
Machine Learning in Fluid Power – Applications, Trends, and Challenges

Faras Brumand-Poor*, Barkha Sharma,
Yannick Duensing, Panagiotis Logothetis,
Emre Atali and Katharina Schmitz

*RWTH Aachen University, Institute for Fluid Power Drives and Systems (ifas),
Campus-Boulevard 30, D-52074 Aachen, Germany*,
E-mail: faras.brumand@ifas.rwth-aachen.de*

**Corresponding Author*

Received 14 October 2024; Accepted 01 December 2025

Abstract

The importance of machine learning (ML) in various engineering disciplines has steadily increased over the past several years, primarily due to the rise in computational power and the development of new, powerful algorithms. ML methods have also found significant applications in fluid power technology, offering substantial benefits such as enhanced system performance, optimized design processes, and improved predictive maintenance. These methods are increasingly used to handle complex, nonlinear systems in fluid power, providing advanced solutions for simulation, system design, control strategies, and condition monitoring. In particular, ML techniques can process vast amounts of data to predict system behavior, identify faults, and optimize energy efficiency, leading to more reliable and efficient fluid power systems. This review aims to introduce and discuss the wide range of ML applications

International Journal of Fluid Power, Vol. 27_2, 359–434.

doi: 10.13052/ijfp1439-9776.2724

© 2026 River Publishers

and ongoing research in fluid power, offering engineers a comprehensive overview of the available literature. By doing so, we aim to help engineers identify and select the most suitable and promising ML methods to address their specific tasks and challenges.

Keywords: Fluid power, machine learning, condition monitoring, data analysis, predictive modeling.

1 Introduction and Motivation

Fluid power Drives and systems are crucial for tasks that require large forces and torques with fast response times. Especially for linear actuation, such drives frequently present the only feasible option for obtaining the necessary force. Furthermore, the absence of electrical components in the drive allows for the use of fluid drives in environmentally challenging situations, such as when working in fields where the drive is likely to come into contact with soil and water. The growing relevance of fluid power across sectors has led to increasing complexity in design, operation, and maintenance. Consequently, engineers tend to categorize the field into specialized areas, including:

- Component and circuit design
- Modern control theory
- Condition Monitoring and fault diagnosis
- Fluid mechanics
- Fluid Properties
- Vibration and noise analysis

Solving challenges in these areas requires a profound understanding of the investigated system and the ability to obtain, process, and analyze data effectively. Particularly, real-world data can introduce challenges in both directions, displaying high dimensionality or limited availability. Additionally, time-series data necessitates thorough processing to ensure that subsequent analysis yields the most optimal conclusions, thereby laying a solid foundation for the application of machine learning (ML). ML is an emerging branch of artificial intelligence that deals with a machine's ability to learn patterns based on data. Using data enables the development of flexible and scalable algorithms. However, since the quantity and quality of the data directly determine the performance of the ML method, caution in its deployment is necessary and not always beneficial. Nevertheless, by utilizing ML methods, fluid power engineers can enhance data-driven approaches

and increase robustness by combining domain knowledge with data in most cases.

One of the primary motivations for integrating ML into fluid power engineering lies in its ability to address several persistent challenges. Traditional models and control systems, while well-established, often lack the flexibility needed to adapt to varying operating conditions and system uncertainties. In contrast, ML offers adaptive capabilities that can accommodate complex and nonlinear behaviors. Moreover, conventional fault diagnosis techniques rely heavily on expert knowledge and predefined rules, which may not generalize well across different systems or fault types. ML-based approaches can overcome these issues by learning directly from data and detecting hidden patterns that often precede failure, which are often missed by human operators or static algorithms.

Beyond that, ML introduces entirely new possibilities that were previously unattainable. It enables engineers to build hybrid models that combine physical principles with data-driven insights, improving the robustness of condition monitoring and supporting real-time decision-making in complex environments. By integrating domain expertise with machine learning techniques, fluid power systems can become not only more efficient but also more intelligent and self-adaptive.

The primary objective of this paper is to provide a comprehensive overview of various utilization approaches in the field of fluid power and to explain the potential benefits of integrating machine learning methods. It is essential to view it holistically to understand the current trends, and it can be a valuable reference for engineers facing specific challenges. Thus, enhancing their ability to engage with ML-driven research and solutions relevant to their task.

The remainder of this paper is organized as follows. Chapter 2 introduces the fundamentals of machine learning. In this chapter, we explain the three main paradigms of machine learning: supervised learning (SL), unsupervised learning (UL), and reinforcement learning (RL). The goal is to provide the reader with the necessary background knowledge for understanding the later chapters.

Chapter 3 focuses on the application of machine learning in the field of fluid power. Building on the areas outlined in the introduction, we discuss how supervised, unsupervised, and reinforcement learning methods are applied in various subdomains of fluid power systems. This chapter also highlights specific characteristics of the fluid power domain that influence the use of ML methods.

Chapter 4 presents a structured overview of existing machine learning studies in the field of fluid power. These studies are categorized according to their application goals, including modeling, control, condition monitoring, and optimization.

Chapter 5 provides comparative insights based on the studies presented in the previous chapter. We analyze and compare the ML methods used, identify patterns and trade-offs, and propose a roadmap to support method selection for different use cases in fluid power.

In Chapter 6, we conclude the work with a summary of key findings. Additionally, we discuss current challenges in applying ML to fluid power systems and outline future research directions and emerging trends in this interdisciplinary field.

2 Fundamentals of Machine Learning

Machine learning is a broad field of algorithms that enable computers to learn from data “without being explicitly programmed” [1].

Figure 1 visualizes the division of ML into its three main learning paradigms **supervised**, **unsupervised learning**, and **reinforcement learning** [2]. Each involves **deep learning**, a subset of ML that considers algorithms that use neural networks with more than three layers. The following subchapters introduce each machine learning paradigm, along with some standard methods and applications. To provide a clear foundation for the subsequent chapters, the following subchapters introduce each of these three paradigms in more detail.

2.1 Supervised Learning

Supervised learning refers to machine learning methods that use labeled data to train a function that can map specific inputs to their correct label [4]. In supervised learning, a model is trained with labeled data to map the input to a desired output. Labeling data often requires manual work from the user, but it enhances the quality of the data by providing initial information. The two predominant tasks in supervised learning are regression and classification. In *classification*, one aims to find a function that categorizes data samples, e.g., images of handwritten digits, into discrete categories. In *Regression*, outputs are predicted as opposed to distinct categories, e.g., prediction of the ambient temperature. Several established supervised learning algorithms are widely used in various applications, including those related to fluid power

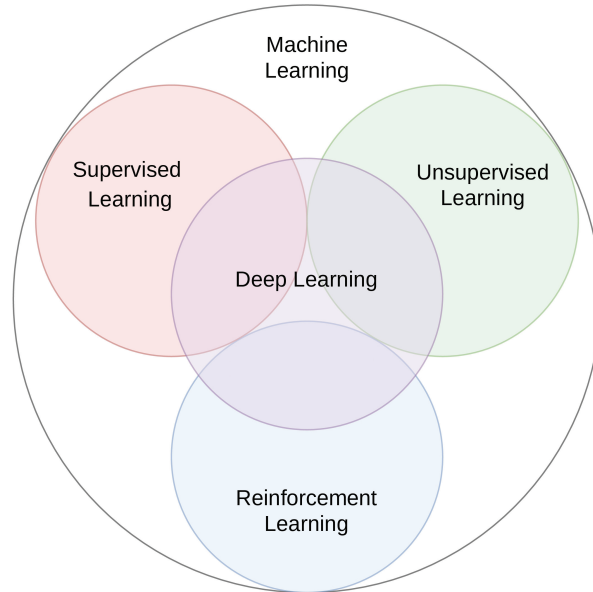


Figure 1 Machine learning can be divided into three paradigms—supervised, unsupervised, and reinforcement learning—while each of them involves *deep learning* as a subdiscipline. Adapted from [3].

systems. For instance, Support Vector Machines (SVM) seek to identify the optimal decision boundary that maximizes the margin between classes. The k-Nearest Neighbors (KNN) classifies new data points based on the majority label of their closest neighbors in the feature space, and Random Forests (RF) combine multiple decision trees trained on random subsets of data to improve predictive accuracy and reduce overfitting. For readers interested in a detailed introduction to these and other algorithms, we recommend *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* by Aurélien Géron [5]. In the following sections, the focus will be on artificial neural networks (ANNs), given their relevance in this work.

Artificial neural networks (ANNs) present one method that is gaining more and more traction. While the term dates back to the early 1940s [6], the advances in computational performance only enabled broader utilization in recent years. The artificial neuron, the perceptron, is inspired by the biological neurons in the human brain. This analogy is referred to in Figure 2. The following part presents the classical neural network, one of the most well-known and promising models of ML. Most importantly, the network can

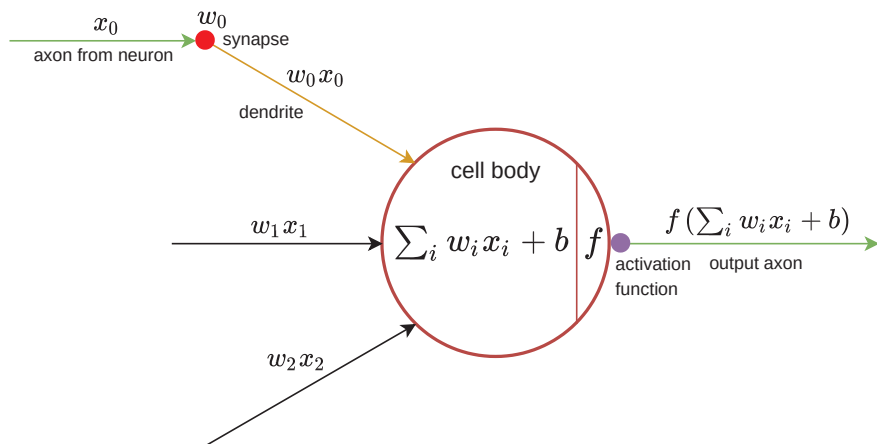


Figure 2 The artificial neuron, or *perceptron*, is inspired by the biological neural cell in the brain. Mathematically, a weighted sum of input values is passed through a nonlinear activation function. Adapted from [7].

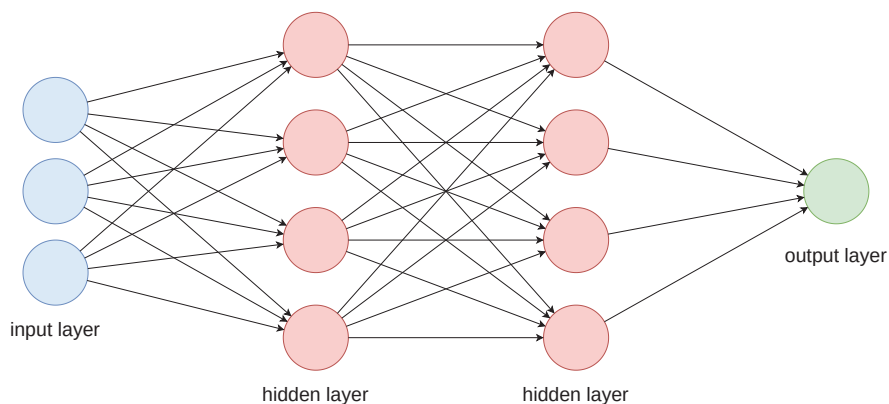


Figure 3 Multiple-layer perceptron (MLP) with fully connected input, no bias neurons, hidden and output layers. Adapted from [7].

be trained using supervised, unsupervised, and reinforcement learning algorithms, and is not bound to any single paradigm. Mathematically, multiple inputs are transformed to a single output: Each scalar input value x_i is multiplied by a scalar weight w_i and summed up, resulting in the weighted sum $\sum_i w_i x_i$. Moreover, a scalar bias b is added, and the result is passed through a nonlinear activation function f , leading to the final equation $f(\sum_i w_i x_i + b)$ of one perceptron [7]. A multi-layer perceptron (MLP) network, as shown in

Figure 3, is obtained if multiple perceptrons are structured into a network. The layers between the input and output layers are referred to as hidden layers. The MLP in Figure 3 is a feedforward ANN with fully connected layers, i.e., there is no feedback in the network, and every neuron in a layer is connected to every neuron in the previous layer. This kind of ANN is a universal function approximator, i.e., an MLP with one hidden layer of sufficient size can approximate any mathematical function to arbitrary accuracy [8]. All the weights and biases in an ANN are the parameters that must be learned during training [4]. The training process consists of two parts. Firstly, the training data samples are used to compute the network's output, which is then compared to the true label of the input data sample within a loss function. The loss function $L(W, b)$ is a function of the weight matrix W , containing all weight parameters of the ANN, and the bias vector b , containing all bias parameters. Secondly, the network parameters are adjusted using gradient descent in an iterative optimization process. More precisely, the weights and biases in the ANN are updated to minimize the loss function, thus reducing the difference between the prediction and the correct label. The update equations according to gradient descent are given as

$$\begin{aligned} W_{ij}^{(l)} &= W_{ij}^{(l)} - \eta \frac{\partial}{\partial W_{ij}^{(l)}} L(W, b) \\ b_i^{(l)} &= b_i^{(l)} - \eta \frac{\partial}{\partial b_i^{(l)}} L(W, b) \end{aligned} \quad (1)$$

where the index (l) indicates the respective layer and the indices i and j indicate the connection between two particular neurons in successive layers [9]. The learning rate η is a parameter that defines the step size of the parameter updates. Choosing a learning rate that is too big can result in slow convergence or even divergence of the optimization process. On the other hand, a small learning rate significantly slows down the training and may result in being stuck in a local optimum. Therefore, adaptive learning rates might be used. In every step, each parameter $W_{ij}^{(l)}$ and $b_i^{(l)}$ is updated in the direction of the negative partial derivative of the loss with respect to the respective parameter. These partial derivatives are computed with the backpropagation algorithm. In particular, state-of-the-art machine learning frameworks, such as *PyTorch*, utilize reverse mode automatic differentiation to determine the partial derivatives efficiently [10, 11].

Typically, the dataset for training and evaluating an ANN is randomly split into chunks of roughly 70%, 20%, and 10% for training, validation, and

testing, respectively [12]. While the data in the training set is used to train the ANN according to the procedure described above, the validation set is used to supervise the performance of the ANN during training and to modify its hyperparameters, such as the number of hidden layers or units per layer. The test data is used to validate the final performance of the ANN on previously unseen data. Eventually, after deploying a trained model, only the forward pass of the network is used to predict outputs for new, unseen inputs.

Depending on the task, other ANN types are deployed. Among the most well-known ANN types are recurrent neural networks (RNNs), which are particularly suitable for time-series data, and Convolutional Neural Networks (CNNs), which are especially well-suited for image data. As mentioned, deep neural networks are often referred to as ANN architectures with multiple hidden layers. Deep learning has proven exceptionally successful in many applications over the past few years, thanks to the increasing availability of data and computational resources [4].

2.2 Unsupervised Learning

Unsupervised learning aims to structure unlabeled data based on its intrinsic features [1]. A model trained with an unsupervised learning algorithm is not provided with labeled data; therefore, it does not know the ground truth of the investigated problems. Instead, a feasible metric is selected to find the optimal parameter for the implemented model. Unsupervised learning has lower requirements compared to supervised learning in terms of the data. A common unsupervised learning task is clustering [4], where data is grouped according to specific resemblances. For instance, k -means minimizes the sum of within-cluster-variations, where variation is calculated regarding distance measures like quadratic Euclidean distance [13]. Depending on the clustering algorithm, the number of clusters has to be set a priori.

Another essential task in unsupervised learning is dimensionality reduction. Here, the goal is to remove redundant information from correlated data and thus represent it in a lower-dimensional space [14]. Principal component analysis (PCA) achieves that by finding principal directions of variation in the investigated data [14]. Another model for this task, involving ANNs, is the autoencoder. The idea is that the autoencoder reconstructs the original data from the low-dimensional space and is optimized based on the difference between the reconstruction and the original data, as shown in Figure 4.

Variational autoencoders (VAEs) are a type of generative model and a specific kind of autoencoder. Next to the compression, the VAE can generate

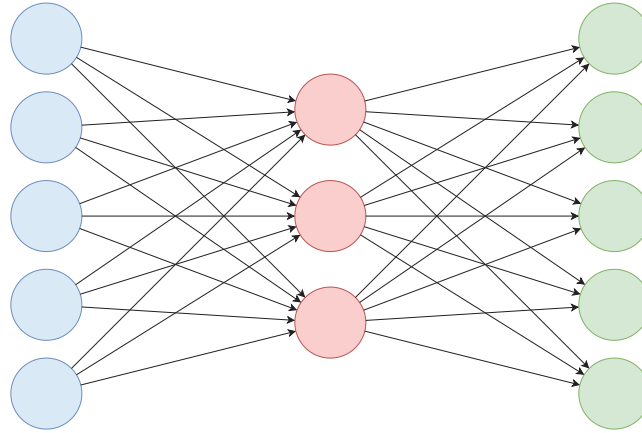


Figure 4 ANN structure of an autoencoder. The hidden layer (red) represents a lower-dimensional embedding of the input data. Adapted from [15].

artificial data by forcing the low-dimensional representation into a mathematical model, a probability distribution, during the training. This distribution can afterward be sampled and generate artificial data [16].

2.3 Reinforcement Learning

While SL and UL involve a somewhat abstract, problem-oriented form of learning, reinforcement learning is based on a more human-like approach: reward-driven learning-by-doing. In particular, the goal is to learn a behavior or policy that enables an agent (decision-maker) to behave optimally in its environment, thereby maximizing the expected cumulative reward. More specifically, the agent remembers which actions led to a high reward in the past trials and consequently repeats these actions with a higher likelihood. The idea embodied by RL is much closer to human learning than other subfields of artificial intelligence. Furthermore, RL is dedicated to efforts towards “simple general principles” in AI [2].

The generality of the RL framework is also reflected by its wide range of related fields. As shown in Figure 5, RL comprises ideas and concepts ranging from engineering to economics, to neuroscience, and many others. Optimal control, particularly dynamic programming for solving stochastic optimal control problems, is one of the leading research fields contributing to modern RL [2]. The other pioneer work was done in the psychological field of animal behavior, particularly trial-and-error learning [2].

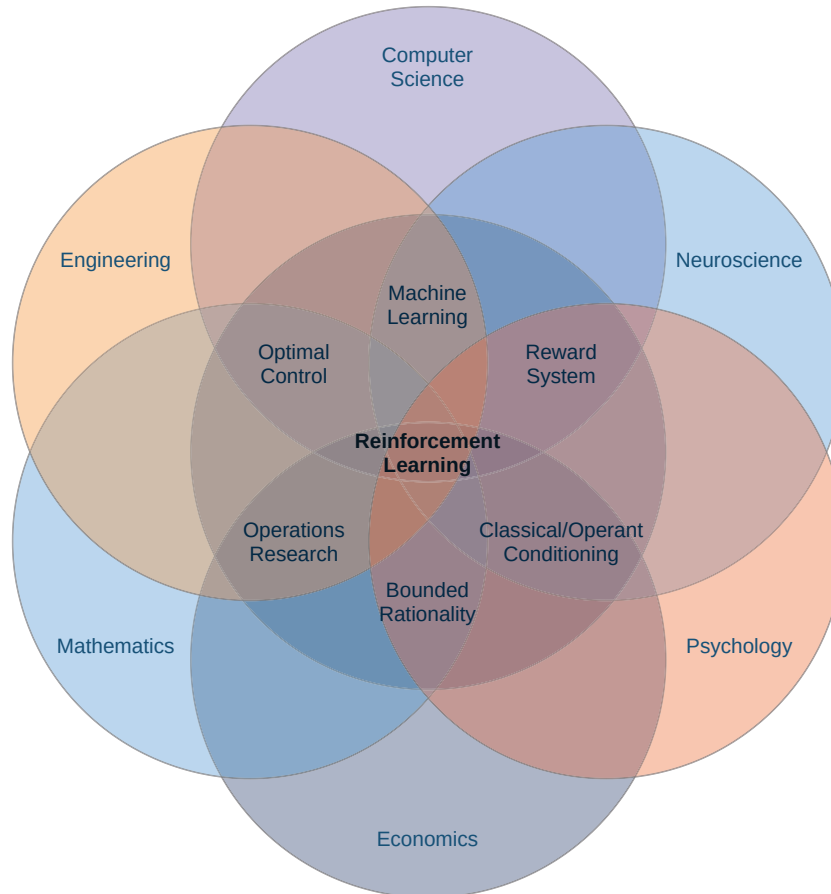


Figure 5 Fields related to reinforcement learning. Adapted from [17].

Two fundamental challenges characterize the RL problem. This first problem is known as the exploration-exploitation dilemma. On the one hand, the agent aims to maximize a reward signal, which means that it always prefers to select an action with a known high reward. On the other hand, the agent should be motivated to explore different actions, which might result in a lower instant reward but a higher cumulative reward in the long run. A stochastic policy is one approach to addressing the exploration-exploitation dilemma. This allows the agent to choose a random action with a small probability of deviating from the most promising one, thus enabling it to explore new interactions while exploiting the most optimal action. The second problem is

known as the curse of dimensionality, which arises from the fact that considering every possible interaction with the environment to find the optimal policy is an impossible task for real-world problems. For example, in a well-defined problem setting like the game of backgammon, it is computationally infeasible to find an optimal solution: With about 10^{20} possible states, it would take thousands of years to compute the optimal solution with the world's fastest computer [2]. This problem is named by Bellman in the context of control theory [18]. One approach to solve this challenge is using ANNs as function approximators [2]. The ANN enables the RL agent to deal with situations similar to previously experienced interactions by generalizing from that experience.

3 Machine Learning in Fluid Power

As mentioned above, the three machine learning paradigms, supervised, unsupervised, and reinforcement learning, have their unique strengths when applied to real-world scenarios. Supervised and unsupervised learning methods are highly regarded for their capacity to uncover hidden patterns within data, patterns that might be invisible to most machine operators or even data analysts. These methods are predominantly utilized in applications such as condition monitoring and form the foundation for predictive maintenance, where data is readily available or continuously generated.

In contrast, reinforcement learning methods typically require an environment, such as a simulation model or operational machinery, for training purposes. They are most commonly employed in preparatory phases before a system's primary operational tasks. These tasks include assisting in the modeling of hydraulic circuits, optimizing system behavior, and handling extensive datasets by learning optimal policies through interaction.

Machine learning techniques thus offer significant potential across various domains in fluid power engineering. The three main learning paradigms play distinct roles depending on the application context. The following subchapters explore their relevance within key areas of fluid power engineering, demonstrating how they contribute to advancements in design, control, monitoring, and analysis.

3.1 Component and Circuit Design

In component and circuit design, supervised learning is widely used to create accurate predictive models based on labeled experimental or simulated data.

For example, neural networks and support vector machines can learn input-output relationships of valves, pumps, and hydraulic circuits, enabling rapid performance prediction and optimization.

Unsupervised learning techniques, such as clustering, help discover patterns in unlabeled data sets from sensor measurements or simulation outputs. This is useful for identifying typical operating modes or detecting unusual behavior during the design phase, which may indicate design flaws or opportunities for improvement.

Reinforcement learning presents opportunities to develop adaptive control strategies for fluid circuits. By interacting with simulation environments, RL agents can learn optimal component configurations or control sequences that maximize performance or efficiency, without requiring explicit system models [19].

3.2 Modern Control Theory

In modern control of fluid power systems, supervised learning methods are employed to model system dynamics when first-principles models are unavailable or incomplete. For instance, artificial neural networks trained on input-output data can serve as accurate system approximators for use in model-based controllers [20].

Unsupervised learning facilitates fault detection and feature extraction from sensor data streams, thereby enhancing controller robustness by enabling the early identification of system changes or degradation without the need for labeled fault data.

Reinforcement learning is compelling for developing adaptive, model-free controllers that improve their policies through trial-and-error interactions with the system or a high-fidelity simulator. RL can optimize control actions in nonlinear, time-varying fluid power processes where classical controllers struggle.

3.3 Condition Monitoring and Fault Diagnosis

Condition monitoring and fault diagnosis benefit significantly from all three ML paradigms. Supervised learning methods, such as support vector machines and random forests, classify known fault conditions from labeled sensor data, enabling the automated identification of specific failures [21].

Unsupervised learning methods, including clustering and autoencoders, are essential when fault labels are scarce or unavailable. These methods detect

anomalies by learning standard operating patterns and flagging deviations, thus supporting early warning systems [22].

Reinforcement learning can optimize maintenance schedules or inspection policies by learning when to intervene based on the evolving condition of fluid power components, balancing operational costs and reliability [23].

3.4 Fluid Mechanics

In fluid mechanics, supervised learning models approximate complex flow behaviors by learning from high-fidelity computational fluid dynamics (CFD) simulations or experimental data, enabling faster predictions of pressure, velocity, or turbulence fields [24].

Unsupervised learning techniques reduce the dimensionality of flow data (e.g., via principal component analysis or autoencoders), facilitating efficient storage, visualization, and analysis of large simulation datasets or measurement campaigns.

Reinforcement learning can guide flow control strategies, such as adjusting actuators to achieve desired flow characteristics, by learning policies that maximize specific objectives like drag reduction or mixing efficiency [25].

3.5 Fluid Properties

Understanding fluid properties under varying conditions is critical in fluid power systems. Supervised learning algorithms can predict properties such as viscosity or density from input variables like temperature and pressure using regression models trained on experimental data [26].

Unsupervised learning helps identify relationships and clusters in complex datasets of fluid properties, aiding in the classification of fluids or the detection of phase changes without labeled data.

Reinforcement learning may assist in optimizing fluid selection or treatment processes by learning decision policies that maximize system efficiency or lifespan based on real-time property measurements.

3.6 Vibration and Noise Analysis

Vibration and noise analysis in fluid power systems relies heavily on ML for condition assessment and fault detection. Supervised learning methods classify vibration signatures associated with specific faults or operating conditions using labeled datasets [27].

Unsupervised learning is used to detect novel or unknown anomalies by learning typical vibration patterns and flagging deviations, which is particularly useful when fault data is limited [28].

Reinforcement learning can be applied to adaptive noise and vibration mitigation strategies by learning control policies that minimize undesired effects in changing operational environments.

4 Machine Learning Applications in Fluid Power

Before we dive into specific application areas, it is essential to distinguish between simulation models and experimental applications, as they form the foundation of most ML studies in fluid power. Simulation models aim to achieve several objectives: reducing development times, decreasing the need for prototype testing, enhancing operational safety, lowering overall development, manufacturing, and operational costs, and conserving resources. Machine-learning methods can accelerate these simulation tasks and thus help produce large data sets in a shorter time. In contrast, experimental applications display deployment on actual test benches or machinery. It is worth noting that this does not exclude simulation if the simulation work is used to test and optimize an application in the real world.

4.1 ML-Based Control Strategies

Machine learning (ML) is increasingly being utilized to control fluid power systems. These approaches can generally be categorized into two groups. First, ML algorithms are embedded within traditional controller architectures to enhance control performance. Second, the control mechanism relies entirely on ML, where the controller itself is a machine learning model. The integration of classical control theory with ML methods offers the advantage of combining established control structures with novel, adaptive techniques.

Simulation-based control applications have laid the groundwork for much of this. For instance, Song et al. pioneered the use of a feedforward multilayer neural network to emulate the inverse dynamics of an excavator. This network is combined with a PID controller and validated for a soil-digging task in a simulation environment. The network takes the three joint positions at the current and two preceding time steps as inputs and produces three torques as outputs. The results indicate that the two-structure controller outperforms a single-structure controller PID controller for high dynamic behavior, fast digging motions, and an increased payload [29]. Similarly,



Figure 6 Movement of the trained excavator model [31].

Hong et al. followed a similar approach to Song [29]. They implemented a control structure with a PID and a neural network to track and control an electro-hydraulic position servo system. The neural network compensated for uncertainties caused by disturbances, resulting in higher tracking accuracy under disturbances compared to the classical PID controller [30].

More advanced studies have shifted towards reinforcement learning (RL). Kurinov et al. utilized Proximal Policy Optimization (PPO) to train a controller for an excavator's loading/unloading tasks, achieving moderate success but noting instability in the learning curve. This agent was trained to move deformable ground to a hopper as shown in Figure 6. The simulation results showed that, on average, the agent moved 67% of the initial mass to the hopper. The author stated that the agent's learning curve still had some fluctuations, which explained the low performance [31].

Building on the reinforcement learning paradigm, Beckman et al. proposed a deep reinforcement learning method for continuously controlling a simulated underground loader. They employed a multi-agent neural network framework, where one agent selected the dig position based on depth camera images, and the other agent controlled the vehicle to load the bucket efficiently. Figure 7 shows the whole framework of the multi-agent controller. The simulation showed that the control structure achieved an average filling rate of 75% [32].

Further contributions include Andersson et al., who implemented a deep reinforcement learning controller for a forestry crane manipulator in a simulation. The PPO agent aimed to pick and place a log while minimizing energy

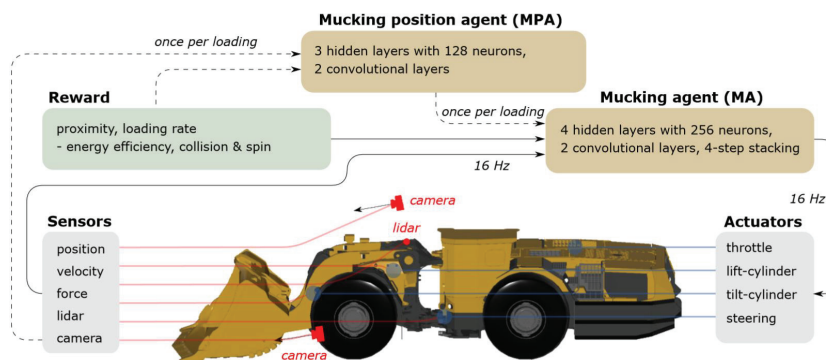


Figure 7 Illustration of the control structure [32]

consumption. This resulted in a slight performance decrease from 97% to 93%, but it significantly reduced energy consumption and provided smoother motion [33].

In a related domain, the real-time prediction of efficient operating points for hydraulic implements in quasi-stationary agricultural processes has been addressed through a combined interaction model and neural network approach [34]. This method enables simulation-based recommendations of operating speeds that optimize system efficiency. Simulation studies demonstrate the potential to propose advantageous operating points, overcoming the limitations of manual operator target adjustments and enhancing adaptive control strategies in agricultural machinery.

Similarly, Hodel et al. compared four different reinforcement learning control algorithms (random search, hill-climbing method, cross-entropy method, and trust region policy optimization) for an excavator tasked with bucket-leveling in an unstructured earthmoving environment. The best-performing algorithm was the trust region policy optimization, which outperformed other methods (Accumulated reward was higher than -1000 compared to the other algorithms, which performed less than -5000). Additionally, a jerk penalty was introduced to avoid excessive control behavior. However, further work is required to operate the controller to different operation modes [35].

In another study, Zhang et al. adopted a two-stage control structure for a hydraulic loading system. The neural network provided an optimal reference for the actual controller, enhancing control performance compared to a single PID controller [36].

Building on the idea of integrating learning-based models within control loops, Brumand et al. took a further step by developing a reinforcement learning algorithm as a controller to solve the classical inverted pendulum problem. In a two-network approach, the actor-critic algorithm, tuned with proximal policy optimization, was implemented, achieving good performance in the upswing and balancing the pole. Subsequent work focused on hyperparameter tuning to further improve the controller's performance [37, 38].

Several studies adopted recurrent neural networks and hybrid control structures. Lizalde et al. implemented a recurrent neural network to represent the model for an actual controller. A second-order sliding mode controller was used for the force-tracking task of an electro-hydraulic actuator. The network successfully modeled the unknown system dynamics, leading to good and robust control performance in simulations [39].

Likewise, Peng et al. utilized RNNs for optimized model predictive control in injection molding. The neural network solved the optimization problem and exhibited enhanced performance compared to a conventional PID controller [40].

In another approach, Aly et al. developed a two-level control structure using neural networks to regulate the flow rate through swash plate displacement of a piston pump. The first network learned the inverse model of the pump from simulations, and the second network served as the actual controller, achieving accurate tracking in simulations [41].

A related simulation-based approach was presented by researchers investigating the energy efficiency of speed- and displacement-variable electro-hydrostatic axes [42]. In their work, a dynamic programming-based optimization algorithm computes loss-optimal pump displacement and speed trajectories for a given duty cycle. The key innovation is the use of a neural network model to represent a radial piston pump, accounting for fluid viscosity across a broad operating range. This enables the inclusion of non-analytic and high-order loss functions within the optimization loop. Simulations across varying viscosities show an average loss reduction of up to 29% for drivetrain components. The ML model ensures accurate system representation, while the dynamic optimizer ensures fast computation times, making this approach a strong example of ML-supported energy-efficient control. This method enhances traditional control strategies and illustrates the growing importance of machine learning in optimizing hydraulic system performance under realistic simulation conditions.

Another simulation-based study addressed the automated selection of optimal working speed in agricultural soil tillage, targeting both process quality and efficiency [43]. In this work, optical sensors capture real-time data about the soil-machine interaction, which is then used to model process quality. At the same time, drivetrain signals are analyzed using a machine learning algorithm that predicts efficiency-related behavior. Both objectives, process quality and process efficiency, are predicted across a range of speed values. The final control decision is derived by combining these into an optimization target and boundary condition to identify the optimal working speed. This ML-based approach replaces manual speed adjustments by the driver and demonstrates how automation can improve both soil processing quality and energy efficiency simultaneously.

Going beyond predictive optimization, researchers from the Karlsruhe Institute of Technology (KIT) investigated how existing control logic for forestry cranes could be transformed into adaptive machine learning models [44]. In their study, an electro-hydraulic flow-on-demand control algorithm was first modeled by an artificial neural network, which was trained to replicate the original behavior. Subsequently, the network was optimized using the Pattern Search algorithm over 41 generations. The resulting ANN controller approximated the performance of the reference controller, while offering a framework for further learning-based refinement. This research illustrates a promising method for converting deterministic control architectures into trainable, stochastic counterparts – particularly relevant for modern hydrostatic drives in forestry and off-road machines.

Similarly, an electro-hydraulic position servo system was controlled using a fuzzy neural network to determine the optimal PID settings. This adaptive controller online modified its time-variant control parameters, resulting in significantly improved control performance compared to a time-invariant PID controller [45].

The transition to real-world implementations has gained momentum. Dhakate et al. employed reinforcement learning for the autonomous control of a hydraulic log manipulator, following a similar approach to Anderson [33]. The authors utilized real-world data to train a simulation model of a similar manipulator, thereby reducing the need for complex kinematic and dynamic mathematical formulations and geometric information for joint mapping in the simulation model. The researchers were able to deploy the simulation policy directly to the real-world manipulator Figure 8 (left) and validated the performance by tracking circular and helical trajectories with minor errors Figure 8 (right).

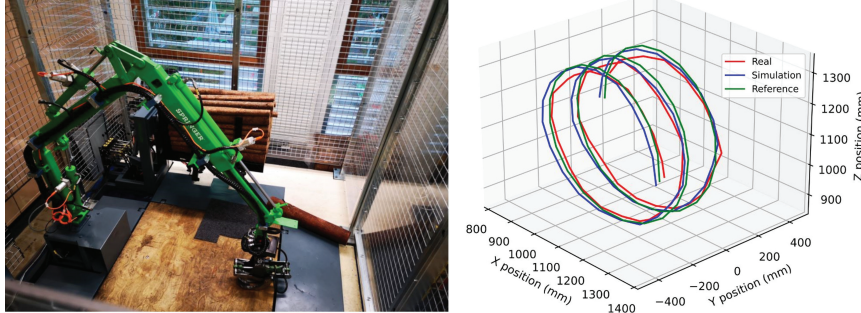


Figure 8 Test environment (left), tracking validation (right) [46].

In a related study, a reinforcement learning-based controller was developed to reduce component damage in wheel loaders by actively damping axle loads during driving maneuvers [47]. The control system applied targeted force pulses via the lifting cylinders to manipulate axle load distribution. Unlike classical control approaches, this method required no descriptive system modeling. Instead, the RL agent learned the behavior through interaction with a validated simulation model. After 850 training runs, the system showed an initial 4.9% reduction in component damage on unseen tracks compared to a passive setup. Although a conventional P-PI controller currently offers a greater reduction, the hybrid approach—combining reinforcement learning with classical control—showed the highest potential for damage reduction. This work highlights the application of reinforcement learning in real-world machine safety and durability optimization, while also demonstrating its current limitations and need for further training and algorithmic improvements.

Complementing this, Yang et al. explored different techniques for managing pile loading in a wheel loader using human control data, employing purely supervised learning methods [48]. The novel architecture contains attention networks, which are used to filter the most critical sensor signals and most relevant control signals. In previous studies [49], a Random Forest (RF) regressor controller demonstrated superior performance over Levenberg-Marquardt (LM) or Bayesian Regularization (BR) backpropagation neural network controllers. Yet, all were trained and tested on the loader under similar summer conditions. When tested again in winter, the results varied drastically. Hence, the investigation was resumed, and a more expressive NN was created. The adaptation of the Attention Neural Network Controller (ANNNet) and the Dual Attention Neural Network Controller (DANNNet)

further improved the controller's performance. Moreover, the quality of sensor data for the control task is investigated, demonstrating the ability to perform the desired pile loading task with data downsampled to 20 Hz from 500 Hz when using DANNet.

In a related study, Egli trained a neural network model based on measurements collected during the excavator's operation to design a controller for the hydraulic actuators of the excavator's arms [50]. RL trained a control policy in the simulation environment after initial training of a behavior model via supervised learning. Lastly, the control policy was deployed on the physical machine. The RL policy directly outputs current setpoints for the excavator's electrically actuated pilot stage valves.

Expanding into energy efficiency, Zhu and Wang propose the use of reinforcement learning for the development of real-time energy management strategies of a hybrid excavator [51]. In the study, conventional energy management strategies, such as thermostats, the equivalent consumption minimization strategy (ECMS), and dynamic programming (DP), are implemented in a simulation environment and compared with their reinforcement learning counterparts. The controller acts as an agent, the action is the motor power, the states are the state of charge (SOC) of the ultracapacitor and the load power, and the reward is low fuel consumption. The agent is trained in the simulation environment and calibrated in the test field to reduce training iterations. DP is assumed optimal during testing, but ECMS and RL show equally good performance. The simulation results are validated on a physical 20t excavator. The thermostat remains the worst performer in terms of energy consumption (fuel economy), and even ECMS falls short compared to RL. This is primarily because the ECMS is cycle-dependent, whereas RL is not. One key advantage of RL is that it can calibrate the RL controller online, further improving its results.

Building on this theme, researchers at the Karlsruhe Institute of Technology (KIT) explored the optimization of control algorithms for forestry cranes by transferring deterministic control laws into neural network-based models [52]. Through iterative optimization with the Pattern Search algorithm, the neural network achieved comparable performance to the original system, demonstrating the feasibility of employing machine learning to modernize and improve hydrostatic function drives in forestry and agricultural machines.

Another real-world application was presented by Karpenko et al., who investigated the potential of using a reinforcement learning neural network (RLNN) as a method for synchronizing the motion of multiple hydraulic manipulators [53, 54]. The task of the manipulators was to move an object

along a predefined trajectory. The RLNN aims to reduce the interaction forces between the manipulators equally. This is performed by adjusting the output of the conventional position control, which generally arises due to imperfect closed-loop actuator dynamics. Initial tests showed that a decentralized control policy would not have been achievable without the communication of the actuators. When communication about the local position error was enabled, the RLNN was capable of reducing the interaction forces with little error in position accuracy.

Focusing on adaptive tracking, Guo et al. presented a different method of adaptive tracking control of a hydraulic actuator. The system's parameters are adjusted simultaneously based on tracking and prediction errors. I.e., the system learns from its tracking performance and past data, making the parameter convergence more effective [55]. Experimental tests were conducted on an electro-hydraulic system with a single-rod actuator controlled by a servo valve. Three controllers were tested and compared: the new composite learning control for nonlinearly parameterized hydraulic systems (CLC-NLP) and a comparison of two classical adaptive controls for both nonlinearly and linearly parameterized hydraulic systems (CLC-NLP and CLC-LP). A prediction error was defined as the deviation of the tracked and predicted cylinder rod displacement. To facilitate a more universal future application, the hydraulic volumes of the system were not accounted for in the control strategy. The results demonstrate the better tracking possibilities of the CLC-NP in direct comparison with the classical ones. The authors elaborate on the advantages of the proposed CLC-NLP, including the elimination of unnecessary input signals such as acceleration and time derivatives of pressure, which ultimately result in increased control robustness. It underlines its superiority over classical control approaches. The proposed CLC approach guarantees parameter convergence in more dynamic situations of excitation compared to conventional CLC approaches.

In a study of deep learning-based control, Kim et al. studied the hydraulic actuator, which an adaptive force controller employs [56]. This controller method is built on a deep neural network inverse model. Data was collected from a test rig and used to train the simulation controller model, making the modeling and tuning obsolete. The learning controller was implemented on a hydraulic actuator hardware platform. Results demonstrate improved tracking performance compared to conventional model-based adaptive control methods.

Kreutmayr and Imlauer investigated the advantages of an optimized PID controller. The test subject was a PID-controlled valve to regulate the pressure

of a hydraulic pump [57]. The controller parameters are often difficult to regulate due to the disturbances caused by the occurring force and oil pressure. A conventional expert-tuned PID controller was used for a performance comparison. The agents for RL are based on the deep deterministic policy gradient (DDPG). Results show that the RL-tuned PID controller outperformed the expert-controlled controller, as it was able to react to unknown system behavior and disturbances. Additionally, applying knowledge gained from a simple simulation model to a real hardware system resulted in minimal performance losses.

Wrywal et al. carried out a similar investigation. Still, instead of using RL to tune the parameters of a PID controller, the authors used an RL algorithm as a substitute for the PID controller [58]. DDPG was used again, this time in conjunction with proximal policy optimization algorithms (PPO), for controlling the output. The presented RL controller demonstrated advantages over the traditional PID controller, as it effectively overcame unwanted behaviors such as overshoot, oscillation, and steady-state errors.

Liam et al. investigated a combination of a feedforward neural network-based PID (FNNPID) and a fuzzy grey predictor (FGP) for controlling an electro-hydraulic actuator (EHA) [59]. The proposed system aims to improve the tracking accuracy and robustness of the control system. Thus, the feedforward neural network is used to compensate for the nonlinearities in the system, while the fuzzy grey predictor-based controller is used to predict the future force of the system. The proposed control system is compared to traditional PID and adaptive fuzzy controllers. Experimental results show that the proposed control system outperforms both in terms of tracking accuracy and robustness. This method promises a viable solution for force control of EHA in industrial applications.

In a distinct application, Shan et al. introduced a pressure control system based on an RL strategy for an electric-pneumatic braking system [60]. The proposed control system utilizes a pneumatic relay system, enabling fast and accurate pressure control. A deep deterministic policy gradient (DDPG) algorithm is employed to train an RL agent to learn an optimal control policy for the system. The DDPG algorithm uses a combination of neural networks and RL to iteratively optimize the control policy through trial-and-error interactions with the system. The performance of the proposed control system is evaluated using a simulation model of the electric-pneumatic braking system, and the results demonstrate that the RL-based control system outperforms a traditional PID control system in terms of settling time and overshoot.

The potential of ML in industrial systems was further explored by Mesmer et al., who presented a machine learning algorithm for the feedforward control of hydraulic clutch actuation path [61], which is needed due to high variability because of manufacturing tolerances and aging effects. A data-based and a model-based feedforward approach is investigated with a high-fidelity simulation model. Performance and computation speed are used as comparison metrics. In the simulation, the model performed better than the data-driven control. Additionally, the computational speed of the model-based approach exceeds that of the data-based approach due to the control updates performed by the back-propagation algorithm. Hence, the model-based approach was deployed on an embedded ECU of a 16-ton wheel loader. To ensure functionality in the presence of noisy signals and external disturbances, the behavior restrictions of the models used in the simulation were relaxed. With these adaptations, a significant improvement in shift quality was achievable after only seven training iterations.

Finally, Besci et al. utilized reinforcement learning algorithms to design a control system for a floating piston pneumatic gearbox actuator in a simulation environment [62] and validated the findings on a test bench [63]. The authors emphasize the importance of balancing both performance and resource needs in the design process. The Policy Gradient-Monte Carlo Tree Search algorithm (PG-MCTS) was introduced, which can adapt the planning length to balance performance and resource demands. The method optimizes the policy parameters directly to maximize the expected cumulative reward. The MCTS is a planning algorithm that builds a tree of possible actions and their corresponding outcomes and then selects actions based on their expected future rewards (Figure 9). The RL algorithm was compared with a PID and a linear–quadratic regulator (LQR) controller. The LQR controller was unable to perform the required shifting task, whereas the PID controller performed even better than the RL algorithm at a constant pressure level. Yet, as soon as the pressure level differs, the agent of the RL algorithm uses its trained knowledge to adjust the control behavior, resulting in superior shifting performance. Still, the tested ECU was not capable of running the algorithm; however, the results emphasize that the PG-MCTS algorithm is considered the most feasible trade-off for the tested application and similar control problems.

An alternative learning model was introduced by Yin et al., who implemented an extreme learning machine (ELM) to control the drive-train torque of wind turbines by predicting the output power of the wind turbine. ELMs are distinguished from traditional neural networks with backpropagation by

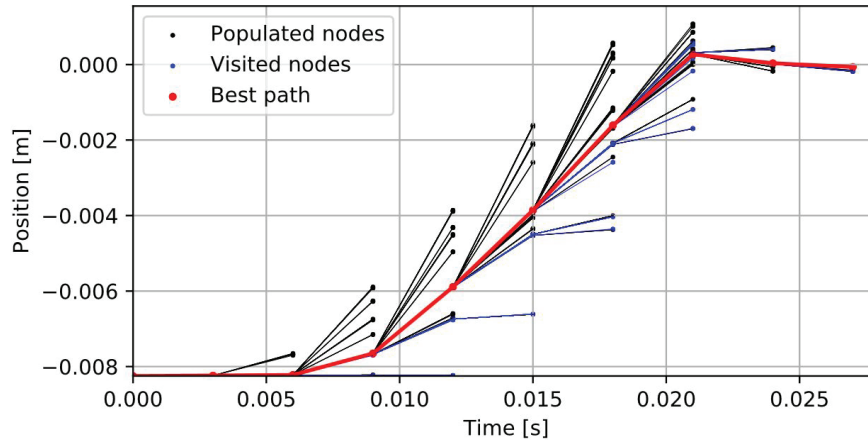


Figure 9 PG-MCTS algorithm example [63].

their simplicity and, thus, training speeds. Instead of iteratively adjusting the network weights, the weights are initialized randomly, and only the output layer weights are updated based on the randomized hidden layer, resulting in faster training speeds. The ELM is trained online, with tip speed ratio and the pitch angle as inputs and power prediction as output [64]. In comparison, the ELM surpassed the traditional controllers tested, as it effectively reduces output power fluctuations.

Table 1 provides a structured overview of simulation-based machine learning approaches in control engineering. It details selected use cases, the applied methodologies, and the corresponding results, thereby reflecting the current research landscape and its experimental foundations. In parallel, Table 2 presents real-world applications, illustrating how machine learning techniques have been successfully implemented in practical control engineering scenarios.

4.2 Condition Monitoring and Fault Diagnosis

In the realm of condition monitoring, ML plays a critical role in increasing system reliability while reducing maintenance costs. The goal is to identify machine operation modes or anomalies through pattern detection in system data.

Initial simulation-based efforts include the work of Theljani et al., who addressed fault diagnosis in a simulation of a three-tank hydraulic system [65]. This system consisted of three tanks, two pumps, and five valves,

Table 1 Simulative application of ML in control

Author	Year	Use Case	Method	Result
Song et al.	1995	Emulate the inverse dynamics of an excavator	Feedforward multilayer neural network + PID	Outperforms single PID (high dynamic, fast digging movement)
Hong et al.	2002	Tracking control of an electro-hydraulic positioning system for an excavator	PID + neural network (latter to compensate for nonlinearities)	Higher tracking accuracy under disturbances than PID alone
Kurinov et al.	2020	Automation of loading and unloading of an excavator	Proximal policy optimization reinforcement learning (PPO)	Low performance due to fluctuations in the learning process
Beckman et al.	2021	Continuous control of a simulated underground mining loader	(Deep reinforcement learning method), multi-agent deep deterministic policy gradient (MADDPG)	Control structure achieved an average filling factor of 0.7
Andersson et al.	2021	Control algorithm imitation and optimization for forestry cranes	Artificial neural network trained and optimized via Pattern Search to replicate electro-hydraulic flow-on-demand control	ANN control behavior comparable to reference control after 41 optimization generations; allows stochastic optimization of deterministic algorithms
Wydra et al.	2020	Forestry crane manipulator/ goal to pick and place logs with reduced cycle time	Deep reinforcement learning controller (PPO agent)	Slight performance decrease but significant reduction in energy consumption
Kazenwadel et al.	2023	Real-time prediction of efficient operating points in agricultural hydraulic systems	Machine learning approach combining interaction modeling with neural networks to optimize system efficiency	Simulations demonstrate the ML-based control suggests advantageous operating points, enabling improved energy efficiency compared to manual operator adjustments

(Continued)

Table 1 Continued

Author	Year	Use Case	Method	Result
Hodel et al.	2018	Excavator tasked with bucket-leveling in an uncertain environment	Comparison of four different reinforcement learning methods	Best-performing: trust region policy optimization
Zhang et al.	2019	Hydraulic loading system	Two-stage control structure (neural network predictive control + PID)	Enhanced control performance compared to single PID
Brumand et al.	2022	Classical inverted hydraulic pendulum	Reinforcement learning algorithm / two-network structure	Good performance in up-swinging and balancing tasks
Peng et al.	2014	Control of an injection molding process	Optimized a model predictive control using a reinforcement learning algorithm	Enhanced performance compared to conventional PID
Aly et al.	2007	Regulation of the flow rate through swash plate of axial piston pump	Two-level control structure using neural networks	Resulted in accurate tracking in simulations
Reidl et al.	2021	Loss optimal control of speed and displacement in electro-hydrostatic axes	Dynamic programming-based optimization algorithm with neural network model of radial piston pump	Simulation shows optimized displacement and speed trajectories reduce average losses by up to 29% in drivetrain components
Kazenwadel et al.	2023	Optimization of process quality and efficiency in agricultural soil tillage	Machine learning algorithm modeling drivetrain relationships combined with optical sensor data for process quality	Enables simultaneous optimization of process quality and efficiency by predicting and aligning targets for working speed
Lizalde et al.	2005	Electro-hydraulic actuator	Recurrent neural network to represent the model of the actuator	Network successfully modeled the unknown system dynamics
H. Chen	2012	Control of an electro-hydraulic position servo system	Fuzzy neural network to determine optimal PID parameters	Greatly improved control performance compared to traditional PID

Table 2 Experimental application of ML in control

Author	Year	Use Case	Method	Result
Dhakate et al.	2022	Autonomous control of a hydraulic log manipulator	RL – Simulation model trained with real-world data	Successful deployment with minor errors in trajectory tracking
Yang et al.	2022	Techniques for managing pile loading in a wheel loader	SL – Attention networks, Random Forest, NN using human control data	Improved controller performance, effective with down-sampled sensor data
Egli	2022	Designing a controller for hydraulic actuators of an excavator	SL, RL – Neural networks	Successfully deployed RL policy on a physical machine, direct current setpoints for pilot stage valves
Zhu and Wang	2022	Real-time energy management strategies for a hybrid excavator	RL – Simulation and physical testing	RL shows superior performance, validated on a 20t excavator, online calibration advantage
Brinschulte et al.	2020	Reduction of component damage in wheel loaders during driving maneuvers	Reinforcement Learning-based control manipulating axle loads via force pulses; trained in simulation	After ≈ 850 training runs, RL showed 4.9% damage reduction on unseen tracks; hybrid with P-PI control shows best potential
Pohlandt et al.	2015	Energy efficiency optimization in electrified off-highway machines	Self-learning control using artificial neural networks to recognize load cycles and adjust system parameters online	Dynamic DC-link voltage adaptation improves efficiency and effectiveness in variable operating conditions
Karpenko et al.	2022	Synchronized motion of multiple hydraulic manipulators	Reinforcement learning neural network (RLNN) – Synchronized motion, communication of local position error	RLNN reduces interaction forces, requires actuator communication for effectiveness
Kim et al.	2022	Hydraulic actuator controlled by an adaptive force controller	Deep Neural Network – Training with test rig data	Improved tracking performance over conventional model-based adaptive control methods

(Continued)

Table 2 Continued

Author	Year	Use Case	Method	Result
Kreutmayr and Imlauer	2022	Optimizing a PID controller for regulating pressure in a hydraulic pump	Reinforcement learning (DDPG)	RL-tuned PID controller shows superior performance, effective in reacting to disturbances, minimal performance loss from simulation to real system
Wrywal et al.	2022	RL algorithm as a substitute for PID controller	Reinforcement learning (DDPG, PPO)	RL controller outperforms traditional PID controller, overcomes overshoot, oscillation, and steady-state errors
Liam et al.	2022	Controlling an electro-hydrostatic actuator (EHA)	Feedforward neural network, fuzzy grey predictor – Comparison with traditional PID and adaptive fuzzy controllers	Improved tracking accuracy and robustness, viable solution for industrial applications
Shan et al.	2022	Pressure control in an electric-pneumatic braking system	Reinforcement learning (DDPG)	RL-based control system outperforms traditional PID control in settling time and overshoot
Mesmer et al.	2022	Feedforward, control of hydraulic clutch actuation	Data-based and model-based feedforward control – High-fidelity simulation model	Model-based approach superior in performance and computation speed, successfully deployed on 16-ton wheel loader, significant improvement in shift quality
Besci et al.	2022	Control system of a floating piston pneumatic gearbox actuator	Policy Gradient-Monte Carlo Tree Search (PG-MCTS), PID, LQR – Simulation and test bench validation	RL algorithm adjusts to varying pressure levels, superior shifting performance

(Continued)

Table 2 Continued

Author	Year	Use Case	Method	Result
Yin et al.	2022	Drive-train torque control in wind turbines	Extreme learning machine (ELM) – Predicting output power	ELM surpasses traditional controllers, effectively reduces output power fluctuations, fast training speeds

operating in six modes: one normal mode, four faulty modes (corresponding to valve closures), and one mode induced by pump failure. The authors employed a semi-supervised classification algorithm that combined an unsupervised clustering algorithm and a supervised multi-layer perceptron model. The unsupervised algorithm clustered simulation data into predefined operation modes, automating label generation. The results demonstrated successful clustering and label generation despite some delay in fault prediction.

This foundational emphasis on simulation has also been reflected in efforts that integrate hybrid ML frameworks for pneumatic systems. One such method applied the operating point strategy to monitor energy efficiency and robustness across various operating states [66]. By incorporating regression and classification models, the system autonomously adjusted key parameters, such as supply pressure, to enhance stability under dynamic conditions.

Building on neural-network-based approaches, Ali implemented a Kohonen neural network to identify fault symptoms in internal gear pumps by analyzing variations in torque loss. An UL algorithm mapped operational states to fixed nodes, detecting changes during operation. A second stage linked faults to torque loss variations, improving the algorithm's performance. The ML pipeline was validated in a simulation with increased bearing friction, successfully detecting and attributing the fault [67].

In contrast to these unsupervised or hybrid approaches, Castilho et al. addressed fault detection using two supervised learning algorithms. They aimed to identify failure modes in a pneumatic pressure relief and shutoff valve system, utilizing a support vector machine and a classification and regression tree. A simulation of the valve generated the required dataset, covering a healthy mode and three faulty modes: leakage between valve chambers, degraded return spring, and excessive piston friction. Both supervised learning models used valve current and angle data to determine the operation mode. The results favored the support vector machine for its higher

precision, especially in minimizing false negatives and false positives while acknowledging the potential choice between the algorithms based on specific use cases [68].

Taking a similar direction but with an added experimental perspective, another approach sought to optimize pneumatic actuator behavior by analyzing real-time pressure data [69]. This hybrid ML model is not only optimized for energy efficiency but also adapted to fluctuating load conditions. Experimental results supported the model's effectiveness in reducing air usage while maintaining actuator robustness.

Building on this setup, Sano et al. continued fault detection in pneumatic pressure relief and shutoff valve systems using supervised learning [70]. They evaluated five different algorithms: k-nearest neighbors, Naive Bayes, classification and regression trees, support vector machines, multilayer perceptrons, and multinomial logistic regression. These algorithms were tested on the same simulation as in Castilho's work, with the addition of multi-fault scenarios. All algorithms achieved a mean accuracy of approximately 99% in single-fault cases, while the Support-Vector machine outperformed others in multi-fault cases with a mean accuracy of 94.28%.

Complementing the discussion on classification accuracy, the detection of production-induced anomalies in external gear pumps has been addressed through a blend of simulation and experimental techniques [71]. Using AI in tandem with numerical tools, the researchers identified geometric inconsistencies, such as conicity and axial misalignments, that often remain undetected. Validation across eight physical units confirmed the potential of this method for enhancing component reliability.

In another simulation-based study, Gareeve et al. explored machine-learning algorithms for hydraulic fault detection in a helicopter's hydraulic system [72]. They tested and compared four approaches: Support Vector Machine (SVM) classifiers with different kernels and gradient boosting on oblivious decision trees. Simulated scenarios included three correct operational conditions and three types of failures: hydraulic leakage, discharged accumulator, and decreased volumetric efficiency. Results indicated that the gradient boosting approach achieved the highest accuracy (99.3%) but sacrificed interpretability.

In line with the theme of safety-critical applications, fault detection in steer-by-wire systems used in heavy earth-moving equipment was explored via a co-simulation setup [73]. Machine learning and deep learning classifiers, such as Bagged Trees and MLPs, were used to recognize dangerous steering faults from pressure signals. The ensemble method demonstrated

real-time responsiveness and offers a pathway to redundant, AI-based control validation.

Similarly, addressing aerospace applications, Kosova and Unver proposed a digital twin framework for detecting aircraft hydraulic system failures [74]. Using digital twins can help identify system failures during the aircraft's development cycle, aiding in the validation of health monitoring systems. They simulated 20 hydraulic system failure scenarios in AMESim, employing support vector machine and ensemble learning algorithms for fault detection. Results were consistent with Gareeve's approach, achieving high accuracy but at the expense of lower interpretability.

From a systems perspective, fault detection has also moved toward sensorless architectures [75]. In the case of Direct Driven Hydraulics (DDH), simulations revealed that fault classification accuracy is strongly influenced by inverter architecture. By analyzing only electric motor signals, the researchers achieved a classification performance of 85%, an outcome that emphasizes the interdependence between power electronics and hydraulic diagnostics.

When examining mobile hydraulic systems, such as mini excavators, a dedicated study focused on monitoring axial piston pumps [76]. This work introduced real-world operating cycles, including digging and operator-controlled scenarios, and demonstrated the impact of valve plate degradation on performance. Machine learning models, particularly KNNs, achieved high classification rates under controlled conditions; however, their performance decreased when models were tested across different operator behaviors.

Practical constraints in mobile equipment were also addressed through a sensor reduction strategy. Researchers employed feature selection to reduce the number of required sensors from 23 to just 5—maintaining high fault classification accuracy while significantly improving the economic and operational feasibility of CM systems [77].

Shen and Zhao introduced a one-dimensional multichannel convolutional neural network (1DMCCNN) for hydraulic fault detection in a landing gear hydraulic system [78]. The hydraulic landing system was simulated in AMESim, and five fault types were introduced alongside a fault-free setup. Pressure signals were processed as multichannel data and analyzed using 1DMCCNN, which improved fault detection precision compared to traditional ML algorithms. The authors noted the challenge of data imbalance between fault-free and defective states and suggested the need for more data from defective states to achieve better results.

In scenarios with limited sensor data, another study proposed the use of augmented simulation data to improve model training for axial piston pumps [79]. The trained classifiers achieved 70–85% accuracy, depending on the data source, demonstrating that synthetic data can help close performance gaps during early development phases.

Quality assurance during manufacturing is another critical point of condition monitoring. A recent approach combined vibroacoustic feature extraction with supervised ensemble classifiers to identify hard-to-detect faults in electric pumps during end-of-line testing [80]. Achieving over 95% accuracy, the method strengthens production reliability while operating under real-world testing constraints.

Targeting hydraulic cylinders, Zhang et al. proposed a particle filter-based method for quantitative fault diagnostics in hydraulic cylinders [81]. A detailed simulation model was developed to reproduce common fault types, including internal leakage, external leakage, and friction faults. The method relies on a model-based stochastic framework that simultaneously estimates system states and fault-related parameters using a joint state and parameter particle filter. Simulation results showed a mean absolute percentage error (MAPE) of less than 9% across all fault types, demonstrating high diagnostic accuracy. The authors emphasized the potential of particle filters for precise, quantitative fault diagnosis under varying operating conditions.

Historical relevance is highlighted in the contribution by Le et al. from 1997. Le et al. [82] show that even a quarter of a century ago, researchers recognized the potential of artificial neural networks for fault detection. The authors used a multilayer perceptron (3 layers: 6 input, 50 hidden, 31 output neurons) to detect and differentiate single and multiple point leakage in hydraulic systems with great success.

Building on advanced feature extraction, Huang et al. proposed a novel approach to analyze leakage faults in hydraulic systems by using permutation entropy fault detection. The features are extracted from decomposed vibration signals. For classification, deep belief networks (DBN) [83] are deployed, which use a deep architecture to learn hierarchical features from the input data automatically. The proposed method is validated on a hydraulic fault diagnosis dataset and was able to correctly differentiate between normal state, slight leakage, moderate leakage, and severe leakage.

A combined simulation and experimental data-driven approach was undertaken by Makansi and Schmitz. The authors used supervised learning and neural networks for condition monitoring of a hydraulic press. They investigated the potential of combining simulation and experimental data

to increase the prediction performance of the models. A lumped parameter simulation model of the real-world press was set up to efficiently generate large time-series datasets with varying operation conditions and fault levels [84, 85]. The implemented faults consisted of internal and external leakage at the cylinder, increased cylinder friction, position sensor offset, worn control edges of a proportional valve, and external leakage at the hydraulic power supply. The same faults were implemented in the real-world hydraulic press. The authors explained the feature extraction and feature selection processes in detail, focusing on their impact on fault prediction performance. The results show that feature selection has a significant effect on the final accuracy of the model. Additionally, the authors determined that for some faults, simple ML models are sufficient, while other faults require extended feature extraction, which in turn results in larger models (e.g., NNs) to achieve acceptable model performance. Lastly, the impact of the ratio between simulation and experimental data was investigated [86]. The findings of this study demonstrate that transferring knowledge from a simulated domain to a real domain can improve fault classification performance. The effect of the transfer, however, is strongly influenced by the mixing ratio of the training data. This is mainly due to inevitable differences between simulated and actual data instances. Adding simulation data can thus introduce noise and decrease the classification performance of the model.

Similarly, in [87] Hornik et al. use an artificial neural network to detect the two most common failures in hydraulic systems: actuator internal leakage and a blocked valve. A feedforward neural network is employed and trained with experimental data for all cases. Results demonstrate the capability of the ANN to isolate, detect, and identify various fault states, as well as unknown states, using only two pressure sensors and one velocity sensor.

Focusing on automated feature extraction, Helwig et al. implemented and assessed a systematic approach for automated feature extraction and supervised fault classification of hydraulic systems [88]. A hydraulic test rig, comprising a working, cooling, and filtration circuit, is utilized to simulate various faults, including overheating, valve wear, and leakage. Linear discriminant analysis was employed to mitigate the impact of one fault on another, thereby enhancing classification robustness. Through scatter-based supervised learning, the automatically extracted and selected features enabled the system to respond quickly to changing circumstances. These comprise modifications to the hydraulic design, sensor failures, or the appearance of a new fault situation.

Further exploring data efficiency, Gareeve et al. tested the capability of a gated convolutional autoencoder for fault detection in a hydraulic cylinder. Noteworthy is the researcher's challenge to use only 0.2% of the available experimental data for training (in addition to the simulation data). This demonstrates the time savings, as only a small amount of time is required at a real-world machine. Results show that this approach can lead to a condition state accuracy of up to 99%. Furthermore, with this fault classification model, it was possible to investigate its functionality by exploring the decision boundaries of the classifier in a two-dimensional embedding space [89].

For lifespan estimation, Gareev applied a feedforward artificial neural network-based technique to detect leakage in a hydraulic system and to estimate the remaining operational lifespan of a hydraulic cylinder as part of a servo-hydraulic actuator [90]. Internal leakage between the two chambers is implemented and controlled via an adjustable gap height in the simulation environment. Cylinder displacement and inlet pressure of the hydraulic booster are used to characterize the progress of internal leakage in the cylinder, which is then used to train the neural network. The difference between the simulation prediction of the error of residual service life (RUL) and true RULs from a physical cylinder indicates the capability of the neural network. When the rod displacement or the inlet pressure of the hydraulic booster is trained to train the model, the error between true and predicted RUL stays mostly below 20%. However, both approaches present lower prediction rates closer to the end of the lifespan due to limited data at this health stage.

In the study of Mallak et al., a combination of unsupervised learning (long short-term memory autoencoder) and supervised learning are used to perform two separate stages of condition monitoring for a hydraulic unit [91]. The autoencoder is used for fault detection. It is trained on data from a faultless machine and thus learns a certain threshold. All data exceeding this threshold are considered faults and are detected and classified by the implemented supervised deep learning method. To validate the capabilities of the presented method, it was tested on two different test scenarios, which varied in data pre-processing, acquisition, and structuring. The authors emphasize their work by aiming to detect not only component faults but also sensor faults.

Investigating hydraulic pump faults, Huiji et al. deployed stacked autoencoders (SAE) to identify four fault modes on a hydraulic pump based on vibration signals [92]. The faults consisted of a cylinder fault, valve plate abrasion, bearing fault, and piston clearance fault. The greedy layer-wise training method is chosen due to its approach of training one layer of the neural network at a time and fine-tuning the resulting network via

backpropagation. Due to the limited training data, a dropout strategy (deactivating nodes at random) is employed to prevent overfitting, and a rectified linear activation function (ReLU, a piecewise linear function that outputs the input directly if positive, otherwise zero) is utilized to enhance the accuracy of fault diagnosis. The results are compared with those of an SVM, a traditional SAE, and a conventional backpropagation algorithm. The SAE presents superior fault diagnostics performance among these four methods in the hydraulic pump case with limited training data.

Complementary to this, Tao et al. and Bei et al. applied wavelet packet decomposition and self-organizing map (SOM) neural network for fault detection from the vibration signal of a hydraulic pump [93, 94]. SOM is an unsupervised neuronal network that aids in reducing the dimensionality of the inputs. In the investigated case, the SOM is trained with data from a faultless system. Minimum Quantization Error (MQE) calculation is then used for fault detection by tracking trajectories in the features that indicate a degradation trend. The deviation computed by the model of new data is used to calculate the degradation level. The testing consisted of 14 samples of normal condition, four samples with a loose slipper, and four states with valve plate wear. Ten normal condition datasets are used for training, and the rest are used for validation. The results show that the presented method can identify the degradation of the pump by the MQE and thus label the current operation status. However, the presented method cannot identify the specific fault mode.

Taking this further, Tang et al. utilized continuous wavelet transform (CWT) to convert an axial piston pump's raw vibration time series signal into time-frequency images [95, 96, 97]. Multiple Convolutional Neural Networks (CNN) were deployed for the feature extraction and fault classification and compared against each other. The visualization of the reduced features was carried out using the t-distributed stochastic neighborhood embedding. The investigated faults included slipper failure, loose slipper, swash plate wear, and central spring wear (1200 images for each failure, Figure 10). With this approach, the authors achieved a high classification performance of up to 98%. The findings demonstrate that the proposed method of transforming vibration signals into images and then extracting features from these images can successfully enable intelligent fault detection of a piston pump in various operational states.

A novel model-based approach was introduced by Lu et al., who presented a chaotic parallel radial basis function (CPRBF) network to identify hydraulic pump faults [98]. 800 simulation datasets are divided into 600 for

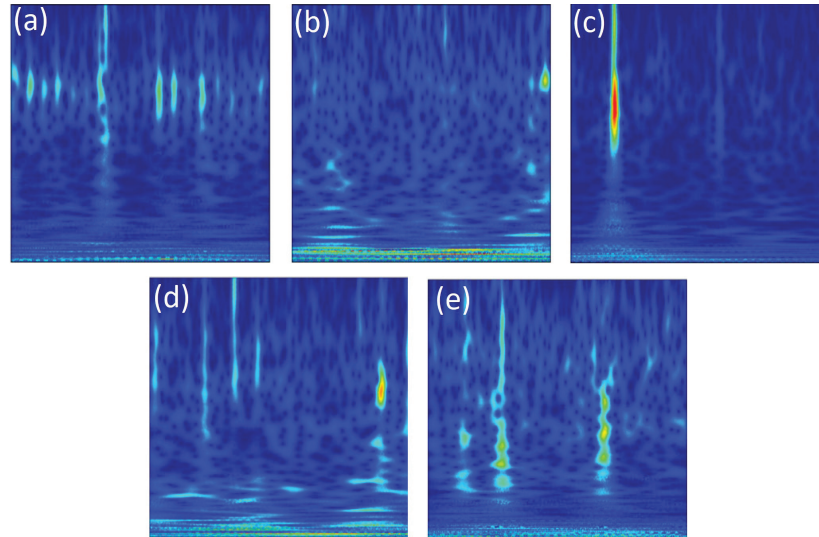


Figure 10 Time-frequency distribution obtained by image conversion of the vibration signals of the axial piston pump failures: faultless (a), slipper failure (b), loose slipper (c), swashplate wear (d), and central spring wear (e) [95].

training, 100 for weight factor optimization, and 100 for testing. CPRBF exhibits superior one-step iterative prediction performance in terms of stability and convergence, as demonstrated in direct comparison with normal radial basis functions. Test bench data is collected and divided for validation, like the simulation data. Again, the CPRBF presented higher prediction accuracy. CFRBF demonstrated good capabilities in comparing normal and faulty states to identify threshold-exceeding residual errors, enabling effective fault detection. Yet, at the time of the study, identifying the exact fault origin is impossible, as different faults (valve plate/rotor, swash plate/slipper) can present similar signal patterns.

Focusing on real-world complexity, Rajakarunakaran et al. used an artificial neural network to detect 20 different faults of a centrifugal hydraulic pumping system [99]. Training and testing data were generated on a real-world test bench. The faults include exceeding pressure, temperature, and speed limits, component wear, failures, improper assembly, and leaks, to name a few. PCA was used to identify fault-representative features, which decreased the number of input features to the ANN to 11. Six hundred data sets were conducted, of which 450 were separated for training and 150 for testing. Before applying PCA, 149 datasets were classified correctly, resulting

in an accuracy of 99.3%. After PCA, only eight inputs remained; the training time accelerated from 81 to 7 seconds, and the accuracy improved to 100%.

Further fault diagnosis studies were conducted by Karpenko et al., who investigated the performance of a multilayer neural network in diagnosing actuator faults in a pneumatic control valve [100, 101]. Three common valve faults were experimentally investigated under various operating conditions: incorrect supply pressure, diaphragm leakage, and actuator vent blockage. For each condition and fault, the same step command was given. Experimental data from faulty states, as well as a faultless state, were collected, including 63% and 86% rise times, dead time, peak time, steady-state error, percent overshoot, and gain. These data were used to train the neural network, creating a discriminatory fault signature. The presented results display a high fault detection performance of the neural network. Additionally, it was capable of identifying different levels of failure and assessing untrained fault levels.

Finally, Duensing et al. developed a general workflow for detecting faults in the cardan shaft of a dump loader by analyzing the pressure signals of the hydraulic drive train [102]. The drive train is connected to the cardan shaft. Experimental data from a test bench are used. Three cardan shaft states are examined: functional, bearing failure, and misalignment. Preprocessing steps to enhance the detection rate are explained, including the segmentation of individual operation states and the removal of outliers. Features for each segment are extracted and evaluated for selection to reduce the overall computing time, as only features with information relevant to the machine health status are selected. Different machine learning algorithms, such as SVM, neural networks, k-Nearest Neighbor, and Support Vector Machines, are trained and validated to determine the best algorithm for the use case. Group cross-validation (5-fold) is used to detect overfitting. By optimizing the model parameters, further improvement in classification is achieved. The results show that the faults are classified with an accuracy of 74% (84% when both faults are combined). The authors further investigate the scope of the data used for training and validation.

Figure 11 displays the classification performance of the neural network over the percentage of all data used. Since convergence has not yet been achieved, it can be concluded that classification performance can be improved with additional data.

The reviewed literature on machine learning applications in condition monitoring and fault diagnosis is summarized in two separate tables. Table 5 presents selected real-world use cases, illustrating how ML methods have

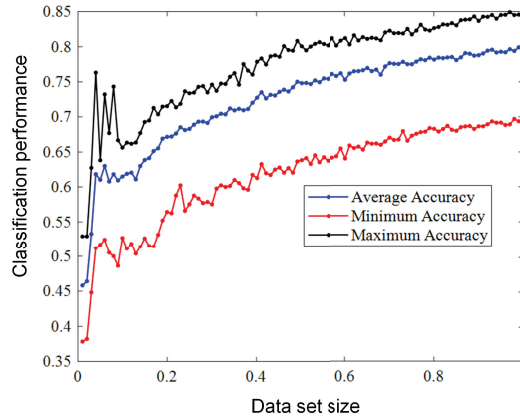


Figure 11 Classification performance depending on the data set percentage [102].

been applied in practical settings for monitoring and diagnosing faults in technical systems. In contrast, Table 3 highlights simulation-based approaches, where ML techniques were evaluated under controlled conditions. Both tables provide a structured overview of the respective use cases, applied methods, and key results, offering the reader a concise comparison of practical and experimental applications of machine learning in this domain.

4.3 Data Reduction and Imbalance Handling

As evident in the multiple studies mentioned above, a primary issue in training models based on real-world data is the scarcity of data for each failure mode. Joo et al. addressed this problem in [103] by comparing different methods for solving class imbalance for hydraulic systems. According to the authors, the methods can be divided into sampling methods and ensemble methods.

The sampling method can be divided into over- and undersampling methods. As the name suggests, when using oversampling techniques, data of the underrepresented class is duplicated. This method is easy yet tends to result in unwanted overfitted models. A more advanced algorithm presents the synthetic minority oversampling technique (SMOTE, Figure 12 (left)), which generates new synthetic data points using k-Nearest Neighbors (kNN) [104]. Borderline SMOTE is a modified version of the original one but focuses the newly generated points closer together to avoid the possible creation of new data far away from the decision boundary (Figure 12 (right)) [105]. Lastly,

Table 3 Simulative application of ML in condition monitoring

Author	Year	Use Case	Method	Result
Theljani et al.	2011	Fault diagnosis in a simulation of a three-tank system	Semi-supervised classification algorithm that utilizes labeled and unlabeled data	Successful clustering and label generation, but some faults were misclassified
Vigolo et al.	2023	Online monitoring of pneumatic actuation systems	Hybrid ML model (2 regression + 1 classification), operating point method	Improved energy efficiency and robustness in 50 simulated conditions
Ali's	2016	Identifying fault symptoms in internal gear pumps	Kohonen neural network – unsupervised learning approach	ML-pipeline was validated and was able to detect anomalies in the data
Castilho et al.	2018	Identify failure modes in a pneumatic pressure system	Two supervised learning algorithms (support vector machine and decision tree)	Results favored support vector machine for higher accuracy
Sano et al.	2019	Continued fault detection in pneumatic pressure system	Supervised learning (k-nearest neighbors, Naive Bayes, SVM)	Single-fault cases, all algorithms achieved mean accuracy above 90%
Gareeve et al.	2019	Hydraulic fault detection in an aircraft system	Test and comparison of four approaches: Support Vector Machine, Decision Tree, Random Forest, Gradient Boosting	Gradient boosting approach achieved the highest accuracy
Kosova and Unve	2022	Digital twin framework for fault detection in hydraulic systems	Digital twin framework – Support Vector Machine, Neural Networks, Random Forest	Similar to Gareeve – high accuracy but lower interpretability
Shen and Zhao	2022	One-dimensional multichannel convolutional neural network for fault detection in hydraulic systems	One-dimensional multichannel convolutional neural network (1D MCNN)	Improved fault detection precision compared to traditional methods

(Continued)

Table 3 Continued

Author	Year	Use Case	Method	Result
Azeez et al.	2022	CM of axial piston pumps using augmented sim data	ML classifier trained on real, simulated, and augmented data	Accuracy: 84.8% (real), 70.1% (sim), 75.7% (augmented)
Zhang et al.	2023	Simulative fault diagnosis for hydraulic cylinders, targeting friction, internal leakage, and external leakage	Model-based stochastic framework employing joint state and parameter estimation via a particle filter	Mean Absolute Percentage Error (MAPE) below 9%, i.e., over 91% accuracy across all three fault types in simulation experiments
Zakharov et al.	2023	Sensorless fault classification in DDH systems	AI-based analysis of electric drive signals, inverter variation	Accuracy 72–87%; strongly influenced by inverter type
Singh et al.	2024	Redundancy in steer-by-wire hydraulic systems	Co-simulation with ensemble classifiers (MLP, Bagged Tree, Naive Bayes)	Real-time fault detection for fail-safe steering control
Vigolo et al.	2024	Online optimization of pneumatic drives	Hybrid ML with model-based optimization, real-time pressure assessment	Reduced air consumption; robust under load variation

adaptive system sampling (ADASYN, Figure 12 (bottom)) again improves on the SMOTE algorithm by adjusting the synthesis coefficients [106].

Undersampling consists of ignoring data of the over-represented class in its simplest form, but can decrease classification performance as crucial data points are omitted. The edited-nearest neighbor (ENN) algorithm is more sophisticated as it only removes data instances of the over-represented class, which are close together, thereby reducing the possibility of removing important data points. The Tomek Link algorithm is similar to the ENN, yet it focuses and removes the data instances close to the underrepresented class, resulting in greater separation between the two classes [107].

The ensemble methods are separated into bagging-based methods and boosting-based methods. Examples of bagging-based methods are bagging-SVM (bootstrap aggregating) [108], random forest [109], and extremely random trees [110]. Bagging is a method that enhances the efficiency and

Table 4 Experimental application of ML for condition monitoring

Author	Year	Use Case	Method	Result
Le et al.	1997	Fault detection in hydraulic systems	Multilayer perception Neuronal Network	Successful detection and differentiation of single and multiple point leakage
Borriello et al.	2023	Detection of manufacturing errors in external gear pumps	Ensemble ML, simulation (Multics-HYGESim) + experimental validation	~92% accuracy in detecting defects like conicity and axial runout
Huang et al.	2022	Analyzing leakage faults in hydraulic systems using permutation entropy fault detection	Deep belief networks (DBN) – Feature extraction from decomposed vibration signals	Correctly differentiates between normal state, slight leakage, moderate leakage, and severe leakage
Makansi & Schmitz	2022	Fault classification in hydraulic press	SL – Combination of simulation and experimental data	Improved fault classification performance, deep dive in the impact of training data mixing ratio on model accuracy
Hornik et al.	2022	Detection of common hydraulic system failures	Feedforward neural network – Trained with experimental data	Isolates, detects, and identifies different fault states and unknown states using two pressure sensors and one velocity sensor
Helwig et al.	2022	Automated feature extraction and supervised fault classification of hydraulic systems	Linear discriminant analysis, scatter-based supervised learning	Increased classification robustness, quick response to changing circumstances, effective for various faults
Gareeve et al.	2022	Fault detection in a hydraulic cylinder	Gated convolutional autoencoder – Minimal experimental data, simulation data	Achieved up to 99% condition state accuracy, explored decision boundaries in two-dimensional embedding space

(Continued)

Table 4 Continued

Author	Year	Use Case	Method	Result
Mallak et al.	2022	Condition monitoring of a hydraulic unit	Long short-term memory autoencoder (LSTM) for fault detection, supervised deep learning (SDL) for classification	Effective fault detection and classification, validated on different test scenarios, detects component and sensor faults
Huiji et al.	2022	Identification of fault modes on a hydraulic pump	Stacked autoencoders (SAE), dropout strategy, ReLU activation function	Superior fault diagnostics performance compared to SVM, trad. SAE and backpropagation, effective with limited training data
Tao et al., Bei et al.	2022	Fault detection from the vibration signal of a hydraulic pump	Wavelet packet decomposition, self-organizing map (SOM), Min. Quantization Error (MQE) calculation	Capable of identifying pump degradation, labels current operation status, not able to identify specific fault mode
Tang et al.	2022	Intelligent fault detection of an axial piston pump using continuous wavelet transform (CWT) and CNNs	Continuous wavelet transform (CWT), Convolutional Neural Networks (CNNs), t-SNE visualization	Achieved up to 98% classification performance, effective feature extraction and fault classification from vibration signal images
Lu et al.	2022	Hydraulic pump fault identification using chaotic parallel radial basis function (CPRBF) network	Chaotic parallel radial basis function (CPRBF) network – Simulation and test bench data	Better prediction performance compared to normal radial basis functions, high prediction accuracy, capable of comparing normal and faulty states but cannot identify exact fault origin

(Continued)

Table 4 Continued

Author	Year	Use Case	Method	Result
Rajakaruna-karan et al.	2022	Fault detection in a centrifugal hydraulic pumping system	Artificial neural network (ANN), Principal Component Analysis (PCA) – Real-world test bench data	PCA improved training speed and accuracy, effective fault detection of pressure, temperature, speed limits, wear, failures, wrongful assembly, and leakages
Karpenko et al.	2022	Diagnosis of actuator faults in a pneumatic control valve	Multilayer neural network – Experimental data on common valve faults	High fault detection performance, identification of different fault levels, assessment of untrained fault levels

Table 5 Experimental application of ML for condition monitoring

Author	Year	Use Case	Method	Result
Duensing et al.	2022	Fault detection in the cardan shaft of a dumb loader by analyzing pressure signals	SVM, neural networks, k-nearest neighbor – Experimental data on functional, bearing failure, and misalignment states	74% classification accuracy (84% for combined faults), preprocessing steps and feature selection to improve detection rate
Keller et al.	2022	CM of piston pumps with minimal sensors	ML + feature selection on stationary test rig	100% accuracy with only 5 sensors instead of 23
Keller et al.	2023	Monitoring axial piston pumps on mini excavators	Fine KNN classifier on sensor data during different duty cycles	100% accuracy in controlled cycles; 60% in cross-cycle validation
Keller et al.	2023	EOL test of electric pumps via vibroacoustic analysis	Ensemble ML using pressure & acceleration signals	95% accuracy in identifying subtle component defects

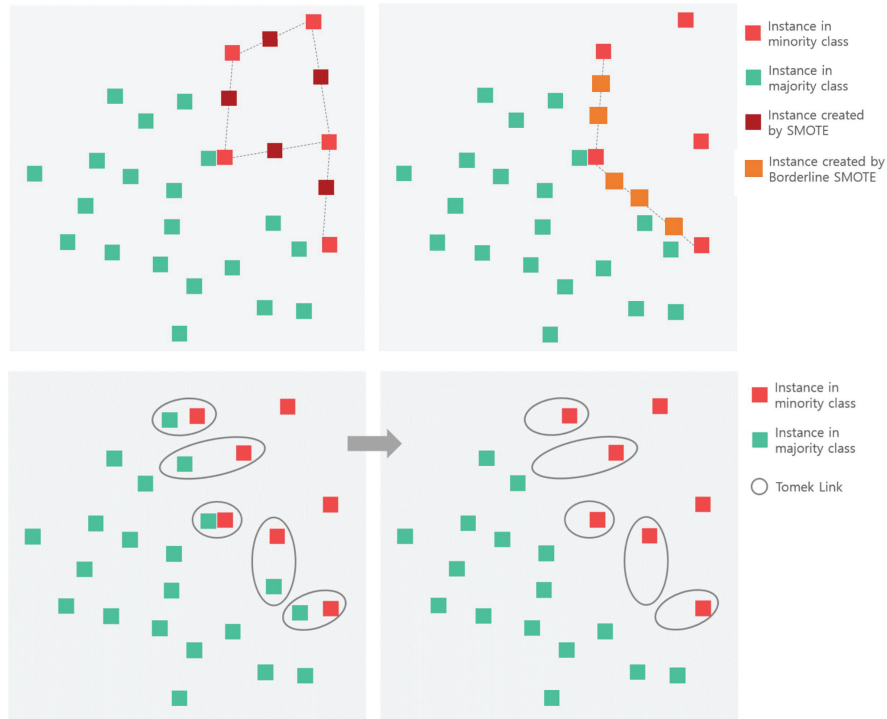


Figure 12 SMOTE (top left), Borderline SMOTE (top right), Tomak Line (bottom)

resilience of machine learning models by training multiple models on different portions of the data and aggregating their predictions. Random forests and extremely randomized trees work similarly, combining the results of multiple decision trees. They differ in their approach to constructing decision trees. A Random Forest selects the best split among a random subset of features, while an extremely randomized tree selects the split point randomly, decreasing the computation time. The boosting methods can be further divided into adaptive boosting (AdaBoosting) and gradient boosting methods. AdaBoosting iteratively trains weak models and assigns higher weights to misclassified data instances after each training iteration, thereby boosting the overall model's performance. The final model is a weighted combination of the weak classifiers, where each classifier is assigned a weight based on its accuracy [111]. Gradient Boosting fits new models to the residual errors made by previous models, with each subsequent model being trained to compensate for the residual error of the previous model [112].

Joo et al. continued to test all presented methods on a data set of a hydraulic system consisting of sensor signals at different working states. Results show that the effectiveness of nearly all methods depends on the imbalance ratio. When the ratio was low, random forest, extremely randomized trees, and AdaBoosting performed best. In cases of increasing imbalance ratios, random undersampling helped machine learning models achieve stable performance.

To improve identification accuracy in modeling electrohydraulic systems, recent work explored adaptive estimation methods for flow mapping in valves [113]. Algorithms such as RLSM, RMLM, and radial basis function neural networks were evaluated for their ability to efficiently capture pressure-flow relationships. Simulation results demonstrated that introducing prior knowledge structures improved both speed and accuracy, offering new tools to manage uncertainty in fluid system models.

Other challenges can still emerge if a large data set is available, sufficient for training effective ML models. Handling and processing large amounts of data quickly is only possible with computationally powerful but expensive equipment. Brumand-Poor et al. showed in [114] the possibilities of using a deep unsupervised learning model for the reduction of high-dimensional raw data without information loss. A VAE was implemented, which contained LSTM layers to allow time-series data processing. The autoencoder was trained with batch sizes of 9 time series in one training iteration, where each time series was compressed to a single value, resulting in 9 latent features (Figure 13). Data from a simulation model of a hydraulic press was used. A cycle was divided into four segments: extraction of the cylinder with low pressure and high speeds, slow approximation in workpiece distance with controlled speeds, increasing pressure during the press procedure, and retraction of the cylinder after the complete press procedure with no speed limits.

One complete press cycle has a duration of 18 seconds, the sensors signals are sampled with 100 Hz, resulting in 1801 measurement points. After passing the raw time-series data through the encoder, the data size was reduced from a 1801×9 matrix to a latent size of 9. This equals a compression rate of 99.94%, with a limited deviation of less than 4% when comparing the original time series to the reconstructed one from the decoder.

Complementing the effort to deal with high data diversity, a recent study introduced an on-site retrainable CRDNN architecture for mobile construction equipment [115]. Originally trained offline, the network adapts to new machine types via transfer learning directly on the onboard ECU.

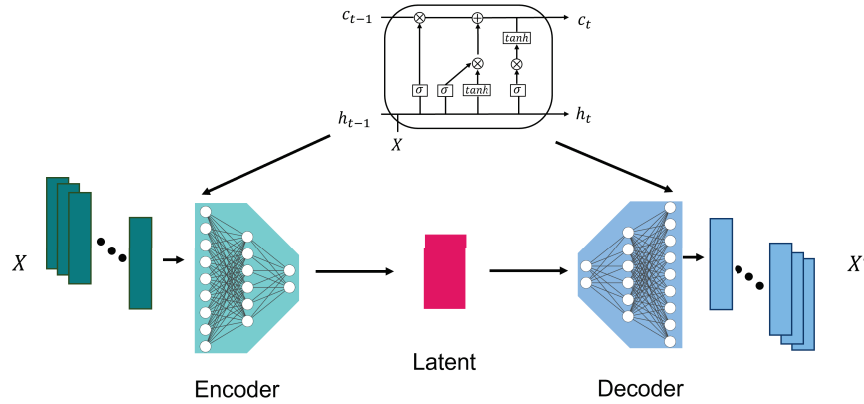


Figure 13 Structure of the Variational Autoencoder

Through IoT connectivity and operator-driven data labeling, the system maintains its accuracy even across variations in machines. In field tests, this approach improved torque cycle detection efficiency by 9% and enabled AI model updates without specialist knowledge, underscoring its usability for real-world mobile applications.

Table 6 summarizes the above-mentioned different real-world approaches of ML in the field of data imbalance/reduction.

4.4 ML for Functional Modeling and Prediction

At the core of most ML tasks is constructing mathematical structures representing the input-output relationship present in a given dataset. These black box models exhibit one main benefit: the adaptive nature of their structure and the ability to improve their function approximation on provided data. In the following subchapter, the use of ML algorithms to enhance fluid power system modeling is presented.

The study of Su et al. presents an approach for developing data-based flow rate prediction models for independent metering hydraulic valves using machine learning algorithms [116]. The study used labeled experimental data obtained from a hydraulic system with an independent metering valve to train and validate the models. Inlet and outlet pressure, as well as temperatures, valve opening, and flow rate, were used as inputs. For ML algorithms, Linear regression (LR), k-nearest neighbor (kNN), decision tree (DT), random forest (RF), support vector regression (SVR), and artificial and deep neural network (ANN, DNN) are selected. Hyperparameter optimization was handled via

Table 6 Experimental application – data imbalance & reduction

Author	Year	Use Case	Method	Result
Liu et al.	2021	Adaptive flow mapping of electrohydraulic valves	RLSM, RMLM, RBF neural network; prior knowledge integration	Accurate parameter estimation in simulation; improved identification speed and model reliability
Xiang et al., Han et al., He et al., I. Tomek,	2021	On-site retraining for mobile machines (CRDNN)	CRDNN with transfer learning and IoT-based data labeling	+9% torque cycle detection efficiency; robust across machine types; AI retraining possible without expert knowledge
Chawla et al., Han et al., He et al., I. Tomek,	2022	Sampling methods for handling class imbalance in datasets	Oversampling, undersampling, SMOTE, Borderline SMOTE, ADASYN, ENN, Tomek Link	SMOTE uses kNN to create synthetic data points, Borderline SMOTE generates points closer to decision boundary, ADASYN adjusts synthesis coefficients, ENN removes over-represented class data, Tomek Link focuses on removing instances close to the underrepresented class
L. Breiman, Geurts et al., Freund & Schapire, J.H. Friedman	2022	Ensemble methods for machine learning	Bagging (Bagging-SVM, Random Forest, Extremely Random Trees), Boosting (AdaBoosting, Gradient Boosting)	Bagging enhances model efficiency by aggregating multiple models' predictions, Random Forest selects best splits, Extremely Random Trees select split points randomly, AdaBoosting assigns higher weights to misclassified instances, Gradient Boosting fits models to residual errors

(Continued)

Table 6 Continued

Author	Year	Use Case	Method	Result
Joo et al.	2022	Evaluation of machine learning methods on hydraulic system data	Random Forest, Extremely Randomized Trees, AdaBoosting, Random Undersampling	Effectiveness depends on imbalance ratio, Random Forest, Extremely Randomized Trees, and AdaBoosting perform best with low imbalance, and Random Undersampling stabilizes performance with high imbalance
Brumand-Poor et al.	2022	Deep unsupervised learning model for data reduction in the hydraulic press	Variational Autoencoder (VAE) with LSTM layers	Compressed high-dimensional raw data with 99.94% compression rate and below 4% deviation, effective for large time-series datasets

grid and random search. Results indicate that SVR and DNN are the best-performing ML algorithms, with mean absolute percentage errors of 3.9% and 4.5%, respectively.

In a similar study, researchers have addressed the challenge of acquiring position information in harsh environments by proposing a soft-sensing model based on back propagation neural networks [117]. This model estimates the velocity of a direct-driven hydraulic cylinder using multi-input signals such as electric motor speed and chamber pressures, which is then integrated to infer position. When tested under various simulated working conditions and validated experimentally on a standalone crane model, the position estimation error remained within $\pm 2.5\%$, highlighting the method's robustness for sensorless applications in difficult environments.

Baronti et al. compare in [118] three approaches to model the dynamics of an electro-hydraulic load simulator: an analytical linear model with four intelligent parameter optimizers, an artificial neural network model (ANN), and a hybrid analytical and ANN model. Parameter optimization is done using data from an experimental test bench. All four optimizers show similar improvement to the analytical model. Yet, while the optimized analytical

model represents state-of-the-art load simulator modeling, the two alternative models involving ANNs present a higher accuracy. The authors suggest that this is due to the unmodelled nonlinearities of the analytical model, which are not present in the hybrid and ANN approaches.

A different approach to workload understanding in hydraulic systems was taken through the use of LSTM networks to recognize time-varying loads during excavator digging [119]. By analyzing pressure, position, and velocity signals from both chambers of a hydraulic actuator, the trained LSTM model successfully identified distinct workload types. This classification enables real-time workload perception, contributing significantly to automation and safety in mobile hydraulic machinery.

In [120] Achieng assesses the performance and efficiency properties of machine learning-based simulated pumps by comparing the simulation results with measurements of a 50-year-old laboratory pump. In the case of the real-world pump, power, head, and discharge are recorded in four separate test runs. Three support vector regression techniques are applied for the machine learning methods: radial basis function, linear, and quadratic kernel. The model is trained to simulate efficiency, power, and head in relation to discharge. The authors explain in detail the impact of different relevant hyperparameters, such as loss and kernel parameters, on the regression plausibility. This is crucial, as incorrect identification of these parameters during training can result in an under- or overfitted model. The investigation results reveal that machine learning techniques can create simulation models for real-world pumps and that using the radial basis function with support vector regression produces the best regression model for simulating the pump's power, efficiency, and head parameters.

Going beyond internal machine modeling, assistance systems for forestry applications have been explored that combine object recognition with data-driven interpretation of hydraulic signals. In particular, a CNN-based crane scale system is used to assess log quality by comparing estimated log weight to expected values for healthy wood. Deviations—e.g., due to beetle infestation—trigger visual operator warnings, thus easing mental workload while improving process reliability [121].

A good example of how to apply machine learning to reduce simulation time is presented in the work of Hosseiny et al. The researchers combine the modeling and 2D hydraulic simulation of a river stream with a random forest (RF) classifier and a multilayer perceptron (MLP) to identify flooded areas and flood depth [122]. This approach aims to tackle the problem of simulating complex systems, such as a river stream, which is currently not feasible in

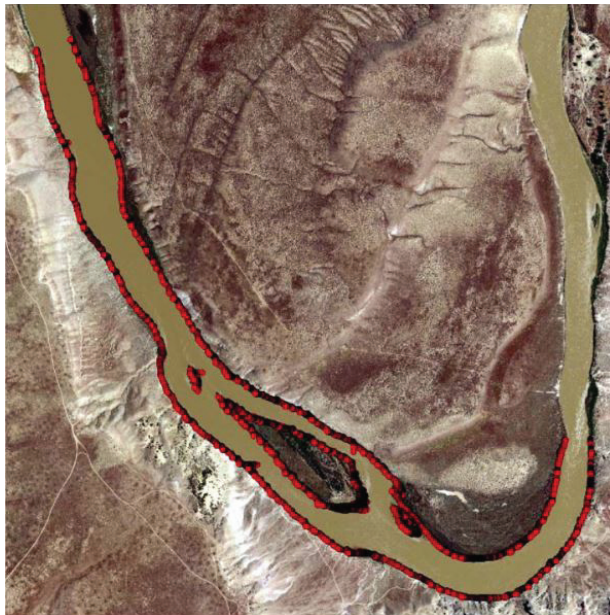


Figure 14 Classification errors of wet and dry nodes [122].

modern simulation environments. The more straightforward hydraulic 2D simulation is calibrated using water surface elevation measurements obtained during low to high flow magnitudes, quantified by the flow exceedance condition and the discharge amount (20–350 m³/s). The simulation results are used to train the two machine-learning models. The random forest model task is to detect wet and dry nodes over the domain. Here, 50 trees presented the sweet spot between computation time and accuracy, as increasing trees does not further improve the results. The red marks in Figure 14 reflect the misclassified nodes of the RF. Overall, this ML method achieves a classification rate of 98%, with a computing time of 1.9 seconds for all 1.1 million nodes. The multilayer perceptron is trained to estimate the depth of each specific node. The structure consists of four hidden layers with 150, 100, 50, and 30 neurons. The overall data consists of 329 discharges, and the number of epochs is 80,000.

To assess the quality of the model, unseen discharge amounts are simulated and computed via the MLP. The differences are displayed in Figure 15. Additionally, the computing time of 5 seconds compared to 5 minutes of simulation presents a reduction 98.3%.

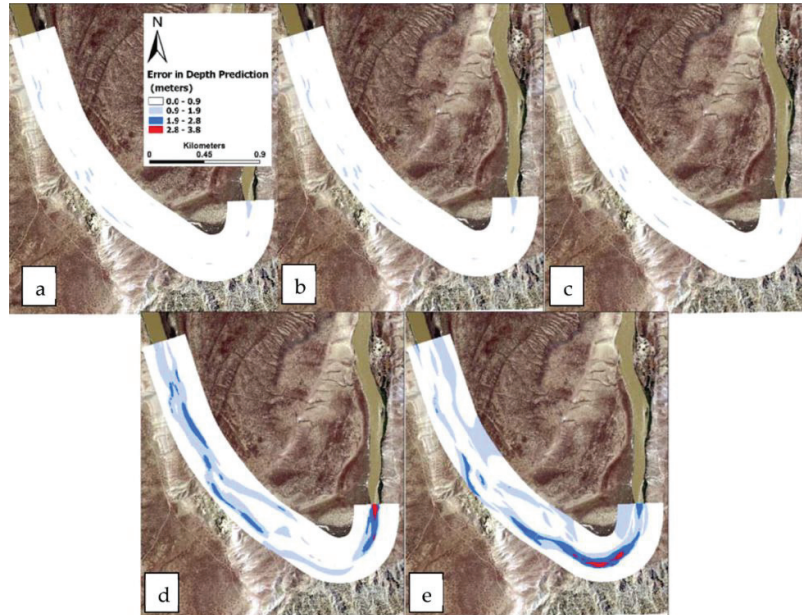


Figure 15 Comparison of simulation and MLP results of different discharges: (a) 20, (b) 30, (c) 45, (d) 225, and (e) 350 m³/s [122].

Extending automation at the excavation site, a deep learning model based on Faster R-CNN and depth maps was developed to estimate the volume of material in excavator buckets [123]. Achieving 96.91% accuracy and real-time performance at 10 fps, this method facilitates monitoring system performance and fill efficiency in semi-autonomous or fully automated excavation workflows.

Finally, functional modeling also proves useful in fuel compatibility prediction [124, 125]. A unified framework using ML regression models enables early estimation of interactions between bio-hybrid fuels and sealing materials, such as elastomer degradation or volume changes, based on fluid properties. By pre-selecting promising fuel/seal combinations before costly immersion testing, this approach addresses both material safety and development efficiency. The dual studies reinforce the practicality of combining experimental data with algorithmic predictions to assess the behavior of bio-based fuels in hydraulic systems.

Table 7 summarizes all the different real-world approaches of ML for Functional Modeling and Prediction.

Table 7 Experimental application – ML for functional modeling and prediction

Author	Year	Use Case	Method	Result
Zhang et al.	2021	Position soft-sensing for DDH in harsh environments	Back Propagation Neural Network trained on simulation data	Position estimation error $\pm 2.5\%$; robust under varied working conditions
Su et al.	2022	Data-based flow rate prediction models for independent metering hydraulic valves	Linear regression (LR), k-nearest neighbor (kNN), decision tree (DT), random forest (RF), support vector regression (SVR), artificial and deep neural networks (ANN, DNN)	SVR and DNN show best performance with mean absolute percentage errors of 3.9% and 4.5% respectively, hyperparameter optimization via grid and random search
Baronti et al.	2022	Comparison of modeling approaches for electro-hydraulic load simulator dynamics	Analytical linear model with parameter optimization, Artificial Neural Network (ANN), hybrid analytical and ANN model	Optimized analytical model shows state-of-the-art performance, ANN and hybrid models present higher accuracy due to handling unmodeled nonlinearities
A. M. Gareev	2022	Leakage detection and RUL estimation in a hydraulic cylinder	Feedforward artificial neural network – Cylinder displacement and inlet pressure data	Prediction error mostly below 20%, lower prediction accuracy closer to end of lifespan due to limited data
K. O. Achieng	2022	Performance and efficiency assessment of ML-based simulated pumps compared to a real-world pump	Support vector regression (radial basis function, linear, quadratic kernel) – Simulation vs. measurements	Radial basis function SVR produces the best regression model, effective simulation of pump's power, efficiency, and head parameters
Hosseiny et al.	2022	ML to reduce simulation time for river stream hydraulic simulation	Random forest (RF) classifier, multilayer perceptron (MLP) – 2D hydraulic simulation	RF achieves 98% classification rate with 50 trees, MLP estimates flood depth with significant reduction in computing time (98.3% reduction)

(Continued)

Table 7 Continued

Author	Year	Use Case	Method	Result
Helian et al.	2024	Workload classification in excavator digging processes	LSTM-based deep-learning classifier using actuator signal data	High accuracy in recognizing time-varying workloads in real-time
Chris et al.	2020	Log quality and assortment estimation in forestry	CNN-based crane scale + object detection for weight deviation	Detected anomalies (e.g., beetle damage); assisted operator decision-making
Helian et al.	2024	Excavator bucket fill estimation	Faster R-CNN on depth maps + hybrid loss function	96.91% accuracy at 10 fps; suitable for real-time automation feedback
Boden et al.	2020	Material compatibility prediction with bio-hybrid fuels	ML regression using experimental data & fluid properties	Preselection of fuel/seal combos; avoided material failure in early screening

5 Comparative Insights and Method Selection

Chapter 4 presented a rich and increasingly diverse landscape of machine learning (ML) approaches in fluid power systems, encompassing classical supervised learning, unsupervised and reinforcement learning (RL), as well as emerging paradigms like self-supervised, graph-based, hybrid, and uncertainty-aware models. To better understand their respective trade-offs, this chapter compares these approaches along four key dimensions: data requirements, interpretability, computational cost, and deployment complexity.

5.1 Cross-Method Comparison

Chapter 4 highlighted a broad spectrum of ML techniques applied to fluid power, ranging from supervised learning and unsupervised approaches to reinforcement learning (RL). To better understand their respective advantages and limitations, it is instructive to compare these paradigms along key dimensions such as data requirements, interpretability, computational cost, and deployment complexity.

Starting with data requirements, supervised learning remains central in applications with sufficient labeled data. Techniques like SVMs, random forests, and deep neural networks have been applied in fault detection, condition monitoring, and control. However, data labeling is often expensive or infeasible in real-world systems. To mitigate this, Chapter 4 highlighted the growing relevance of self-supervised learning methods, which generate labels from structure in the input data itself, e.g., via contrastive learning or pretext tasks. These approaches help reduce dependency on manually labeled datasets. Similarly, transfer learning and few-shot learning allow pretraining models on simulation or related domains before fine-tuning them with limited task-specific data.

Unsupervised methods, such as clustering and autoencoders, require no labels but need comprehensive examples of normal behavior. Techniques such as variational autoencoders (VAEs) and topological data analysis (TDA) are being used for system health monitoring. Reinforcement learning still requires extensive data, often millions of simulation steps or training episodes. To address this, hybrid pipelines now use supervised pretraining, domain randomization, or curriculum learning to improve sample efficiency.

When considering interpretability, model transparency is a key concern in safety-critical applications. Interpretable models, such as decision trees, random forests, and SVMs, remain favored in regulated environments. However, many modern ML systems are inherently opaque. Deep RL policies and graph-based models, such as Graph Neural Networks (GNNs), offer high expressiveness but are difficult to inspect. To address this, hybrid models that embed physical priors or equations have gained traction, improving both interpretability and generalization. Techniques such as symbolic regression, physics-informed neural networks, and sparse regression are examples of these methods. Additionally, uncertainty quantification methods (e.g., Bayesian deep learning, ensemble variance) offer ways to assess confidence in predictions.

In terms of computational cost, complex models (e.g., multi-layer LSTMs, multi-agent RL, or GNNs) often require significant computing resources and training time. This is particularly true for simulation-based RL or when modeling temporal and spatial dynamics. However, emerging lightweight alternatives, such as compressed neural networks, surrogate models, and online learning frameworks, offer reduced inference costs. These are suited for deployment on embedded hardware. Neural surrogates trained on high-fidelity simulations can also be used to accelerate optimization or control design tasks.

Finally, deployment complexity varies widely among these methods. Traditional supervised pipelines are relatively easy to integrate once trained, especially with structured input features. However, continual learning and online adaptation methods are now necessary in changing environments. These allow models to adapt incrementally to system drift without retraining from scratch. RL agents, while powerful, require real-time safety mechanisms and extensive validation, especially when interacting directly with hardware. Hybrid strategies, which combine physics-based modeling for stability and ML for residual learning, help address deployment concerns in safety-critical environments. Meanwhile, uncertainty-aware models are increasingly utilized in supervisory roles, where prediction confidence is essential for implementing fallback logic and isolating faults.

The evolving range of ML methods shows that no single paradigm is ideal for all fluid power tasks. Instead, a successful application depends on aligning the method with specific project constraints, data availability, and operational requirements. The recent shift toward hybrid, adaptive, and self-supervised systems reflects an increasing need for robust, efficient, and transparent ML deployments. This comparative understanding lays the groundwork for the method selection roadmap presented in the next section.

5.2 Practical Guidelines for Engineers

Roadmap for Selecting Machine Learning Methods in Fluid Power Applications

The increasing complexity and diversity of ML applications in fluid power necessitate a structured, task-oriented approach to method selection. This roadmap serves as a practical guide for choosing machine learning methods based on specific technical conditions, available data, and deployment requirements.

1. Define the Task Type

- *Fault Classification or Regression?*
→ Use *Supervised Learning* methods such as Support Vector Machines (SVM), Random Forests (RF), or Deep Neural Networks (DNNs).

Example: Zhang et al. [36] used supervised learning to map sensor data to fault types in hydraulic loading systems. Similarly, Guo et al. [55] used regression-based adaptive tracking controllers with strong robustness.

- *Anomaly Detection without Labeled Fault Data?*
 → Use Unsupervised Learning methods such as Clustering, Autoencoders, or Self-Organizing Maps (SOM).
Example: Mallak et al. [91] combined autoencoders with supervised methods for dual-stage condition monitoring. Autoencoders were also used to detect faults in valve systems and identify abnormal system behavior without prior labels.
- *Adaptive Control or Policy Optimization?*
 → Use Reinforcement Learning methods such as Proximal Policy Optimization (PPO), Deep Deterministic Policy Gradient (DDPG), or Actor-Critic architectures.
Example: Andersson et al. [33] and Dhakate et al. [46] used PPO-based agents to control forestry cranes and hydraulic manipulators, outperforming PID controllers in dynamic environments.

2. Assess Data Availability and Quality

- *Limited Labeled Data:*
 → Prefer ensemble models like Random Forest or Gradient Boosting trained on engineered features.
 → Consider semi-supervised or data augmentation strategies.
Example: Theljani et al. [65] used semi-supervised learning for classifying fault modes in tank systems. In fault scenarios with rare data, synthetic oversampling such as SMOTE proved beneficial [103].
- *High-dimensional Time Series Data:*
 → Apply dimensionality reduction techniques like Principal Component Analysis (PCA) or Variational Autoencoders (VAE).
Example: Brumand-Poor et al. [114] achieved a 99.94% data compression using an LSTM-VAE model trained on hydraulic press cycles.
- *Imbalanced Datasets:*
 → Use oversampling methods like SMOTE, Borderline-SMOTE, or ADASYN.
 → Use undersampling like Edited Nearest Neighbor (ENN) or Tomek Links.
 → Evaluate ensemble classifiers (e.g., Random Forest, AdaBoost) for robustness.
Example: Joo et al. [103] showed that SMOTE + ensemble methods effectively mitigated performance loss on imbalanced hydraulic datasets.

3. Evaluate Computational and Deployment Constraints

- *Real-Time Inference (Embedded Hardware):*
 - Choose lightweight models, e.g., Support Vector Regression (SVR), shallow MLPs, or compressed deep networks.
 - Consider on-device retraining or field adaptation.
 - Example:* CRDNN architectures were successfully deployed on mobile construction equipment ECUs with field-updatable weights [115].
- *Sufficient Compute Available (GPU/Server):*
 - Use complex DNNs, RL controllers, or multi-agent architectures.
 - Example:* Beckman et al. [32] implemented a multi-agent RL structure with image processing for real-time excavation tasks.
- *Deployment Environment:*
 - Simulation-only use cases allow more freedom in architecture and training time.
 - Hardware deployment requires guaranteed inference time, robust fail-safe handling, and explainability.
 - Example:* Reinforcement learning for hydraulic log manipulators was only deployed after simulation-to-real validation using synthetic training data [46].

4. Plan for Simulation-to-Real Transfer

- *Incorporate domain randomization and noise injection to improve robustness.*
 - Example:* Dhakate et al. [46] trained agents in simulation and transferred policies to real hardware with minimal tuning.
- *Start with supervised modeling for system understanding, then extend to RL or hybrid models.*
 - Example:* Egli et al. [50] used supervised learning to build a behavior model before applying RL for actuator control on an excavator.
- *Use hybrid models (data-driven + physics-informed) to enhance accuracy and generalizability.*
 - Example:* Mesmer et al. [61] used both simulation and physical knowledge in modeling a hydraulic clutch control system, leading to a significant shift in quality improvement.

5. Implement Monitoring and Maintenance Strategies

- *Compare model predictions continuously with live sensor feedback.*
→ Integrate automated health checks to detect drift or degradation.
- *Retrain models periodically using updated field data.*
→ Adapt to wear, seasonal variations, or new operating conditions. *Example:* Yang et al. [48] demonstrated seasonal sensitivity of loading tasks and addressed it with retrained attention-based models.
- *Adjust thresholds in anomaly detection systems as system behavior evolves.*
→ Use online learning or threshold calibration strategies. *Example:* Helwig et al. [88] applied adaptive thresholding with feature selection to handle changing system dynamics in real-time.

6 Summary and Concluding Remarks – Challenges and Trends

This chapter summarizes the previously presented research and highlights the current challenges and trends in applying machine learning to fluid power.

Current Applications of Machine Learning in Fluid Power

1. **Energy Efficiency:** Machine learning models support system designers to identify optimized control and behavior parameters in real time to reduce wasted energy. Adaptive control algorithms enable rapid adjustments based on system conditions, facilitating the optimal balance between improved efficiency and loss of functionality.
2. **Fault Detection and Diagnostics:** The complexity of modern fluid power systems challenges maintenance experts to assign unwanted system behavior to specific components or working states. Machine learning algorithms enable fault diagnosis and anomaly detection by simultaneously analyzing and correlating multiple sensor signals.
3. **Predictive Maintenance:** Deployment of predictive maintenance applications continuously increases in fluid power systems. With the help of machine learning algorithms, large amounts of sensor data can be analyzed and aid in early failure detection and lifetime prediction, thus increasing system reliability. By substituting reactive and preventive maintenance with predictive maintenance, downtime and costs can be minimized.

4. **Control Strategies:** Increasing the complexity of machinery results in increased complexity of control strategies. Machine Learning models can aid during development and in real-world deployment to improve system behavior by tuning all available parameters in parallel.

Key Challenges for Machine Learning in Fluid Power

1. **Data Availability:** For machine learning models to perform well, a sufficient amount of training data is essential. While mobile machines may not always be equipped with high-frequency sensors, this does not necessarily prevent effective data collection. In many cases, meaningful datasets can be obtained using standard sensor setups, depending on the specific application and modeling goals. Nevertheless, limited data, regardless of sampling frequency, can still pose challenges for model training and generalization, particularly in complex or highly variable operating conditions.
2. **Complexity of Fluid Power System:** The system or controller model needs to be as detailed as possible to use machine learning for condition monitoring or to improve control tasks.
Specifically for supervised learning tasks, unlabeled data of a specific behavior or failure can result in poor prediction performance. Similarly, control functions depend on information about the components and their boundaries. In the case of a hydraulic system, for example, a pressure or temperature threshold can represent a boundary, whereas an impossible negative movement of a cylinder can represent necessary information about a component—the need for system information scales with the complexity of the system and deployment.
3. **Interdisciplinary Knowledge:** Applying machine learning in fluid power systems requires deep functional knowledge about the system and an understanding of the different forms of machine learning and their utilization. While necessary, acquiring expert knowledge in both fields poses a significant challenge for individual experts.
4. **Transfer to Real-World:** While simulations display the opportunities and potential of ML in fluid power systems, the adaptation into real-world machinery is challenging. Changes in working behavior due to environmental changes are difficult to model, which is why extensive real-world tests are crucial. In addition, the black-box-like decision-making of some machine learning algorithms necessitates rigorous testing to ensure compliance with safety regulations.

Trends

1. **Hybrid Models:** In recent years, combining machine learning models with physical formulas marks the next step in the evolution of ML deployment in fluid power systems. Studies such as [126, 127, 128, 129, 130, 131, 132, 133, 134] represent initial efforts to establish frameworks for physics-informed machine learning applications in this field. Integrating domain-specific expertise with data-driven insights can enhance the accuracy and robustness of predictions.
2. **Explainable AI:** Current machine learning models often present a black-box-like form. Its decision-making reasoning is beyond the user's understanding, which leads to skepticism, especially in critical, safety-relevant applications. Countering the fear requires explainable AI, allowing users to build trust by comprehending the model's decisions. Simpler models, like decision trees or linear models, often perform as well as neural networks but are far easier to interpret. Alternatively, close documentation of the output of a neural network regarding defined inputs can help increase the model's transparency. This enables a general understanding of the relationship between specific features and the output.
3. **Edge Computing:** Enabling the potential of machine learning in fluid power systems requires powerful local computational data-processing engines to handle large amounts of data, as well as its analysis and interpretation. This is especially true for control tasks, as the controller must process input signals in real-time and compute the desired control output with minimal latency.

The impact of machine learning on fluid power is evident in the presented research. This technological advancement represents the next step in the ongoing digitalization process, offering vital support to engineers to improve efficiency, reliability, and sustainability. Still, substantial research and adaptation are required to fully utilize machine learning's capabilities in fluid power. Nonetheless, the importance and potential of machine learning ensure that its influence and deployment will continue to grow in the coming years.

References

- [1] T. C. Silva and L. Zhao, *Machine Learning in Complex Networks*. ProQuest Ebook Central, Cham: Springer International Publishing, 1st ed. 2016 ed., 2016.

- [2] R. S. Sutton and A. Barto, *Reinforcement learning: An introduction*. Adaptive computation and machine learning, Cambridge, Massachusetts and London, England: The MIT Press, second edition ed., 2020.
- [3] Y. Li, “Deep reinforcement learning,” *Under review for Morgan & Claypool: Synthesis Lectures in Artificial Intelligence and Machine Learning*, arXiv, 2018.
- [4] M. Raghu and E. Schmidt, “A survey of deep learning for scientific discovery,” 2020-03-26.
- [5] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. O’Reilly Media, Inc., 2022.
- [6] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.
- [7] Stanford Computer Science, “Convolutional neural networks for visual recognition,” <https://cs231n.github.io/neural-networks-1/>, last visited on 2023-02-22.
- [8] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [9] Stanford Edu, “Multi-layer neural network,” <http://ufldl.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks/>, last visited on 2023-02-22.
- [10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [11] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” 2019.
- [12] M. Candela-Leal, E. Gutiérrez-Flores, G. Presbítero-Espinosa, A. Sujatha-Ravindran, R. Ramirez-Mendoza, J. Lozoya-Santos, and M. A. Ramírez Moreno, “Multi-output sequential deep learning model for athlete force prediction on a treadmill using 3d markers,” *Applied Sciences*, vol. 12, p. 5424, 05 2022.
- [13] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, vol. 103. New York, NY: Springer New York, 2013.

- [14] Stanford Edu, “Pca whitening,” <http://ufldl.stanford.edu/tutorial/unsupervised/PCAWhitening/>, last visited on 2023-02-22.
- [15] Stanford Edu, “Autoencoders,” <http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/>, last visited on 2023-02-22.
- [16] D. P. Kingma and M. Welling, “An introduction to variational autoencoders,” *Foundations and Trends in Machine Learning*, 2019.
- [17] David Silver, “Lecture 1: Introduction to reinforcement learning,” <http://www.davidsilver.uk/teaching/>, last visited on 2023-02-22.
- [18] R. Bellman, *Dynamic programming*. Princeton landmarks in mathematics, Princeton: Princeton University Press, 2010.
- [19] L. Wang, K. C. Tan, and D. Liu, “A survey on reinforcement learning for control optimization in engineering systems,” *Journal of Control and Decision*, vol. 3, no. 1, pp. 19–43, 2016.
- [20] W. Su, W. Ren, H. Sun, C. Liu, X. Lu, Y. Hua, H. Wei, and H. Jia, “Data-based flow rate prediction models for independent metering hydraulic valve,” *Energies*, vol. 15, no. 20, p. 7699, 2022.
- [21] A. Widodo and B.-S. Yang, “Support vector machine in machine condition monitoring and fault diagnosis,” *Mechanical systems and signal processing*, vol. 21, no. 6, pp. 2560–2574, 2007.
- [22] M. Xia, T. Li, L. Xu, L. Liu, and C. W. De Silva, “Fault diagnosis for rotating machinery using multiple sensors and convolutional neural networks,” *IEEE/ASME transactions on mechatronics*, vol. 23, no. 1, pp. 101–110, 2017.
- [23] E. Jove, J.-L. Casteleiro-Roca, H. Quintián, J. A. Méndez-Pérez, and J. L. Calvo-Rolle, “A fault detection system based on unsupervised techniques for industrial control loops,” *Expert Systems*, vol. 36, no. 4, p. e12395, 2019.
- [24] N. Thuerey, K. Weißenow, L. Prantl, and X. Hu, “Deep learning methods for reynolds-averaged navier–stokes simulations of airfoil flows,” *AIAA journal*, vol. 58, no. 1, pp. 25–36, 2020.
- [25] J. Rabault and A. Kuhnle, “Accelerating deep reinforcement learning strategies of flow control through a multi-environment approach,” *Physics of Fluids*, vol. 31, p. 094105, 09 2019.
- [26] N. Thuerey, K. Weißenow, L. Prantl, and X. Hu, “Deep learning methods for reynolds-averaged navier–stokes simulations of airfoil flows,” *AIAA Journal*, vol. 58, p. 25–36, Jan. 2020.
- [27] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, “Deep learning and its applications to machine health monitoring: A survey,” 2016.

- [28] A. Malhi and R. Gao, “Pca-based feature selection scheme for machine defect classification,” *Instrumentation and Measurement, IEEE Transactions on*, vol. 53, pp. 1517–1525, 01 2005.
- [29] B. Song and A. J. Koivo, “Neural adaptive control of excavators,” in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, pp. 162–167, IEEE Comput. Soc. Press, 1995.
- [30] Z. Hong, D. Kaifang, and L. Tingqi, “A online-trained neural network controller for electro-hydraulic servo system,” in *Proceedings of the 4th World Congress on Intelligent Control and Automation (Cat. No.02EX527)*, pp. 2983–2986, IEEE, 2002.
- [31] I. Kurinov, G. Orzechowski, P. Hamalainen, and A. Mikkola, “Automated excavator based on reinforcement learning and multibody system dynamics,” *IEEE Access*, vol. 8, pp. 213998–214006, 2020.
- [32] S. Backman, D. Lindmark, K. Bodin, M. Servin, J. Mörk, and H. Löfgren, “Continuous control of an underground loader using deep reinforcement learning,” *Machines*, vol. 9, no. 10, p. 216, 2021.
- [33] J. Andersson, K. Bodin, D. Lindmark, M. Servin, and E. Wallin, “Reinforcement learning control of a forestry crane manipulator,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.
- [34] B. Kazenwadel and M. Geimer, “Real-time prediction of efficient operating points in quasi-stationary agricultural processes with hydraulic implements,” 11 2023.
- [35] B. J. Hodel, “Learning to operate an excavator via policy optimization,” *Procedia Computer Science*, vol. 140, pp. 376–382, 2018.
- [36] J. Zhang, Q. Wang, and T. Wang, “A two-stage optimal controller for hydraulic loading system with mechanical compensation,” in *2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pp. 304–309, IEEE, 2019.
- [37] F. Brumand-Poor, G. Matthiesen, and K. Schmitz, “Control of a hydromechanical pendulum with a reinforcement learning agent,” in *13. Internationales Fluidtechnisches Kolloquium, IFK, Aachen*, 2022.
- [38] F. Brumand-Poor, L. Kauderer, G. Matthiesen, and K. Schmitz, “Application of deep reinforcement learning control of an inverted hydraulic pendulum,” *International Journal of Fluid Power*, 2023.
- [39] C. Lizalde, A. Loukianov, and E. Sanchez, “Force tracking neural control for an electro-hydraulic actuator via second order sliding mode,” in

- Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005*, pp. 292–297, IEEE, 2005.
- [40] Y.-g. Peng, J. Wang, and W. Wei, “Model predictive control of servo motor driven constant pump hydraulic system in injection molding process based on neurodynamic optimization,” *Journal of Zhejiang University SCIENCE C*, vol. 15, no. 2, pp. 139–146, 2014.
- [41] A. A. Aly, “Flow rate control of variable displacement piston pump with pressure compensation using neural network,” *JES. Journal of Engineering Sciences*, vol. 35, no. 6, pp. 1401–1412, 2007.
- [42] T. Reidl, P. Schraft, J. Weber, and S. Ihlenfeldt, “Loss optimal control strategy of speed and displacement variable electrohydrostatic axes,” 04 2021.
- [43] B. Kazenwadel, S. Becker, M. Graf, and M. Geimer, “Aligning process quality and efficiency in agricultural soil tillage,” *at – Automatisierungstechnik*, vol. 71, pp. 979–986, 11 2023.
- [44] M. Wydra, A. Bauer, G. Chris, and M. Geimer, “Moderne steuerungsalgorithmen für forstkräne mittels künstlichen neuronalen netzen imitieren und optimieren/imitate and optimize modern control algorithms for forestry cranes by means of artificial neural networks,” *Landtechnik*, vol. 75, pp. 118–140, 06 2020.
- [45] H. Chen, “Research of the electro-hydraulic servo system based on rbf fuzzy neural network controller,” *Journal of Software*, vol. 7, no. 9, 2012.
- [46] R. Dhakate, C. Brommer, C. Bohm, H. Gietler, S. Weiss, and J. Steinbrener, “Autonomous control of redundant hydraulic manipulator using reinforcement learning with action feedback,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7036–7043, IEEE, 2022.
- [47] L. Brinkschulte, M. Graf, and M. Geimer, “Reinforcement learning: A control approach for reducing component damage in mobile machines,” 10 2020.
- [48] W. Yang, N. Strokina, J. Pajarinen, J. Vihonen, M. M. Aref, and J.-K. Kämäräinen, “Neural network controller for autonomous pile loading revised,” *IEEE International Conference on Robotics and Automation (ICRA), Xi’an, China,*, pp. 2198–2204, 2021.
- [49] E. Halbach, J. Kamarainen, and R. Ghabcheloo, “Neural network pile loading controller trained by demonstration,” in *2019 International*

- Conference on Robotics and Automation (ICRA)*, pp. 980–986, IEEE, 2019.
- [50] P. Egli and M. Hutter, “A general approach for the automation of hydraulic excavator arms using reinforcement learning,” *IEEE Robotics and Automation Letters*, pp. 5679–5686, vol. 7, no. 2, 2022.
- [51] Q. Zhu and Q.-f. Wang, “Real-time energy management controller design for a hybrid excavator using reinforcement learning,” *Journal of Zhejiang University-SCIENCE A*, vol. 18, no. 11, pp. 855–870, 2017.
- [52] C. Pohlandt and M. Geimer, “Self-learning control strategy for electrified off-highway machines to optimize energy efficiency,” *SAE International Journal of Commercial Vehicles*, vol. 8, pp. 513–518, 09 2015.
- [53] M. Karpenko, J. Anderson, and N. Sepehri, “Coordination of hydraulic manipulators by reinforcement learning,” in *2006 American Control Conference*, p. 6 pp, IEEE, 2006.
- [54] M. Karpenko, N. Sepehri, and J. Anderson, “Decentralized coordinated motion control of two hydraulic actuators handling a common object,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 729–741, 2007.
- [55] K. Guo, M. Li, W. Shi, and Y. Pan, “Adaptive tracking control of hydraulic systems with improved parameter convergence,” *IEEE Transactions on Industrial Electronics*, vol. 69, no. 7, pp. 7140–7150, 2022.
- [56] S.-W. Kim, B. Cho, S. Shin, J.-H. Oh, J. Hwangbo, and H.-W. Park, “Force control of a hydraulic actuator with a neural network inverse model,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2814–2821, 2021.
- [57] F. Kreutmayr and M. Imlauer, “Application of machine learning to improve to performance of a pressure-controlled system,” in *Volume 1 – Symposium*, pp. 77–86, Technische Universität Dresden, 12.–14.10.2020.
- [58] D. Wyrwal, T. Lindner, P. Nowak, and M. Bialek, “Control strategy of hydraulic cylinder based on deep reinforcement learning,” in *2020 International Conference Mechatronic Systems and Materials (MSM)*, pp. 1–5, IEEE, 2020.
- [59] D. T. Liem, D. Q. Truong, H. G. Park, and K. K. Ahn, “A feedforward neural network fuzzy grey predictor-based controller for force control of an electro-hydraulic actuator,” *International Journal of Precision Engineering and Manufacturing*, vol. 17, no. 3, pp. 309–321, 2016.

- [60] T. Shan, L. Li, X. Wu, and S. Cheng, "Pressure control based on reinforcement learning strategy of the pneumatic relays for an electric-pneumatic braking system," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, p. 095440702211088, 2022.
- [61] F. Mesmer, T. Szabo, and K. Graichen, "Learning feedforward control of a hydraulic clutch actuation path based on policy gradients," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 585–590, IEEE, 2020.
- [62] T. Bécsi, S. Aradi, Á. Szabó, and P. Gáspár, "Policy gradient based reinforcement learning control design of an electro-pneumatic gearbox actuator," *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 405–411, 2018.
- [63] T. Becsi, A. Szabo, B. Kovari, S. Aradi, and P. Gaspar, "Reinforcement learning based control design for a floating piston pneumatic gearbox actuator," *IEEE Access*, vol. 8, pp. 147295–147312, 2020.
- [64] X.-x. Yin, Y.-g. Lin, and W. Li, "Predictive pitch control of an electro-hydraulic digital pitch system for wind turbines based on the extreme learning machine," *Transactions of the Institute of Measurement and Control*, vol. 38, no. 11, pp. 1392–1400, 2016.
- [65] F. Theljani, K. Laabidi, M. Lahmari-Ksouri, and S. Zidi, "New approach for systems monitoring based on semi-supervised classification," in *2011 International Conference on Communications, Computing and Control Applications (CCCA)*, pp. 1–6, IEEE, 2011.
- [66] V. Vígolo, V. Boyko, J. Weber, A. Valdiero, and V. De Negri, "Online monitoring of pneumatic actuation system for energy efficiency and dynamic performance," 11 2023.
- [67] E. Emad, J. Weber, and M. Wahler, "A machine learning approach for tracking the torque losses in internal gear pump – ac motor units," in *10th International Fluid Power Conference (10. IFK)*, 2016.
- [68] H. M. Castilho, C. L. Nascimento, and W. O. L. Vianna, "Aircraft bleed valve fault classification using support vector machines and classification trees," in *2018 Annual IEEE International Systems Conference (SysCon)*, pp. 1–7, IEEE, 2018.
- [69] V. Vígolo, V. Boyko, J. Weber, and V. De Negri, "A model-based approach for online optimization of pneumatic drives," *IEEE/ASME Transactions on Mechatronics*, vol. PP, pp. 1–11, 01 2024.
- [70] H. Sano, J. P. P. Malere, and L. Berton, "Single and multiple failures diagnostics of pneumatic valves using machine learning," *Conference*

Encontro Nacional de Inteligência Artificial e Computacional, Brasil, 2019.

- [71] P. Borriello, A. Pawar, E. Frosina, F. Tessicini, A. Vacca, and A. Senatore, “Detection of typical manufacturing errors in external gear machines using numerical simulation and data driven machine learning,” 11 2023.
- [72] A. M. Gareev, E. Y. Minaev, D. M. Stadnik, N. S. Davydov, V. I. Protsenko, I. A. Popelniuk, A. V. Nikonorov, and A. G. Gimadiev, “Machine-learning algorithms for helicopter hydraulic faults detection: model based research,” *Journal of Physics: Conference Series*, vol. 1368, no. 5, p. 052027, 2019.
- [73] V. Singh, A. Abdul Azeez, and T. Minav, “Intelligent approach to enhance redundancy in novel steer-by-wire for heavy earth moving machinery,” 06 2024.
- [74] F. Kosova and H. O. Unver, “A digital twin framework for aircraft hydraulic systems failure detection using machine learning techniques,” *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, p. 095440622211326, 2022.
- [75] V. Zakharov, A. Abdul Azeez, X. Han, and T. Minav, “The impact of electric drive structures on sensorless ai-based hydraulic valve fault classification,” 11 2023.
- [76] N. Keller, A. Sciancalepore, and A. Vacca, “Condition monitoring of an axial piston pump on a mini excavator,” *International Journal of Fluid Power*, 05 2023.
- [77] N. Keller, A. Sciancalepore, and A. Vacca, “Demonstrating a condition monitoring process for axial piston pumps with damaged valve plates,” *International Journal of Fluid Power*, 02 2022.
- [78] K. Shen and D. Zhao, “Fault diagnosis for aircraft hydraulic systems via one-dimensional multichannel convolution neural network,” *Actuators*, vol. 11, no. 7, p. 182, 2022.
- [79] A. Abdul Azeez, E. Vuorinen, T. Minav, and P. Casoli, “Ai-based condition monitoring of a variable displacement axial piston pump,” 06 2022.
- [80] P. Borriello, A. Senatore, F. Tessicini, G. Ricucci, and E. Frosina, “A fault detection strategy for an epump during eol tests based on a knowledge-based vibroacoustic tool and supervised machine learning classifiers,” *Meccanica*, vol. 59, pp. 1–26, 02 2024.

- [81] Y. Zhang, A. Vacca, G. Gong, and H. Yang, “Quantitative fault diagnostics of hydraulic cylinder using particle filter,” *Machines*, vol. 11, no. 11, p. 1019, 2023.
- [82] T. T. Le, J. Watton, and D. T. Pham, “An artificial neural network based approach to fault diagnosis and classification of fluid power systems,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 211, no. 4, pp. 307–317, 1997.
- [83] J. Huang, X. Wang, D. Wang, Z. Wang, and X. Hua, “Analysis of weak fault in hydraulic system based on multi-scale permutation entropy of fault-sensitive intrinsic mode function and deep belief network,” *Entropy (Basel, Switzerland)*, vol. 21, no. 4, 2019.
- [84] F. Makansi and K. Schmitz, “Simulation-based data sampling for condition monitoring of fluid power drives,” *IOP Conference Series: Materials Science and Engineering*, vol. 1097, no. 1, p. 012018, 2021.
- [85] F. Makansi and K. Schmitz, “Data-driven condition monitoring of a hydraulic press using supervised learning and neural networks,” *Energies*, vol. 15, no. 17, p. 6217, 2022.
- [86] F. Makansi and K. Schmitz, “Fault detection and diagnosis for a hydraulic press by use of a mixed domain database,” in *BATH/ASME 2022 Symposium on Fluid Power and Motion Control*, American Society of Mechanical Engineers, 2022.
- [87] A. El-Betar, M. Abdelhamed, A. El-Assal, and R. Abdelsatar, “Fault diagnosis of a hydraulic power system using an artificial neural network,” *Journal of King Abdulaziz University-Engineering Sciences*, vol. 17, no. 1, pp. 115–136, 2006.
- [88] N. Helwig, E. Pignanelli, and A. Schutze, “Condition monitoring of a complex hydraulic system using multivariate statistics,” in *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, pp. 210–215, IEEE, 2015.
- [89] A. Gareev, V. Protsenko, D. Stadnik, P. Greshniakov, Y. Yuzifovich, E. Minaev, A. Gimadiev, and A. Nikonorov, “Improved fault diagnosis in hydraulic systems with gated convolutional autoencoder and partially simulated data,” *Sensors (Basel, Switzerland)*, vol. 21, no. 13, 2021.
- [90] A. M. Gareev, E. V. Shakhmatov, A. B. Prokofev, and D. M. Stadnik, “Machine learning method for predicting remaining useful life of hydraulic equipment,” *Journal of Machinery Manufacture and Reliability*, vol. 51, no. 3, pp. 253–260, 2022.

- [91] A. Mallak and M. Fathi, “Sensor and component fault detection and diagnosis for hydraulic machinery integrating lstm autoencoder detector and diagnostic classifiers,” *Sensors (Basel, Switzerland)*, vol. 21, no. 2, 2021.
- [92] Z. Huijie, R. Ting, W. Xinqing, Z. You, and F. Husheng, “Fault diagnosis of hydraulic pump based on stacked autoencoders,” in *2015 12th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, pp. 58–62, IEEE, 2015.
- [93] X. Tao, Z. Wang, J. Ma, and H. Fan, “Study on fault detection using wavelet packet and som neural network,” in *Proceedings of the IEEE 2012 Prognostics and System Health Management Conference (PHM-2012 Beijing)*, pp. 1–5, IEEE, 2012.
- [94] J. Bei, C. Lu, and Z. Wang, “Performance assessment and fault diagnosis for hydraulic pump based on wpt and som,” *Vibroengineering Procedia*, vol. 2, pp. 23–28, 01 2013.
- [95] S. Tang, S. Yuan, Y. Zhu, and G. Li, “An integrated deep learning method towards fault diagnosis of hydraulic axial piston pump,” *Sensors (Basel, Switzerland)*, vol. 20, no. 22, 2020.
- [96] S. Tang, Y. Zhu, S. Yuan, and G. Li, “Intelligent diagnosis towards hydraulic axial piston pump using a novel integrated cnn model,” *Sensors (Basel, Switzerland)*, vol. 20, no. 24, 2020.
- [97] Y. Zhu, G. Li, R. Wang, S. Tang, H. Su, and K. Cao, “Intelligent fault diagnosis of hydraulic piston pump based on wavelet analysis and improved alexnet,” *Sensors (Basel, Switzerland)*, vol. 21, no. 2, 2021.
- [98] C. Lu, N. Ma, and Z. Wang, “Fault detection for hydraulic pump based on chaotic parallel rbf network,” *EURASIP Journal on Advances in Signal Processing*, vol. 2011, no. 1, 2011.
- [99] S. Rajakarunakaran, D. Devaraj, and K. S. Rao, “Fault detection in centrifugal pumping systems using neural networks,” *International Journal of Modelling, Identification and Control*, vol. 3, no. 2, p. 131, 2008.
- [100] M. Karpenko and N. Sepehri, “A neural network based fault detection and identification scheme for pneumatic process control valves,” in *E-systems and e-man for cybernetics in cyberspace*, (Piscataway, NJ), pp. 93–98, IEEE Service Center, 2001.
- [101] M. Karpenko, N. Sepehri, and D. Scuse, “Diagnosis of process valve actuator faults using a multilayer neural network,” *Control Engineering Practice*, vol. 11, no. 11, pp. 1289–1299, 2003.

- [102] Y. Duensing, A. Rodas Rivas, and K. Schmitz, "Machine learning for failure mode detection in mobile machinery," *Karlsruhe : KIT Scientific Publishing, Kolloquium Mobilhydraulik, Karlsruhe*, vol. 11, pp. 1–26, 2020.
- [103] Y. Joo, K. Kim, and J. Jeong, "Performance comparison of machine learning algorithms for imbalanced class classification in hydraulic system," in *2020 14th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, pp. 1–8, IEEE, 2020.
- [104] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
- [105] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: A new over-sampling method in imbalanced data sets learning," in *Advances in intelligent computing* (D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, D.-S. Huang, X.-P. Zhang, and G.-B. Huang, eds.), vol. 3644 of *Lecture Notes in Computer Science*, pp. 878–887, Berlin: Springer, 2005.
- [106] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 1322–1328, IEEE, 2008.
- [107] I. Tomek, "Two modifications of cnn," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, no. 11, pp. 769–772, 1976.
- [108] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [109] L. Breiman, "Random forest," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [110] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.
- [111] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.
- [112] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, 2001.
- [113] J. Liu, A. Sitte, and J. Weber, "Adaptive identification and application of flow mapping for electrohydraulic valves," 06 2021.

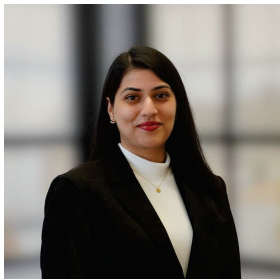
- [114] F. Brumand-Poor, F. Makansi, L. Jiakun, and K. Schmitz, “Implementation of a variational autoencoder for dimension reduction of a hydraulic system.” Global Fluid Power Symposium , Neapel, Global Fluid Power Symposium, Neapel, 2022.
- [115] Y. Xiang, T. Tang, T. Su, C. Brach, L. Liu, S. Mao, and M. Geimer, “Fast crdnn: Towards on site training of mobile construction machines,” *IEEE Access*, vol. PP, pp. 1–1, 09 2021.
- [116] W. Su, W. Ren, H. Sun, C. Liu, X. Lu, Y. Hua, H. Wei, and H. Jia, “Data-based flow rate prediction models for independent metering hydraulic valve,” *Energies*, vol. 15, no. 20, p. 7699, 2022.
- [117] S. Zhang, T. Chen, T. Minav, X. Cao, and A. Wu, “Position soft-sensing of direct-driven hydraulic system based on back propagation neural network,” *Actuators*, vol. 10, p. 18, 12 2021.
- [118] L. Baronti, B. Zhang, M. Castellani, and D. T. Pham, “Machine learning of electro-hydraulic motor dynamics,” *SN Applied Sciences*, vol. 2, no. 1, 2020.
- [119] B. Helian, X. An, Z. Yong, Z. Chen, and M. Geimer, “Lstm-based workload recognition for hydraulic actuators: A case study on excavator digging process,” 10 2024.
- [120] K. O. Achieng, “Evaluating pump performance using laboratory observations and machine learning,” *ISH Journal of Hydraulic Engineering*, vol. 27, no. sup1, pp. 174–181, 2021.
- [121] G. Chris, M. Geimer, and N. Maier, “Assistance system for an automated log-quality and assortment estimation based on data-driven approaches using hydraulic signals of forestry machines,” 06 2020.
- [122] H. Hosseiny, F. Nazari, V. Smith, and C. Nataraj, “A framework for modeling flood depth using a hybrid of hydraulics and machine learning,” *Scientific reports*, vol. 10, no. 1, p. 8222, 2020.
- [123] B. Helian, X. Huang, M. Yang, Y. Bian, and M. Geimer, “Computer vision-based excavator bucket fill estimation using depth map and faster r-cnn,” *Automation in Construction*, vol. 166, p. 105592, 07 2024.
- [124] L. Boden, M. Hofmeister, F. Brumand-Poor, L. Pleninger, and K. Schmitz, “Predicting compatibility of sealing material with bio hybrid fuels development and comparison of machine learning methods,” 09 2024.
- [125] L. Boden, M. Hofmeister, F. Brumand-Poor, L. Pleninger, and K. Schmitz, “Predicting compatibility of sealing material with bio

- hybrid fuels development and comparison of machine learning methods,” 09 2024.
- [126] F. Brumand-Poor, N. Bauer, N. Plückerhahn, and K. Schmitz, “Fast computation of lubricated contacts: A physics-informed deep learning approach,” *International Journal of Fluid Power*, vol. 19, pp. 1–12, 2024.
- [127] F. Brumand-Poor, N. Bauer, N. Plückerhahn, M. Thebelt, S. Woyda, and K. Schmitz, “Extrapolation of hydrodynamic pressure in lubricated contacts: A novel multi-case physics-informed neural network framework,” *Lubricants*, vol. 12, no. 4, p. 122, 2024.
- [128] F. Brumand-Poor, M. Rom, N. Plückerhahn, and K. Schmitz, “Physics-informed deep learning for lubricated contacts with surface roughness as parameter.” 63. Tribologie-Fachtagung 2022 , Göttingen , Germany, 2024.
- [129] F. Brumand-Poor, F. Barlog, N. Plückerhahn, M. Thebelt, and K. Schmitz, “Advancing lubrication calculation: A physics-informed neural network framework for transient effects and cavitation phenomena in reciprocating seals.” 22nd International Sealing Conference, Stuttgart, Germany, 2024.
- [130] F. Brumand-Poor, F. Barlog, N. Plückerhahn, M. Thebelt, N. Bauer, and K. Schmitz, “Physics-informed neural networks for the reynolds equation with transient cavitation modeling,” *Preprints*, October 2024.
- [131] A. Al-Issa and J. Weber, “Predicting hydraulic oil thermophysical properties using physics-informed neural networks,” *International Journal of Fluid Power*, 2024.
- [132] A. Al-Issa and J. Weber, “Predicting hydraulic oil thermophysical properties using physics-informed neural networks,” *International Journal of Fluid Power*, vol. 25, pp. 59–88, 07 2024.
- [133] A. Al-Issa, T. Schulze, and J. Weber, “Rapid thermal energy modeling and analysis of complex industrial hydraulic systems,” *International Journal of Fluid Power*, vol. 26, pp. 65–98, 04 2025.
- [134] L. Boden, F. Brumand-Poor, L. Pleninger, and K. Schmitz, “Enhancing predictive accuracy under data scarcity: Modeling molecular interactions to describe sealing material compatibility with bio-hybrid fuels,” *Physchem*, vol. 5, 04 2025.

Biographies



Faras Brumand-Poor received a bachelor's degree in electrical engineering from RWTH Aachen University in 2017, a master's degree in electrical engineering from RWTH Aachen University in 2019, and a master's degree in automation engineering from RWTH Aachen University in 2020, respectively. He is a Group Leader of the research groups Fluids and Smart Systems at the Institute for Fluid Power Drives and Systems at RWTH Aachen University. His research areas include machine learning, particularly deep learning, physics-based learning, fluid transmission lines, and virtual sensory systems.



Barkha Sharma received a bachelor's degree in computer science from the University of Stuttgart in 2021 and a master's degree in computer science from the University of Stuttgart in 2025. She is a Research Associate in the Smart Systems group at the Institute for Fluid Power Drives and Systems at RWTH Aachen University. Her research areas include machine learning, especially large language models and AI planning, as well as data analysis.



Yannick Duensing received a bachelor's degree in mechanical engineering from RWTH Aachen University in 2016 and a master's degree in mechanical engineering from RWTH Aachen University in 2019. He was a Research Associate at the Institute for Fluid Power Drives and Systems at RWTH Aachen University. His research areas include condition monitoring of mobile machinery and axial piston pumps, data analysis, and machine learning.



Panagiotis Logothetis received a Bachelor's Degree in Mechanical Engineering from RWTH Aachen University in 2022 and is currently pursuing a master's degree in Aeronautical Engineering and Astronautics at RWTH Aachen University.



Emre Atali Emre Atali received a bachelor's degree in mechanical engineering from RWTH Aachen University in 2021 and is currently pursuing a master's degree in Power Engineering at RWTH Aachen University.



Katharina Schmitz received a graduate's degree in mechanical engineering from RWTH Aachen University in 2010 and an engineering doctorate from RWTH Aachen University in 2015. She is currently the director of the Institute for Fluid Power Drives and Systems (ifas), RWTH Aachen University.

