugtechnik: Bericht zur GQW-Jahrestagung 2014 in Chemnitz. Shaker Verlag, 2014.
- [62] J. M. Horcas, J. Monteil, M. Bouroche, M. Pinto, L. Fuentes, and S. Clarke. Context-dependent reconfiguration of autonomous vehicles in mixed traffic. *Journal of Software: Evolution and Process*, 30(4), 2018.
- [63] H. Hotelling. Analysis of a complex of statistical variables into principal components. Journal of educational psychology, 24(6):417, 1933.
- [64] W. Hu, T. Zhang, X. Deng, Z. Liu, and J. Tan. Digital twin: A state-of-the-art review of its enabling technologies, applications and challenges. *Journal of Intelligent Manufacturing and Special Equipment*, 2(1):1–34, 2021.
- [65] X. Hu, L. Xu, X. Lin, and M. Pecht. Battery lifetime prognostics. Joule, 4(2):310-346, 2020.
- [66] R. Isermann. Fault diagnosis of machines via parameter estimation and knowledge processing—tutorial paper. Automatica, 29(4):815–835, 1993.
- [67] R. Isermann. Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance. Springer Science & Business Media, 2006.
- [68] T. Ishigooka, S. Honda, and H. Takada. Cost-effective redundancy approach for fail-operational autonomous driving system. In 2018 IEEE 21st International Symposium on Real-Time Distributed Computing (ISORC), pages 107–115. IEEE, 2018.

- [69] M. M. Islam and J.-M. Kim. Reliable multiple combined fault diagnosis of bearings using heterogeneous feature models and multiclass support vector machines. Reliability Engineering & System Safety, 184:55–66, 2019.
- [70] N. Ivanova and V. Vishnevsky. Application of k-out-of-n:g system and machine learning techniques on reliability analysis of tethered unmanned aerial vehicle. In A. Dudin, A. Nazarov, and A. Moiseev, editors, Information Technologies and Mathematical Modelling. Queueing Theory and Applications, volume 1605 of Communications in Computer and Information Science, pages 117–130. Springer International Publishing, Cham, 2022.
- [71] M. J. and V. G. Bayesian estimation of linear/circular consecutive k-out-of-n: F system reliability. *International Journal of Performability Engineering*, 16(10):1509, 2020.
- [72] R. H. Jhaveri, A. Revathi, K. Ramana, R. Raut, and R. K. Dhanaraj. A review on machine learning strategies for real-world engineering applications. *Mobile Information Systems*, 2022(1):1833507, 2022.
- [73] X. Jiang and S. Mahadevan. Bayesian risk-based decision method for model validation under uncertainty. Reliability Engineering & System Safety, 92(6):707–718, 2007.
- [74] P. Jieyang, A. Kimmig, W. Dongkun, Z. Niu, F. Zhi, W. Jiahai, X. Liu, and J. Ovtcharova. A systematic review of data-driven approaches to fault diagnosis and early warning. *Journal of Intelligent Manufacturing*, 34(8):3277–3304, 2023.
- [75] J. J. M. Jimenez, S. Schwartz, R. Vingerhoeds, B. Grabot, and M. Salaün. Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics. *Journal of manufacturing systems*, 56:539– 557, 2020.
- [76] M. I. Jordan and T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. Science, 349(6245):255–260, 2015.
- [77] T. M. Julitz, A. Tordeux, and M. Löwer. Reliability of fault-tolerant system architectures for automated driving systems. In M. C. Leva, E. Patelli, L. Podofillini, and S. Wilson, editors, Proceedings of the 32nd European Safety and Reliability Conference, pages 120–127, Singapore, 2022. Research Publishing Services.
- [78] T. M. Julitz, A. Tordeux, and M. Löwer. Computer-aided design of fault-tolerant hardware architectures for autonomous driving systems. Proceedings of the Design Society, 3:1047–1056, 2023.
- [79] T. M. Julitz, A. Tordeux, N. Schlüter, and M. Löwer. Reliability of redundant M-Out-Of-N architectures with dependent components: Comprehensible approach with monte carlo simulation. In K. Kołowrocki and E. Dąbrowska, editors, Advances in Reliability, Safety and Secruity. ESREL 2024 Monograph Book Series, volume 4, pages 97–106. Polish Safety and Reliability Association, Gydnia, 2024.
- [80] H. Kahn and A. J. Wiener. The next thirty-three years: A framework for speculation. Daedalus, Vol. 96, 96(3):705-732, 1967.
- [81] N. Kalra and S. M. Paddock. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94:182–193, 2016.
- [82] D. Kaur, S. Uslu, K. J. Rittichier, and A. Durresi. Trustworthy artificial intelligence: a review. ACM computing surveys (CSUR), 55(2):1–38, 2022.
- [83] M. Kelkinnama and M. Asadi. Optimal redundancy allocation in coherent systems with heterogeneous dependent components. *Journal of Applied Probability*, 59(4):1144–1177, 2022.
- [84] A. Kohn, M. Kabmeyer, R. Schneider, A. Roger, C. Stellwag, and A. Herkersdorf. Fail-operational in safety-related automotive multi-core systems. In 10th IEEE International Symposium on Industrial Embedded Systems (SIES), pages 1–4. IEEE, 2015.
- [85] S. Kotz, C. D. Lai, and M. Xie. On the effect of redundancy for systems with dependent components. IIE Transactions, 35(12):1103-1110, 2003.
- [86] K. Kowol., M. Rottmann., S. Bracke., and H. Gottschalk. Yodar: Uncertainty-based sensor fusion for vehicle detection with camera and radar sensors. In Proceedings of the 13th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART, pages 177–186, 2021.
- [87] R. Krishnamoorthy, B. Chokkalingam, and J. L. Munda. Design of fault-tolerant automotive gateway architecture using mc9s12xdp512 microcontroller device. *Energies*, 16(16):5923, 2023.
- [88] J. Kudela and R. Matousek. Recent advances and applications of surrogate models for finite element method computations: a review. *Soft Computing*, 26(24):13709–13733, 2022.
- [89] S. Kumar, R. Saket, D. K. Dheer, J. B. Holm-Nielsen, and P. Sanjeevikumar. Reliability enhancement of electrical power system including impacts of renewable energy sources: a comprehensive review. *IET Generation, Transmission & Distri*bution, 14(10):1799–1815, 2020.
- [90] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin. Machinery health prognostics: A systematic review from data acquisition to rul prediction. Mechanical systems and signal processing, 104:799–834, 2018.
- [91] J. Y. Lettvin, H. R. Maturana, W. S. McCulloch, and W. H. Pitts. What the frog's eye tells the frog's brain. Proceedings of the IRE, 47(11):1940–1951, 1959.
- [92] M. C. Leva, E. Patelli, L. Podofillini, and S. Wilson, editors. Book of Extended Abstracts for the 32nd European Safety and Reliability Conference, Singapore, 2022. Research Publishing Services.
- [93] J. Li, Y. Huang, Y.-F. Li, and S. Wang. Redundancy allocation under state-dependent distributional uncertainty of component lifetimes. Production and Operations Management, 32(3):930–950, 2023.
- [94] X. Li, P. Ye, J. Li, Z. Liu, L. Cao, and F.-Y. Wang. From features engineering to scenarios engineering for trustworthy AI: I&I, C&C, and V&V. IEEE Intelligent Systems, 37(4):18–26, 2022.
- [95] K. Liggieri and O. Müller. Mensch-Maschine-Interaktion. Handbuch zu Geschichte, Kultur und Ethik. Springer, 2019.
- [96] Y.-H. Lin, Y.-F. Li, and E. Zio. A reliability assessment framework for systems with degradation dependency by combining binary decision diagrams and monte carlo simulation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(11):1556-1564, 2015.

- [97] G. Liu, V. Pandey, C. Somarakis, and N. Motee. Risk of cascading failures in multi-agent rendezvous with communication time delay. In 2022 American Control Conference (ACC), pages 2172–2177. IEEE, 2022.
- [98] H. Liu, M. Xia, D. Williams, J. Sun, and H. Yan. Digital twin-driven machine condition monitoring: A literature review. Journal of Sensors, 2022(1):6129995, 2022.
- [99] H.-C. Liu, L.-J. Zhang, Y.-J. Ping, and L. Wang. Failure mode and effects analysis for proactive healthcare risk evaluation: a systematic literature review. *Journal of evaluation in clinical practice*, 26(4):1320–1337, 2020.
- [100] M. Liu, S. Fang, H. Dong, and C. Xu. Review of digital twin about concepts, technologies, and industrial applications. Journal of manufacturing systems, 58:346–361, 2021.
- [101] H. Lyu, H. Qu, H. Xie, Y. Zhang, and M. Pecht. Reliability analysis of the multi-state system with nonlinear degradation model under markov environment. Reliability Engineering & System Safety, 238:109411, 2023.
- [102] J. Macqueen. Some methods for classification and analysis of multivariate observations. In Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press, pages 281–297, 1967.
- [103] Y. A. Mahmood, A. Ahmadi, A. K. Verma, A. Srividya, and U. Kumar. Fuzzy fault tree analysis: a review of concept and application. *International Journal of System Assurance Engineering and Management*, 4:19–32, 2013.
- [104] M. R. Mamdikar, V. Kumar, and P. Singh. Dynamic reliability analysis framework using fault tree and dynamic bayesian network: A case study of npp. Nuclear Engineering and Technology, 54(4):1213–1220, 2022.
- [105] N. V. Martyushev, B. V. Malozyomov, S. N. Sorokova, E. A. Efremenkov, D. V. Valuev, and M. Qi. Review models and methods for determining and predicting the reliability of technical systems and transport. *Mathematics*, 11(15):3317, 2023.
- [106] A. Maurya and D. Kumar. Reliability of safety-critical systems: A state-of-the-art review. Quality and Reliability Engineering International, 36(7):2547–2568, 2020.
- [107] X.-L. Meng and D. Van Dyk. The em algorithm—an old folk-song sung to a fast new tune. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 59(3):511–567, 1997.
- [108] M. Meuwly. Machine learning for chemical reactions. Chemical Reviews, 121(16):10218-10239, 2021.
- [109] J. Mi, Y.-F. Li, W. Peng, and H.-Z. Huang. Reliability analysis of complex multi-state system with common cause failure based on evidential networks. *Reliability Engineering & System Safety*, 174:71–81, 2018.
- [110] S. Narteni, M. Ferretti, V. Orani, I. Vaccari, E. Cambiaso, and M. Mongelli. From explainable to reliable artificial intelligence. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 255–273. Springer, 2021.
- [111] J. Navarro and P. Fernández-Martínez. Redundancy in systems with heterogeneous dependent components. *European Journal of Operational Research*, 290(2):766–778, 2021.
- [112] M. R. Nazabadi, S. E. Najafi, A. Mohaghar, and F. Movahedi Sobhani. The joint policy of production, maintenance, and product quality in a multi-machine production system by reinforcement learning and agent-based modeling. *International journal of research in industrial engineering*, 13(1):71–87, 2024.
- [113] G. Noorsumar, S. Rogovchenko, K. G. Robbersmyr, and D. Vysochinskiy. Mathematical models for assessment of vehicle crashworthiness: a review. *International journal of crashworthiness*, 27(5):1545–1559, 2022.
- [114] F. Oszwald, J. Becker, P. Obergfell, and M. Traub. Dynamic reconfiguration for real-time automotive embedded systems in fail-operational context. In 2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pages 206–209. IEEE, 2018.
- [115] V. O'Neill and B. Soh. Improving fault tolerance and reliability of heterogeneous multi-agent iot systems using intelligence transfer. *Electronics*, 11(17):2724, 2022.
- [116] Z. Qiu, R. Huang, X. Wang, and W. Qi. Structural reliability analysis and reliability-based design optimization: Recent advances. Science China Physics, Mechanics and Astronomy, 56:1611–1618, 2013.
- [117] M. Ram. Modeling and simulation based analysis in reliability engineering, volume 40. CRC Press, 2019.
- [118] S. B. Ramezani, L. Cummins, B. Killen, R. Carley, A. Amirlatifi, S. Rahimi, M. Seale, and L. Bian. Scalability, explain-ability and performance of data-driven algorithms in predicting the remaining useful life: A comprehensive review. *Ieee Access*, 11:41741–41769, 2023.
- [119] Y. Ran, X. Zhou, P. Lin, Y. Wen, and R. Deng. A survey of predictive maintenance: Systems, purposes and approaches. arXiv preprint arXiv:1912.07383, 2019.
- [120] C. R. Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2):159–203, 1948.
- [121] M. Rao and V. Nakain. Review of simulation approaches in reliability and availability modeling. International Journal of Performability Engineering, 12(4):369, 2016.
- [122] C. Reyes, F. Jaramillo, B. Zhang, C. Kulkarni, and M. Orchard. Just-in-time point prediction using a computationally-efficient lebesgue-sampling-based prognostic method: Application to battery end-of-discharge prediction. Annual Conference of the PHM Society, 10(1), 2018.
- [123] R. F. Ribeiro Junior, I. A. dos Santos Areias, M. M. Campos, C. E. Teixeira, L. E. B. da Silva, and G. F. Gomes. Fault detection and diagnosis in electric motors using convolution neural network and short-time fourier transform. *Journal of Vibration Engineering & Technologies*, 10(7):2531–2542, 2022.
- [124] C. Rosebrock and S. Bracke. Simulation of anode corrosion in the electrowinning process of non-ferrous metals. In 2020 Annual Reliability and Maintainability Symposium (RAMS), pages 1–6. IEEE, 2020.
- [125] M. Rumez, D. Grimm, R. Kriesten, and E. Sax. An overview of automotive service-oriented architectures and implications for security countermeasures. IEEE Access, 8:221852–221870, 2020.
- [126] S. Safari, M. Ansari, H. Khdr, P. Gohari-Nazari, S. Yari-Karin, A. Yeganeh-Khaksar, S. Hessabi, A. Ejlali, and J. Henkel. A survey of fault-tolerance techniques for embedded systems from the perspective of power, energy, and thermal issues. IEEE Access, 10:12229–12251, 2022.

- [127] A. K. Samantaray, K. Medjaher, B. O. Bouamama, M. Staroswiecki, and G. Dauphin-Tanguy. Diagnostic bond graphs for online fault detection and isolation. Simulation Modelling Practice and Theory, 14(3):237–262, 2006.
- [128] G. Saporta. Models for understanding versus models for prediction. In COMPSTAT 2008: Proceedings in computational statistics, pages 315–322. Springer, 2008.
- [129] N. Sapoval, A. Aghazadeh, M. G. Nute, D. A. Antunes, A. Balaji, R. Baraniuk, C. Barberan, R. Dannenfelser, C. Dun, M. Edrisi, et al. Current progress and open challenges for applying deep learning across the biosciences. *Nature Communications*, 13(1):1728, 2022.
- [130] B. Sari. Fail-operational Safety Architecture for ADAS/AD Systems and a Model-driven Approach for Dependent Failure Analysis. Springer Fachmedien Wiesbaden, Wiesbaden, 2020.
- [131] T. Schmid, S. Schraufstetter, S. Wagner, and D. Hellhake. A safety argumentation for fail-operational automotive systems in compliance with iso 26262. In 2019 4th International Conference on System Reliability and Safety (ICSRS), pages 484–493. IEEE, 2019.
- [132] S. Schwendemann, Z. Amjad, and A. Sikora. A survey of machine-learning techniques for condition monitoring and predictive maintenance of bearings in grinding machines. *Computers in Industry*, 125:103380, 2021.
- [133] X. Shi and A. M. Bazzi. Fault tree reliability analysis of a micro-grid using monte carlo simulations. In 2015 IEEE Power and Energy Conference at Illinois (PECI), pages 1–5. IEEE, 2015.
- [134] C. Shyalika, R. Wickramarachchi, and A. Sheth. A comprehensive survey on rare event prediction. arXiv preprint arXiv:2309.11356, 2023.
- [135] K. Siebertz, D. van Bebber, and T. Hochkirchen. Statistische Versuchsplanung: Design of Experiments (DoE). VDI-Buch. Springer Vieweg, Berlin and Heidelberg, 2. auflage edition, 2017.
- [136] C. Song and R. Kawai. Monte carlo and variance reduction methods for structural reliability analysis: A comprehensive review. Probabilistic Engineering Mechanics, 73:103479, 2023.
- [137] M. L. H. Souza, C. A. da Costa, G. de Oliveira Ramos, and R. da Rosa Righi. A survey on decision-making based on system reliability in the context of industry 4.0. Journal of Manufacturing Systems, 56:133–156, 2020.
- [138] R. Stetter. Fault-Tolerant Design and Control of Automated Vehicles and Processes, volume 201. Springer International Publishing, Cham, 2020.
- [139] H. Stoll, D. Grimm, M. Schindewolf, M. Brodatzki, and E. Sax. Dynamic reconfiguration of automotive architectures using a novel plug-and-play approach. In 2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops), pages 70–75. IEEE, 2021.
- [140] T. Stolte, S. Ackermann, R. Graubohm, I. Jatzkowski, B. Klamann, H. Winner, and M. Maurer. Taxonomy to unify fault tolerance regimes for automotive systems: Defining fail-operational, fail-degraded, and fail-safe. *IEEE Transactions on Intelligent Vehicles*, 7(2):251–262, 2022.
- [141] J. Su, J. Huang, S. Adams, Q. Chang, and P. A. Beling. Deep multi-agent reinforcement learning for multi-level preventive maintenance in manufacturing systems. Expert systems with applications, 192:116323, 2022.
- [142] J. Sun, H. Zhang, H. Zhou, R. Yu, and Y. Tian. Scenario-based test automation for highly automated vehicles: A review and paving the way for systematic safety assurance. *IEEE transactions on intelligent transportation systems*, 23(9):14088– 14103, 2021.
- [143] S. Tang, Z. Zhang, Y. Zhang, J. Zhou, Y. Guo, S. Liu, S. Guo, Y.-F. Li, L. Ma, Y. Xue, et al. A survey on automated driving system testing: Landscapes and trends. ACM Transactions on Software Engineering and Methodology, 32(5):1–62, 2023
- [144] P. Tchakoua, R. Wamkeue, T. A. Tameghe, and G. Ekemb. A review of concepts and methods for wind turbines condition monitoring. In 2013 world congress on computer and information technology (WCCIT), pages 1–9. IEEE, 2013.
- [145] A. Thelen, X. Zhang, O. Fink, Y. Lu, S. Ghosh, B. D. Youn, M. D. Todd, S. Mahadevan, C. Hu, and Z. Hu. A comprehensive review of digital twin—part 1: modeling and twinning enabling technologies. Structural and Multidisciplinary Optimization, 65(12):354, 2022.
- [146] A. Thelen, X. Zhang, O. Fink, Y. Lu, S. Ghosh, B. D. Youn, M. D. Todd, S. Mahadevan, C. Hu, and Z. Hu. A comprehensive review of digital twin—part 2: roles of uncertainty quantification and optimization, a battery digital twin, and perspectives. Structural and multidisciplinary optimization, 66(1):1, 2023.
- [147] E. L. Trist and K. W. Bamforth. Some social and psychological consequences of the longwall method of coal-getting: An examination of the psychological situation and defences of a work group in relation to the social structure and technological content of the work system. *Human Relations*, 4(1):3–38, 1951.
- [148] E. Ulich. Arbeitspsychologie. Vdf, Hochschulverlag AG an der ETH Zürich, 2001.
- [149] S. R. Vadyala, S. N. Betgeri, J. C. Matthews, and E. Matthews. A review of physics-based machine learning in civil engineering. Results in Engineering, 13:100316, 2022.
- [150] J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer, et al. Applications of machine learning in drug discovery and development. *Nature reviews Drug discovery*, 18(6):463–477, 2019.
- [151] R. van Dinter, B. Tekinerdogan, and C. Catal. Predictive maintenance using digital twins: A systematic literature review. Information and Software Technology, 151:107008, 2022.
- [152] A. J. C. van Gemund and G. L. Reijns. Reliability analysis of k-out-of-n systems with single cold standby using pearson distributions. *IEEE Transactions on Reliability*, 61(2):526–532, 2012.
- [153] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri. A review of process fault detection and diagnosis: Part i: Quantitative model-based methods. *Computers & chemical engineering*, 27(3):293–311, 2003.
- [154] S. vom Dorff, B. Boddeker, M. Kneissl, and M. Franzle. A fail-safe architecture for automated driving. In 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), pages 828–833. IEEE, 2020.

- [155] G. H. Walker, N. A. Stanton, P. M. Salmon, and D. P. Jenkins. A review of sociotechnical systems theory: a classic concept for new command and control paradigms. *Theoretical Issues in Ergonomics Science*, 9(6):479–499, 2008.
- [156] Wayne Nelson. Accelerated Testing: Statistical Models, Test Plans, and Data Analysis. John Wiley & Sons, Inc., 1990.
- [157] P. Weiss, A. Weichslgartner, F. Reimann, and S. Steinhorst. Fail-operational automotive software design using agent-based graceful degradation. In 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), pages 1169–1174. IEEE, 2020.
- [158] Y. Wilhelm, P. Reimann, W. Gauchel, and B. Mitschang. Overview on hybrid approaches to fault detection and diagnosis: Combining data-driven, physics-based and knowledge-based models. *Procedia Cirp*, 99:278–283, 2021.
- [159] D. H. Wolpert. Stacked generalization. Neural networks, 5(2):241–259, 1992.
- [160] R. Wu, D. Kang, Y. Chen, and C. Chen. Assessing academic impacts of machine learning applications on a social science: Bibliometric evidence from economics. *Journal of Informetrics*, 17(3):101436, 2023.
- [161] L. Xing. Reliability in internet of things: Current status and future perspectives. *IEEE Internet of Things Journal*, 7(8):6704–6721, 2020.
- [162] B. Xu, G. Bai, Y. Fang, J. Tao, et al. Failure analysis of unmanned autonomous swarm considering cascading effects. Journal of Systems Engineering and Electronics, 33(3):759-770, 2022.
- [163] Z. Xu and J. H. Saleh. Machine learning for reliability engineering and safety applications: Review of current status and future opportunities. *Reliability Engineering & System Safety*, 211:107530, 2021.
- [164] W. Yan, B. Zhang, G. Zhao, S. Tang, G. Niu, and X. Wang. A Battery Management System With a Lebesgue-Sampling-Based Extended Kalman Filter. IEEE Transactions on Industrial Electronics, 66(4):3227–3236, Apr. 2019.
- [165] M. Yazdi, J. Mohammadpour, H. Li, H.-Z. Huang, E. Zarei, R. G. Pirbalouti, and S. Adumene. Fault tree analysis improvements: A bibliometric analysis and literature review. Quality and Reliability Engineering International, 39(5):1639–1659, 2023.
- [166] J. Yu, S. Zheng, H. Pham, and T. Chen. Reliability modeling of multi-state degraded repairable systems and its applications to automotive systems. Quality and Reliability Engineering International, 34(3):459-474, 2018.
- [167] Z. Zeng, A. Barros, and D. Coit. Dependent failure behavior modeling for risk and reliability: A systematic and critical literature review. *Reliability Engineering & System Safety*, page 109515, 2023.
- [168] K. Zhang, Z. Yang, and T. Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. Handbook of reinforcement learning and control, pages 321–384, 2021.
- [169] Y. Zhang, E. Amini-Seresht, and W. Ding. Component and system active redundancies for coherent systems with dependent components. Applied Stochastic Models in Business and Industry, 33(4):409–421, 2017.
- [170] H. Zhou, X. Huang, G. Wen, Z. Lei, S. Dong, P. Zhang, and X. Chen. Construction of health indicators for condition monitoring of rotating machinery: A review of the research. Expert Systems with Applications, 203:117297, 2022.
- [171] E. Zio. Some challenges and opportunities in reliability engineering. IEEE Transactions on Reliability, 65(4):1769–1782, 2016.
- [172] E. Zio. Prognostics and health management (phm): Where are we and where do we (need to) go in theory and practice. Reliability Engineering & System Safety, 218:108119, 2022.

APPENDIX

Appendix 1: Examples of fail-operational M-out-of-N architectures. We detail two examples of fail-operational M-out-of-N electronic control unit architectures in this appendix: a 2-out-of-2 Dual Fail Safe (DFS) architecture and a Trimodular 2-ouf-of-3 (TMR) architecture.

The 2-out-of-3 (TMR) architecture involves a redundancy scheme where three independent processing paths receive sensor information from the same sources (e.g., camera, radar, lidar), see Fig. 11b. A majority decision is made between the three independent processing paths, and the system continues to operate as long as at least two of the three paths remain functional, ensuring fail-operational behavior. However, a drawback of the 2003 architecture is that all three processing paths typically reside within the same ECU. This makes the system more vulnerable to common-cause failures, such as hardware defects or environmental impacts affecting the entire ECU.

In contrast, the 2-out-of-2 Dual Fail Safe (DFS) architecture consists of two independent fail-safe subsystems distributed across two separate ECUs, see Fig. 11a. These subsystems communicate with each other in isolation, and in the event of a fault, one ECU can take over the functionality of the other. A key advantage of the 2oo2 architecture is the greater ability to implement separation and diversification, as the two ECUs can be physically isolated and use different technologies, which significantly reduces the risk of common-cause failures. However, a drawback is that once one ECU fails, the system is no longer fail-operational and can only maintain fail-safe behavior, requiring restoration of the faulty ECU for full functionality.

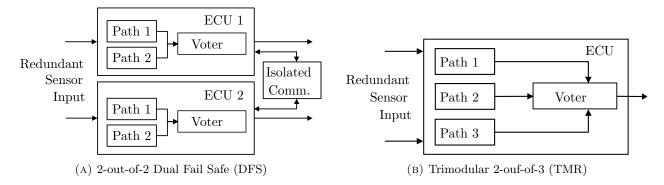


FIGURE 11. Fail-operational M-out-of-N electronic control unit (ECU) architectures [77, 78]

Appendix 2: Markov transition diagram of a 2-out-of-3 system. We present in this appendix the Markov transition diagram of a 2-out-of-3 System with 4 operational modes, see Fig. 12.

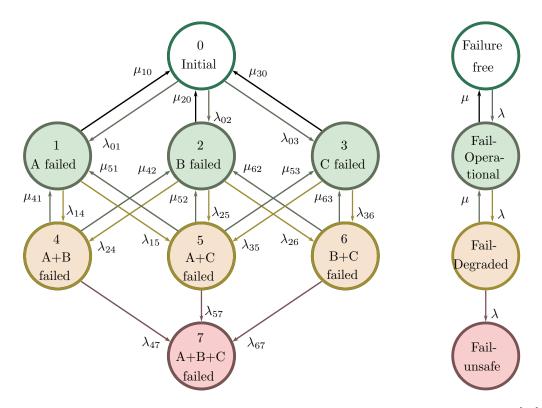


FIGURE 12. Markov transition diagram of a 2-out-of-3 System with 4 operational modes [59]

Appendix 3: Principal component analysis of banknote measurements. As an illustrative example, we analyse the principal components of a dataset of six measurements made on 100 genuine and 100 counterfeit old Swiss 1000 franc banknotes [49] (see Figure 13). The height measurements x^2, \ldots, x^5 are all positively correlated while the banknote width x^1 is more independent. In addition, the diagonal measurement x^6 is negatively correlated with all the height measurements x^2, \ldots, x^5 and slightly positively correlated with the banknote width x^1 .

The pairwise scatter plots of the six measurements are shown in Figure 14. The blue circles are the genuine banknotes, while the grey triangles are the counterfeits. Two clusters of observations corresponding to the

The principal component analysis of the banknote measurements can be computed online at ${\tt colab.research.google.com/drive/1D-Z-1WD_Z74Ts88EQfEjc2W1BLZBdqNw}$

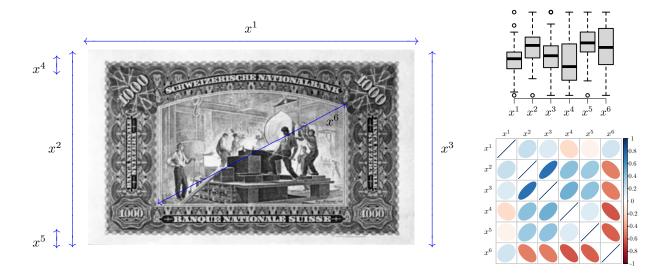


FIGURE 13. Multivariate dataset of six measurements made on 100 genuine and 100 counterfeit old Swiss 1000 franc banknotes [49]. Left panel: six measurements made. Top right panel: Boxplots of the normalised measurements. Bottom right panel: Correlation table of the measurements

genuine and counterfeit notes can be identified, particularly using the diagonal measurement x^6 . However, it is not straightforward to determine whether a given banknote is genuine or counterfeit from the scatter plots. Figure 15 presents some synthetic representations of the datasets using a parallel plot, a radar chart and Andrews' curves, while Figure 16 shows Chernoff's faces of twenty genuine and twenty counterfeit banknotes, specifically in the parallel plot and the radar chart. Different features can be observed for genuine and counterfeit banknotes. However, again, it is not easy to determine from these plots whether a particular banknote is genuine or counterfeit.

Figure 17 shows the principal component analysis of the banknote dataset. The first two principal components contain about 90% of the data variability (top left panel). Therefore, we can reasonably resume the data in two dimensions using the first two components. The correlation table with the original data shows that both first components are correlated with all measurements (bottom left panel). The first component is negatively correlated with all height measures x^2, \ldots, x^5 and positively correlated with the banknote width x^1 and the diagonal x^6 . This component characterises the fact that a banknote is wide rather than long. The second component is mainly negatively correlated with the height of the lower edge x^5 and positively correlated with the height of the upper edge x^4 and the diagonal x^6 . This component rather focuses on the position of the central drawing. Notwithstanding these interpretations, the scatter plot of the first two components (right panel) clearly shows two groups consisting of the genuine banknotes (blue circles) and the counterfeits (grey triangles). Most of the variability in the data is specific to the genuine and counterfeit banknotes, and the representation over the first two components captures this directly.

Appendix 4: Benchmarking pedestrian speed prediction algorithms. Pedestrian trajectory prediction is of great interest, especially for autonomous vehicles. In this appendix we compare the performance of different pedestrian speed prediction algorithms. The data come from a single file experiment where pedestrians walk in a raw on a periodic geometry without overtaking possibility. The explanatory variables are the acceleration, the spacing to the predecessor and its speed, and the spacing of the predecessor with a total of six thousand observations. The predicted variable is the speed of the pedestrian. A summary of the data including scatterplot, boxplot and correlation table is given in Figure 18. Speed is correlated with all explanatory variables except acceleration. In addition, we can identify a nonlinear relationship between speed and distance.

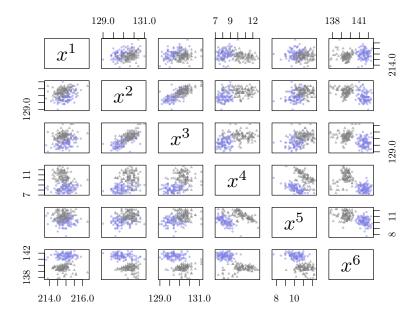


FIGURE 14. Pairwise scatterplots of the six measurements on the notes. The blue circles are the genuine banknotes while the grey triangles are the counterfeits.

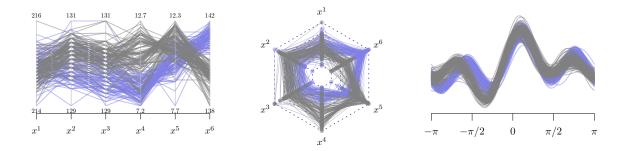


FIGURE 15. Parallel-plots (left panel), radar charts (middle panel) and Andrews' curves (right panel) for the six measurements on the notes. The blue curves are the genuine banknotes while the grey curves are the counterfeits.

A cross-validated benchmark analysis of different pedestrian speed prediction algorithms is performed. The testing prediction results and the \mathbb{R}^2 metric quantifying the goodness of fit are shown in Figure 19. We first consider univariate linear and nonlinear regression before considering multivariate algorithms, including multiple linear regression, a two-layer artificial neural network with three and two neurons respectively, a random forest and a support vector machine. Multivariate algorithms clearly outperform univariate algorithms. Among the multivariate algorithms, the random forest performs best, followed by multiple linear regression, artificial neural network and support vector machine which perform similarly.



FIGURE 16. Examples of Chernoff's faces for the six measurements on the notes. The twenty faces on the left are genuine banknotes, while the twenty grey faces on the right are counterfeits.

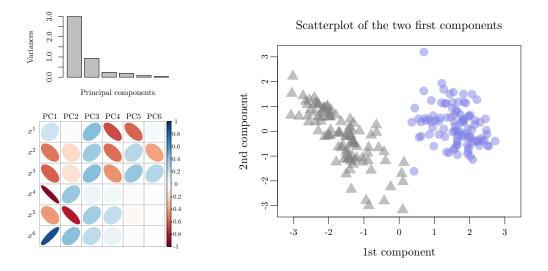


FIGURE 17. Principal component analysis of the banknote dataset. Top left panels: variance per component – the first two components contain most of the variability. Bottom left panel: Correlation table with the original data - the first two components are correlated with all the six measurements. Right panel: scatter plot of the first two components, clearly showing two groups consisting of the genuine banknotes (blue circles) and the counterfeits (grey triangles).

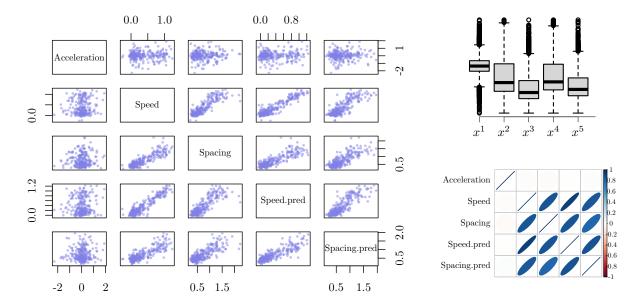


FIGURE 18. Acceleration (in m/s^2), speed (in m/s), spacing (m) and speed and spacing of the neighbour in front of pedestrian in a single-file experiment. Left panel: scatterplots of the features. Top right panel: Boxplots of the normalised measurements. Bottom right panel: Correlation table.

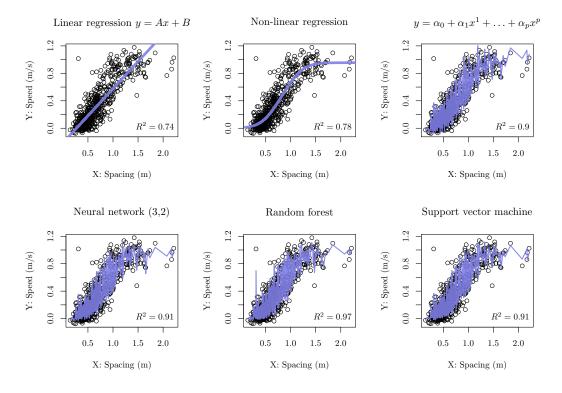


FIGURE 19. Prediction of the speed with different algorithms. For univariate algorithms (linear and nonlinear regressions in top left and middle panels), the speed is regressed on the distance to the neighbour in front only.