Engbers, Hendrik; Freitag, Michael

Automated model selection for multivariate anomaly detection in manufacturing systems

Journal of Intelligent Manufacturing

Provided in Cooperation with:
Springer Nature

Suggested Citation: Engbers, Hendrik; Freitag, Michael (2024) : Automated model selection for multivariate anomaly detection in manufacturing systems, Journal of Intelligent Manufacturing, ISSN 1572-8145, Springer US, New York, NY, Vol. 36, Iss. 7, pp. 5015-5033, https://doi.org/10.1007/s10845-024-02479-z

Journal of Intelligent Manufacturing (2025) 36:5015–5033
https://doi.org/10.1007/s10845-024-02479-z

Automated model selection for multivariate anomaly detection in manufacturing systems

Hendrik Engbers[1] · **Michael Freitag**[1,2]

Received: 20 November 2023 / Accepted: 8 August 2024 / Published online: 21 September 2024

Abstract
As machine learning is widely applied to improve the efficiency and effectiveness of manufacturing systems, the automated selection of appropriate algorithms and hyperparameters becomes increasingly important. This paper presents a model selection approach to multivariate anomaly detection for applications in manufacturing systems using a multi-output regression-based meta-learning method. The proposed method exploits the capabilities of meta-learning to explore and learn the intricate relationships within multivariate data sets in order to select the best anomaly detection model. It also facilitates the construction of an ensemble of algorithms with dynamically assigned weights based on their respective performance levels. In addition to the framework, new meta-features for the application domain are presented and evaluated. Experiments show the proposed method can be successfully applied to achieve significantly better results than benchmark approaches. This enables an automated selection of algorithms that can be used for enhanced anomaly detection under changing operating conditions.

Keywords Meta-learning · Algorithm selection · Anomaly detection · Multivariate manufacturing data

Introduction

Digitization and availability of real-time data in manufacturing systems allow to continuously predict the state of machines and components. Machine learning algorithms for anomaly detection have proven to be very effective in this field (Tao et al., 2018; Wang et al., 2022). The goal of applying these algorithms is to detect patterns and relationships in process data that do not match expected behavior and thus indicate advanced wear or other critical processes (Chandola et al., 2009; Lopez et al., 2017). In this way, unplanned downtime can be avoided and maintenance can be scheduled more efficiently (Colledani and Tolio, 2012). To this end, appropriate algorithms must be selected, adapted and deployed. When faced with a large number of potentially suitable algorithms, identifying the optimal one and its hyperparameters is a challenge. Generally, these tasks require not only computational power, but also data science skills and domain-specific knowledge (Li et al., 2022). Automating these tasks could therefore be a great benefit for the application of machine learning techniques in manufacturing systems.

Machine learning bottlenecks

To better understand why an automated selection of algorithms is desirable, we will first have a look at where human activities and expertise are required in the modeling process. Let a machine be a bottleneck process in a manufacturing system and the monitoring system required to setup a Predictive Maintenance (PdM) is based on prognostic methods that use current sensor data to assess the condition of a machine or equipment. According to (Karmaker et al., 2022), two human actors are particularly relevant in this process: The domain expert and the data expert. The domain expert has extensive technical and organizational knowledge. The data expert has experience with predictive modeling but has limited knowl-

✉ Hendrik Engbers
eng@biba.uni-bremen.de

✉ Michael Freitag
fre@biba.uni-bremen.de

[1] BIBA - Bremer Institut für Produktion und Logistik GmbH, University of Bremen, Hochschulring 20, 28359 Bremen, Germany

[2] Faculty of Production Engineering, University of Bremen, Badgasteiner Straße 1, 28359 Bremen, Germany

edge of the specific application and process. Together, these specialists understand the environmental conditions under which the data is collected and formulate an overall target. The data expert translates the high-level target into a diagnostic or predictive task, extracts relevant features from the data, and selects an appropriate algorithm to solve the task. Essential steps such as understanding domain-specific attributes, formulating the prediction task, and partitioning the training and test data are performed. Once these steps have been completed, different machine learning algorithms can be applied (Karmaker et al., 2022).

It appears that an intensive exchange between data and domain experts is required not only in the formulation of the task, but also for data preparation and synthesis of the results. In these phases there is a bottleneck in terms of expertise and communication. In addition, the development of key features, the evaluation of alternative algorithms, and the validation of models require significant computational power and effort on the part of data experts (Karmaker et al., 2022). To facilitate the application of machine learning algorithms an automation of these tasks would be beneficial. There is also an opportunity to make the exchange between experts more efficient and to optimize the processes of data annotation and result compilation. At the same time, automated approaches to key feature development and algorithm evaluation can help reduce the workload on human experts. In general, by automating these critical processes, resources could be used more efficiently.

Approaches for automation

Automated Machine Learning (autoML) aims to automate feature and algorithm selection, and hyperparameter optimization (Hutter et al., 2019). In contrast to autoML systems, which start without prior knowledge to find a suitable algorithm-hyperparameter combination, meta-learning aims at using machine learning techniques to learn from past tasks for new tasks (Lemke and Gabrys, 2010; Gabbay et al., 2009; Lemke et al., 2015; Smith-Miles, 2008). According to Gabbay et al. (2009) 'Meta-learning is the study of principled methods that exploit meta-knowledge to obtain efficient models and solutions by adapting the machine learning and data mining process.'

The problem of selecting appropriate algorithms and hyperparameters arises from the No-Free-Lunch Theorem (NFL), which states that no single algorithm achieves the best results for a wide range of problems (Wolpert and Macready, 1997). In the context of PdM, it has been shown that this assumption also holds for anomaly detection. For example, Schmidl et al. (2022) evaluated over 70 state-of-the-art supervised, unsupervised, and semi-supervised algorithms on time series datasets from different domains. The results showed that no single algorithm was superior. They also found that deep learning methods are not yet competitive, despite their high training effort, and that simpler methods can quickly produce results almost as good as complex methods.

Therefore, the central hypothesis of this work is that an individual automatic selection of the best algorithm from a set of robust candidates can contribute to better anomaly detection, which, if the appropriate measures are derived from these insights, should also lead to more effective manufacturing systems. In this way, the manual effort for algorithm selection could be reduced. It would also allow for easy adaptation to changing conditions without the need for experts. Finally, we assume that the resulting selection could be superior to conventional approaches.

Meta-learning has been successfully applied to algorithm selection in several fields (Sect. 2). However, the design of this technique depends on the data structure and the demanded task (Ali et al., 2018). This work therefore takes the step of transferring the meta-learning approach to a new domain. For the first time, novel meta-features are introduced that describe the relationship between multivariate time series and domain-specific information, and a framework for meta-learning using multi-output regression is presented. Furthermore, in contrast to other methods it is suitable for both single algorithm selection and ensemble formation. In this way, the meta-model can predict the predictive performance of all algorithms in the set of candidate anomaly detectors for multivariate manufacturing data.

The rest of the paper is organized as follows. Section 2 reviews the state of the art in meta-learning for algorithm selection with a focus on anomaly detection. After identifying the research gap, the proposed method is presented in Sect. 3. Section 4 is dedicated to the evaluation of the method in terms of prediction performance and comparison with benchmark methods. In addition, the importance of the individual meta-features and the computational times of the candidate algorithms are examined. The paper concludes with a discussion of the results and a conclusion in Sect. 5.

Related works

One application of meta-learning is to select an appropriate algorithm for a particular use case. However, a further distinction must be made with respect to the task of the algorithm, such as whether it should be used for predictive tasks, and if so, for what types of predictive tasks. (Ali et al., 2018; Vanschoren, 2018) In general, a task $t_j \in T$ is described by a vector $m(t_j) = (m_{j,1}, \ldots, m_{j,n})$ of n meta-features $m_{j,n} \in M$. The distance between $m(t_i)$ and $m(t_j)$ can be determined to transfer information from an old task to a new one. Further, a meta learner L can be trained on earlier evaluation results to predict the performance $P_{i,\text{new}}$ of configurations λ_i of a new task t_{new}. Here P represents the

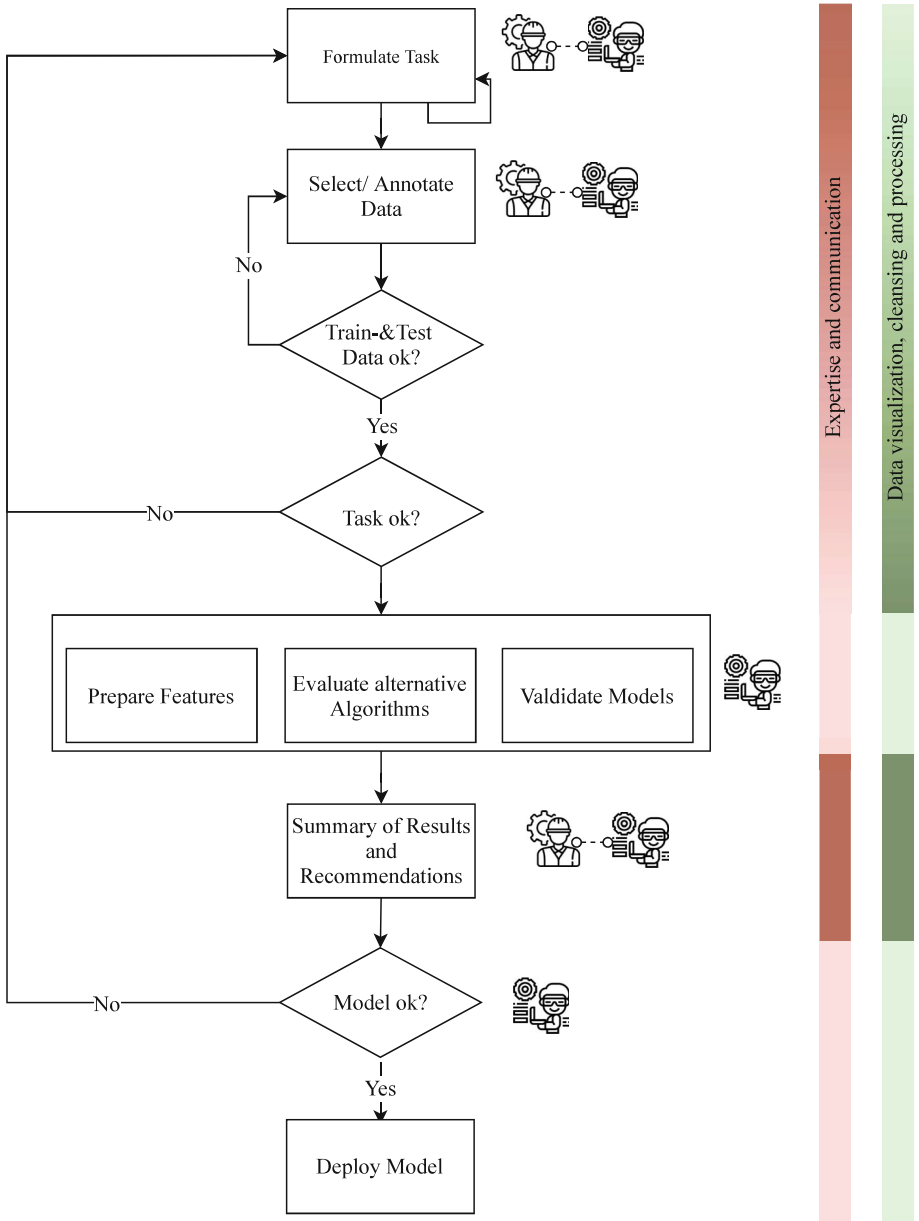


Fig. 1 Activities and communication of human experts for the implementation of a machine learning algorithm as propopsed by Karmaker et al. (2022)

set of all previous scalar evaluations $P_{i,j} = P(\lambda_i, t_j)$ of the configuration λ_i for the task t_j according to a suitable validation criterion and a model validation method (Bahri et al., 2022; Vanschoren, 2018). Figure 2 illustrates the concept of meta-learning as proposed by (Bahri et al., 2022).

Feature-based algorithm selection

According to Vanschoren (2018) meta-learning approaches can be distinguished based on what types of meta-features and how they are used. The concept depicted in Fig. 2 learns from existing model evaluations and meta-features to identify meaningful configurations of models and narrow down the search space. This can accelerate the search for optimal models and reduce computational effort, as only particularly relevant regions of the search space are explored (Zöller and Gabrys, 2020). Furthermore, there is the possibility of transferring suitable configurations to new tasks. Performance differences between various model configurations can be used as meta-features (Fürnkranz and Petrak, 2001), or placeholder models can be trained for existing tasks to weigh them depending on the similarity of a new task (Vanschoren, 2018). Additionally, the learning curves of candidate algorithms can be used as meta-features to evaluate the suitability of a configuration for new tasks (Leite and Brazdil, 2005). Another approach is meta-learning from the properties of different tasks, using a variety of meta-features to characterize the task and then select a suitable configuration (Mantovani,

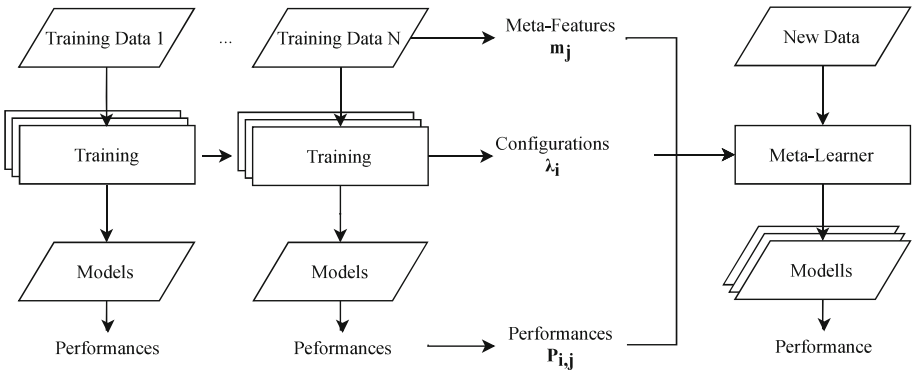


Fig. 2 The concept of meta-learning as proposed by Bahri et al. (2022)

2018; Reif et al., 2014; Alcobaça et al., 2020b; Vanschoren, 2010). Instead of manually defining meta-features, they can also be learned themselves to represent task groups (Sun et al., 2013). Furthermore, meta-features can be used to initiate an optimization process optimally or to predict the best configuration directly (Gomes et al., 2012). Approaches to constructing entire pipelines to select the best one for Bayesian optimization also fall into this category (Feurer et al., 2015; Fusi et al., 2017). At this point, it may also be useful to predict whether model optimization is promising (Ridd and Giraud-Carrier, 2014), or what improvement in prediction accuracy can be expected from optimization (Sanders and Giraud-Carrier, 2017). Another approach is meta-learning from previous models, such as their structure or learned model parameters. Here, an attempt is made to train a meta-learner to learn how a candidate algorithm should be trained for a new task. This approach can be further differentiated into transfer learning and meta-learning with Artifical Neural Networks (ANN). In transfer learning, models pre-trained on multiple tasks are used as a starting point for a new task. In meta-learning with ANN, neural networks are enabled to use changes in their model parameters during training as meta-features for further training (Baxter, 2019; Bengio, 2012; Caruana, 1994; Thrun and Mitchell, 1995). Learning with few data describes a variant of meta-learning that aims to adapt deep neural networks with only a few training data so that they are suitable for a new, similar task. In addition to all these mainly supervised learning approaches, meta-learning is by definition not limited to them (Bart and Ullman, 2005; Fei-Fei et al., 2006; Fink, 2004). Meta-learners can also be distinguished according to the goal of prediction. For example, the goal may be to create a ranking of suitable algorithms or entire configurations. For example, a ANN can be used to group similar tasks and then determine a ranking of the best configurations for these groups (Brazdil et al., 2003; Maforte Dos Santos et al., 2005). Alternatively, the meta-model can directly estimate the performance of a configuration (Köpf et al., 2000). A more detailed overview of these approaches can be found in the work of (Vanschoren, 2018).

Algorithm selection for anomaly detection

Anomalies are data points that significantly differ from other data points in a dataset (Hawkins, 1980). Anomalies can be caused by current disturbances and serve as early indicators of impending issues, such as declining production quality, decreasing machine functionality, or shortened component lifespan (Lughofer and Sayed-Mouchaweh, 2019; Lopez et al., 2017). The state of a machine can be defined by the degree of deviation from the expected operating behavior, with behavior being evaluated based on specific operating conditions (Vichare et al., 2004). For PdM, it is therefore essential to identify deviations from normal behavior to detect potential failures early and take necessary actions in time. Anomalies can be classified further into point, contextual, and collective anomalies (Schneider and Xhafa, 2022; Chandola et al., 2009). Point anomalies are individual data points that exhibit anomalous characteristics compared to the rest of the data. Contextual anomalies refer to data points that are considered anomalous only in a specific context or under certain conditions. In collective anomalies, a group of connected data points may be considered anomalous compared to the other data points, even if the individual instances do not represent anomalies themselves; in this case, the collective occurrence of these data points constitutes the anomaly (Cook et al., 2020). Figure 3 shows examples of these three types of anomalies.

The results of a model for anomaly detection can be in the form of anomaly scores or binary labels. Anomaly scores quantify the degree of deviation of each individual data point and allow ranking the data points according to their probability of being considered anomalous. Although this output contains all relevant information, unlike a label, it does not provide a concise summary of potential outliers. Binary labels indicate whether a data point is considered an outlier or not. Some algorithms output these labels directly, while others allow the conversion of anomaly scores into binary labels (Aggarwal, 2017).

In the field of meta-learning for algorithm selection, the following contributions to anomaly detection have been iden-

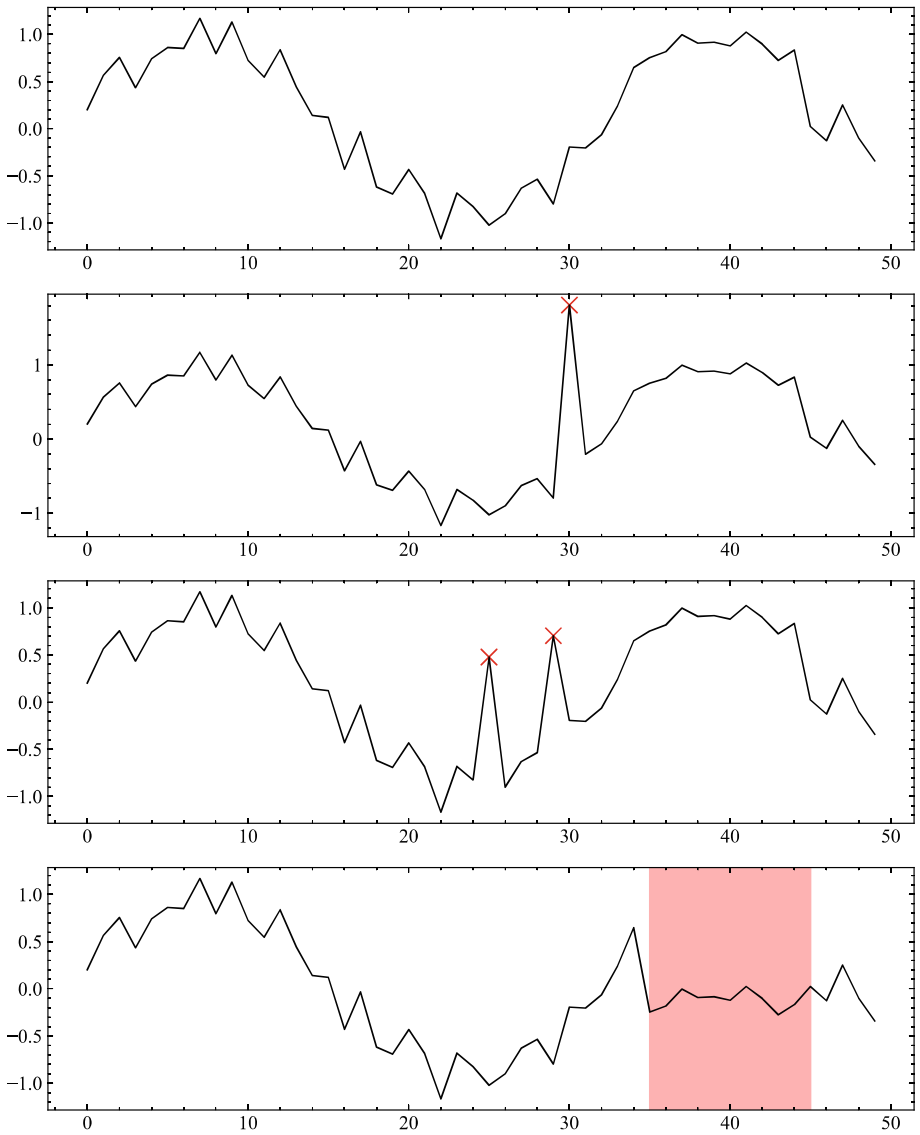


Fig. 3 Examples of point, context and collective anomalies

tified in the literature. Fu et al. (2022) propose a meta-learning method to detect anomalies in power dispatch data. They use a hybrid ensemble selection approach that combines different models based on meta-learning to improve the overall accuracy and generalizability of the detection model. Yu et al. (2022) developed a meta-learning method based on deep unsupervised learning for engine vibration anomaly detection. Vibration signals are transformed into 52 physical and statistical features to build models for different engines. Papastefanopoulos et al. (2021) present a meta-learning algorithm for unsupervised outlier detection that combines the best techniques of existing methods through ensemble voting and unsupervised feature selection. Zhao et al. (2021) propose to select a model based on the performance of many models on historical outlier detection benchmark datasets. The method uses specialized meta-features to capture task similarity within the meta-learning framework. Meta-learning approaches based on statistical and information-theoretic meta-features require large amounts of data and computational resources. To address this issue, Kotlar et al. (2021) propose a novel set of meta-features based on domain-specific properties of data that can be efficiently extracted or estimated using a small amount of data. This demonstrates that domain-specific features can provide high added value to algorithm selection. Tavares and Junior (2021) focused on the detection of anomalous traces in event logs, which can negatively affect the quality of process execution. The authors propose a meta-learning strategy that combines coding techniques with meta-feature extraction to improve anomaly detection performance. Poulakis et al. (2020) present a framework for automated clustering. The framework consists of two modules: algorithm selection and hyperparameter tuning. Algorithm selection relies on meta-learning with novel meta-features that capture sim-

ilarities in clustering structure, and hyper-parameter tuning uses Bayesian optimization with an optimization objective that combines different cluster validity indices. Cacoveanu et al. (2009) present a system that combines dataset characterizations with landmarking to increase prediction accuracy to select the best classifier for a dataset while minimizing user effort and providing flexibility.

Overall, the existing literature supports the idea that meta-learning is a viable method for selecting suitable algorithms for new tasks. However, there is a notable gap in understanding how to design a meta-learner effectively, as it heavily relies on the choice of meta-features, and their potential needs to be fully exploited. It is evident that the effectiveness of meta-features is contingent on the specific use case, indicating the potential benefits of incorporating domain-specific features. The range of meta-learning algorithms varies, from simpler voting mechanisms to deep-learning architectures. Various approaches, including ensemble selection, single model selection, tuning mechanisms, and coding techniques, have been explored. Despite the usefulness of landmarking features, their application is limited in PdM due to the absence of labels in most cases. Furthermore, existing research primarily involves one-time selection decisions, leaving a research gap in the development of automated procedures for dynamically selecting appropriate algorithms for anomaly detection. Meta-learning approaches tailored for multivariate time series are also lacking. This raises challenges in effectively describing individual time series within a dataset, aggregating this information, and recognizing relationships between the time series. Additionally, there is a lack of documented real-use cases in the literature that demonstrate the application of meta-learning in industrial settings, particularly in manufacturing processes. This paper attempts to develop a solution that addresses the problem of meta-learning for multivariate anomaly detection in manufacturing systems. For this purpose, an adequate framework for meta-learning and domain-specific meta-features are introduced and validated in the following sections.

Automated model selection approach

Since the suitability of anomaly detection algorithms depends on the available data and the specific application (Schmidl et al., 2022; Wolpert and Macready, 1997; Zhao et al., 2020), the challenge is to select the most appropriate algorithm for the different machines or components of a manufacturing system. Figure 4 illustrates this challenge for different of types of machines in a manufacturing system. The set of possible anomaly detection algorithms is represented by a collection of differently colored magnifying glasses. These colors represent their different characteristics and capabilities. The meta-model's task is to predict the performance of each of these candidates in order to make a reliable selection. It does this by using the incoming machine and operational data from each monitored object. At the bottom of the figure, the different colored magnifying glasses indicate that a different algorithm is selected for each machine. The selected algorithm is then used for anomaly detection. The green checkmark indicates that the machine is in normal condition. The orange triangle indicates that the machine is currently exhibiting abnormal behavior.

Formally, this challenge can be expressed as follows: Manufacturing systems comprise various types of machines $M = m^1, \ldots, m^m$, each with distinct configurations and workloads subject to fluctuating operating conditions. The state of each machine is denoted by a binary variable y_t^m, where:

$$y_t^m = \begin{cases} 0 & \text{for the state "normal"} \\ 1 & \text{for the state "anomalous"} \end{cases} \tag{1}$$

Assuming the dataset X^m implicitly encodes information about the machine's state y_t^m at time t, $A = a^1, \ldots, a^a$ denotes a set of candidate algorithms, where each algorithm a^i is linked with a set of hyperparameters within a domain Λ^i. If algorithm a^i contains p hyperparameters, the entire hyperparameter space is denoted as $\Lambda^1 = \lambda_i^1, ..., \lambda_p^1$. For X^m, the aim is to discover an algorithm and a hyperparameter configuration that minimizes the following loss:

$$(a^*, \lambda^*) \in \arg \min_{a^i \in A, \lambda \in \Lambda^i} \frac{1}{K} \sum_{j=1}^{K} L(a_\lambda^i, X_{train}^j, X_{test}^j), \tag{2}$$

where $L(a_\lambda^i, X_{\text{train}}^j, X_{\text{test}}^j)$ represents the loss experienced by algorithm a^i with hyperparameter λ on X_{test}^j when trained on X_{train}^j using K cross-validations.

Procedure overview

Figure 5 depicts the meta-learning process. The initial phase, conducted offline, entails the training and testing of the meta-model. To establish the training database, various candidate algorithms undergo evaluation on diverse dataset collections (Sect. 3.3). Their performances and hyperparameter configurations, are stored alongside 155 meta-features (Sect. 3.4).

For assessing the suitability of candidate algorithms and serving as a target for meta-learning, the F1-Score is selected as the primary evaluation criterion. The F1-Score offers a balanced assessment of both precision and recall, making it a sensible choice considering that the costs associated with false positives and false negatives depend on the application scenario. Moreover, the F1-Score proves advantageous

Table 1 Related works in the field of meta-learning for anomaly detection

Author(s)	Contribution
Fu et al. (2022)	Proposed meta-learning dynamic ensemble selection for power dispatching anomaly detection, addressing inaccurate benchmarks and poor universality, demonstrating advanced performance in a real-world application
Yu et al. (2022)	Introduced Meta-learning for motor vibration anomaly detection in industrial PHM, achieving a 33.50% accuracy improvement over unsupervised learning for new sensor models
Papastefanopoulos et al. (2021)	Developed a meta-learning algorithm for unsupervised outlier detection, combining strengths through ensemble voting and unsupervised feature selection, outperforming existing techniques
Zhao et al. (2021)	Proposed METAOD for automatic unsupervised outlier model selection based on meta-learning, leveraging performance on historical datasets for effective model selection, significantly outperforming other model selection techniques tailored for UOMS
Kotlar et al. (2021)	Addressed model selection in anomaly detection tasks within AutoML frameworks, introducing a novel set of domain-specific meta-features for efficient model selection, achieving 87% accuracy across diverse domains
Tavares and Junior (2021)	Enhanced anomaly detection in event logs using a meta-learning strategy with encoding, recommending suitable encoding techniques based on meta-features, outperforming baseline methods
Poulakis et al. (2020)	Introduced AutoClust, a framework for automated clustering, using meta-learning for algorithm selection and Bayesian optimization for hyperparameter tuning, demonstrating promising results on 24 datasets
Cacoveanu et al. (2009)	Presented an evolutional meta-learning framework for automatic classifier selection, combining dataset characterization with landmarking to increase prediction accuracy, aiming to minimize user intervention while offering flexibility in classifier selection

in scenarios where there is an uneven distribution of classes within the data.

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

In a preprocessing step, relevant meta-features are selected and transformed to attain the desired structure for subsequent processing. It is important to differentiate between actual meta-features, which describe the properties of the dataset and individual attributes, and performance measures utilized as meta-targets.

Before passing them to the meta-model for prediction, the meta-features may be encoded. For this work, the effectiveness of an autoencoder, a PCA (Pedregosa et al., 2011), as well as a t-Distributed Stochastic Neighbor Embedding (t-SNE) (Pedregosa et al., 2011) were investigated for dimensionality reduction. The autoencoder is used to learn a compact representation of the meta-features. In this case, we designed a dense autoencoder with eight fully connected layers. After dividing the data sets into training and test data, the training data is passed through the encoder. The test data are transformed accordingly. To construct the meta-model, a multi-output regression model is used to model the target variables, i.e., the performance measures of the candidate algorithms. In this case, the validation criterion is the Mean Squared Error (MSE) between the true and predicted target variables.

In the second phase, performed online, the meta-model is applied. The goal is to identify a suitable algorithm for a previously unknown application dataset. To achieve this, meta-features are extracted from the application dataset and submitted to the meta-model, which returns an estimate of each candidate's prognostic performances. Based on a

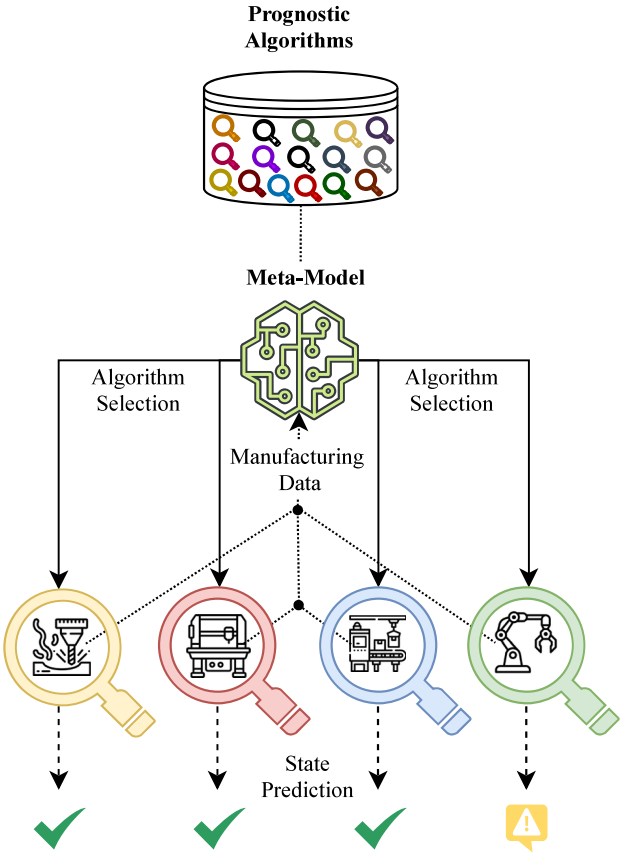


Fig. 4 Conceptual framework for prognostic algorithm selection in manufacturing systems

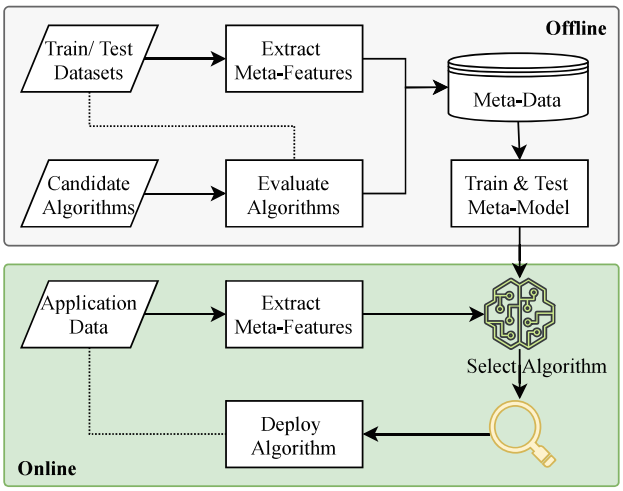


Fig. 5 Comprehensive flow of the meta-learning process for algorithm selection

selection mechanism that takes into account the performance predictions, a single or a combination of algorithms is selected. It is assumed that if there is a dominant algorithm in the candidate set according to the prediction of the meta-model, that this algorithm should be selected. Forming an ensemble of a very good algorithm and other less good algorithms would tend to result in lower prediction performance than using only the dominant algorithm. Forming an ensemble would also be advantageous when a dominant algorithm exists but cannot be accurately predicted. We further assume that in the absence of a particularly well-suited algorithm, an ensemble should be formed to compensate for uncertainties of the individual algorithms for this data set. The weighting of the algorithms within the ensemble is determined as a function of the predicted performance. Therefore, two approaches are considered: First, the selection of the best algorithm (Meta-Single-Selection (Meta-S-Sel)). Second, the formation of an ensemble of several good algorithms or the selection of a single algorithm (Meta-Multi-Selection (Meta-M-Sel)). For the second case, we define an upper bound σ_u and a lower bound σ_l. Algorithms whose predicted performance is below σ_l are excluded. Algorithms are added to the ensemble until the sum of the predicted performances exceeds the upper bound σ_u.

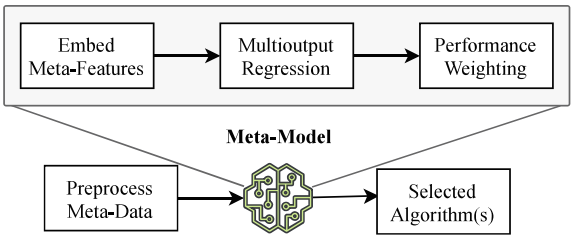


Fig. 6 Processing steps within the meta-model

Algorithm 1 Procedure to construct the Meta-M-Sel

Require: Estimated prognostic performance (F1-Score) of each candidate algorithm sorted in descending order ($AlgorithmPerformances$).
Require: Thresholds σ_u and σ_l.
Ensure: List $selectedAlgorithms$ containing the selected algorithms.
$selectedAlgorithms \leftarrow []$ ▷ Initialize an empty list for selected algorithms.
$currentSum \leftarrow 0$ ▷ Initialize sum of prognostic performances.
for $algorithm$ in $AlgorithmPerformances$ **do**
  if $algorithmPerformance < \sigma_l$ **then**
    continue ▷ Exclude algorithms below σ_l.
  end if
  $currentSum \leftarrow currentSum + algorithmPerformance$
  if $currentSum > \sigma_u$ **then**
    break ▷ Stop once σ_u is exceeded.
  end if
  $selectedAlgorithms \leftarrow selectedAlgorithms + [algorithm]$ ▷ Add algorithm to the ensemble.
end for
return $selectedAlgorithms$

Candidate algorithms for anomaly detection

To define the set of algorithms from which the meta-model is supposed to predict the most appropriate one, some boundary conditions need to be established. It is crucial that the algorithms excel in handling multivariate data. Moreover, the datasets may contain time series, such as sensor and control signals, in addition to categorical attributes describing production history or other relevant context. Therefore, the chosen algorithms should demonstrate proficiency in effectively processing these types of data. Another consideration arises from the necessity to pre-evaluate all candidate algorithms on the training datasets. Thus, it is imperative to minimize the number of hyperparameters to reduce the potential solutions and the computational complexity of preparing the meta-learning system. Additionally, a diverse set of anomaly detection algorithms should be available as candidates to ensure a comprehensive suitability profile. In light of these considerations, the following algorithms were selected as alternatives for the selection process and implemented using the PYOD library (Zhao et al., 2019): Histogram-related Outlier Score (HBOS), Principal Component Analysis (PCA), Clustering Based Local Outlier Factor (CBLOF), Lightweight Online Detector of Anomalies (LODA), Copula Based Outlier Detector (COPOD), Local Outlier Factor (LOF), One-Class Support Vector Machine (OCSVM), IsolationForest Outlier Detector (IForest), k-Nearest Neighbors Detector (KNN), Feature Bagging (FB) and Connectivity-Based Outlier Factor (COF).

Dataset collections for meta-model construction

The composition of the training datasets plays a pivotal role in shaping the performance and generalizability of the meta-model. The datasets, selected for the problem at hand, encompass multivariate data. The number of attributes per dataset varies from 2 to 21, while the length ranges from 148 to 5473 data points per attribute. Each data point must also be associated with a binary label, a prerequisite for evaluating the performance of the candidate algorithms. The construction of the meta-model relies on publicly available data collections, ensuring the reproducibility of the study (see Table 2). Care was taken to ensure that only about 80% of the datasets were used to train the meta-model and the remaining 20% were used for evaluation (Sect. 4).

The fundamental premise of this study is grounded in the concept of the No Free Lunch (NFL) theorem, which posits that no single algorithm consistently outperforms others across different tasks. Previous research by (Schmidl et al., 2022) has demonstrated that the NFL also applies to anomaly detection algorithms. However, it remains uncertain whether this assumption holds true for the specific algorithms and datasets considered in this study. Therefore, it is imperative to investigate how the performance of the various algorithms is distributed across the data, and whether there exists a dominant algorithm with consistently superior predictive capabilities for all datasets, thereby obviating the need for individual algorithm selection.

Figure 7 shows the predictive performance of various anomaly detection algorithms on the training datasets. The top 7a illustrates that there is no clear dominant algorithm; instead, many of the algorithms occupy valid positions within the candidate set. The bottom Fig. 7b further illustrates how algorithm performance varies from one dataset to another. Some algorithms yield impressive results for a particular dataset, while the average performance across all datasets is relatively poor. A direct comparison between the plots reveals that PCA achieves the highest median result (as shown in Fig. 7b), but it is not the absolute best algorithm overall (as shown in a). It is worth noting that the effectiveness of an algorithm on the training dataset does not necessarily guarantee similar performance on new incoming data or on a fresh test dataset. In summary, these results support the hypothesis that the NFL theorem holds for the given data and algorithms. Consequently, it should be advantageous to individually select algorithms for anomaly detection.

Meta-features to describe task similarity

The performance of the meta-model is significantly influenced by a thoughtful selection and effective incorporation of meta-features. It is crucial to describe datasets with meta-features that the meta-model can distinguish well and that also reflect properties defining the suitability of algorithms for specific tasks (Kotlar et al., 2021; Tavares and Junior, 2021).

The choice of meta-features is contingent on the structure and dimensionality of the input data. In this paper, we introduce a set of meta-features tailored for anomaly detection and multivariate time series. This set is complemented with previously established meta-features from the literature. Statistical summaries are employed for most meta-features to aggregate individual time series features for each dataset.

Multivariate time series are a common data structure for characterizing the state of technical systems. Often there is a causal relationship between several time series within the same data set. These relationships may contain crucial information for the selection of a suitable prognostic algorithm. Therefore, we introduce the following meta-features, which consider the properties of each time series in isolation, as well as its relationships to other time series within the same dataset. It is important to note that these features are not new for describing time series; their novelty lies in their application in this context to support the meta-learning process. Detailed information on the features used can be found in

Table 2 Open source dataset collections used for meta-feature generation and model construction, along with references and download sources

Dataset collection	Reference	Download source
Daphnet	doi:10.1109/TITB.2009.2036165	UCI ML
Exathlon	doi:10.14778/3476249.3476307	Github
Genesis	doi:10.1007/978-3-662-57805-6-4	Kaggle
GHL	arXiv:1612.06676	Paper Website
GutenTAG	doi:10.14778/3554821.3554873	Github
IOPS	doi10.1145/3292500.3330680	Challenge Website
KDD-TSAD	arXiv:2009.13807	Challenge Website
Kitsune	doi:10.14722/ndss.2018.23204	UCI ML
LTDB	doi:10.1161/01.CIR.101.23.e215	PhysioNet
Metro	doi:10.1109/I2MTC.2015.7151267	UCI ML
MITDB	doi:10.1161/01.CIR.101.23.e215	PhysioNet
NAB	doi:10.1016/j.neucom.2017.04.070	Benchmark Website
NASA-MSL	doi:10.1145/3219819.3219845	Paper Website
NASA-SMAP	doi:10.1145/3219819.3219845	Paper Website
NormA	doi:10.1145/3055366.3055375	Paper Website
Occupancy	doi:10.1016/j.enbuild.2015.11.071	UCI ML
Opportunity	doi:10.1109/INSS.2010.5573462	UCI ML
SMD	doi:10.1145/3292500.3330672	Paper Website
TSAD 2021	doi:10.1109/TKDE.2021.3112126	UCR TSCA
Yahoo! WS	doi:LATS	Yahoo! Webscope

(Bhattacharya and Burman, 2016), from which the following definitions are also adopted.

- **Autoregressive coefficients** quantify the relationship between a time series and its lagged values, providing insight into the temporal dependencies within the data.

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \varepsilon_t \quad (4)$$

  X_t is the value of the time series at time t, c is a constant term, $\phi_1, \phi_2, \ldots, \phi_p$ are the autoregressive coefficients. $X_{t-1}, X_{t-2}, \ldots, X_{t-p}$ are the lagged values of the time series and ε_t is the error term at time t.
- **Cross-correlation** measures the similarity between two time series as a function of the lag of one relative to the other, indicating potential relationships or patterns shared between them.

$$R_{xy}(\tau) = \frac{\sum_{t=1}^{N-\tau}(x_t - \mu_x)(y_{t+\tau} - \mu_y)}{\sqrt{\sum_{t=1}^{N}(x_t - \mu_x)^2 \sum_{t=1}^{N}(y_t - \mu_y)^2}} \quad (5)$$

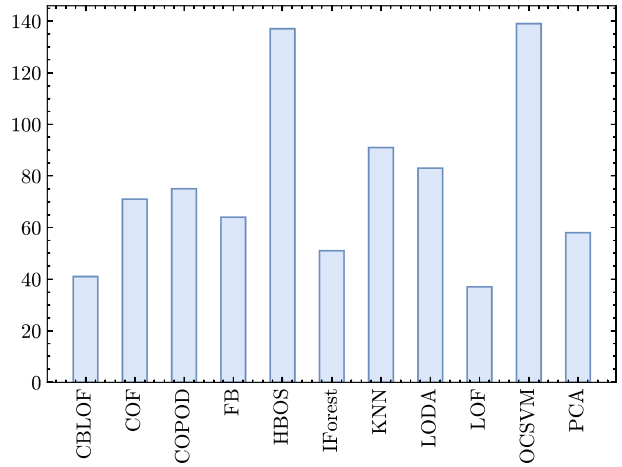
  where $R_{xy}(\tau)$ is the cross-correlation coefficient at lag τ, x_t and y_t are the values of the two time series at time t, μ_x and μ_y are the means of the two time series, respectively, τ is the lag parameter, and N is the total number of data points in the time series. The cross-correlation coefficient $R_{xy}(\tau)$ ranges from -1 to 1, where 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 indicates no correlation.
- **Covariance** quantifies the extent to which two time series X and Y change together, providing information about their joint variability.

$$\mathrm{Cov}(X, Y) = \frac{\sum_{i=1}^{n}(x_i - \mu_X)(y_i - \mu_Y)}{n - 1} \quad (6)$$

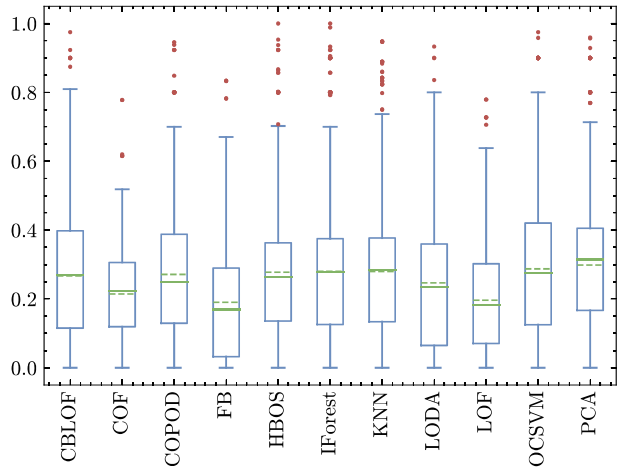
  where $\mathrm{Cov}(X, Y)$ is the covariance between X and Y, x_i and y_i are individual observations of X and Y, μ_X and μ_Y are the means of X and Y respectively, and n is the number of observations. A positive covariance indicates a positive linear relationship, while a negative covariance indicates a negative linear relationship. A covariance close to zero indicates a weak or no linear relationship.
- **Transfer entropy** quantifies the directional flow of information from one time series to another, revealing the extent of information transfer between them. Given two discrete-time processes X and Y, the transfer entropy from X to Y at lag k is defined as:

$$TE_{X\to Y}(k) = H(Y_{t+1}|Y_t^{(k)}, X_t^{(k)}) - H(Y_{t+1}|Y_t^{(k)}) \quad (7)$$

  where $TE_{X\to Y}(k)$ is the transfer entropy from X to Y at lag k, $H(\cdot)$ denotes the conditional entropy, Y_{t+1} is the future value of process Y, and $Y_t^{(k)}$ and $X_t^{(k)}$ are the histories of Y and X up to lag k respectively. The first



(a) Top performing algorithms as measured by F1-Score for all datasets.



(b) Distribution of prediction scores (F1-Score) for each candidate algorithm across all training data sets

Fig. 7 Verification of the suitability of the data set collections for the evaluation of the meta-learning procedure

term on the right-hand side measures the uncertainty in the future of Y given both the past of Y and the past of X up to lag k. The second term measures the uncertainty in the future of Y given only the past of Y up to lag k. The difference between these two terms quantifies the additional information provided by the past of X up to lag k.

- **Maximum power frequencies** represent the dominant oscillatory patterns within the time series, providing insight into periodic behavior or trends.Given a continuous signal $x(t)$, the power spectral density (PSD) $S_x(f)$ represents the distribution of power as a function of frequency. The MPF is defined as:

$$f_{\text{MPF}} = \arg\max_f S_x(f) \tag{8}$$

where f_{MPF} is the Maximum Power Frequency, and $S_x(f)$ is the power spectral density of the signal.

Besides the above described metrics to describe time series, it seems reasonable to expect that the specific nature of anomalies present in a dataset can significantly affect the effectiveness of a chosen algorithm. Therefore, we introduce meta-features that provide additional details about the nature of anomalies. It is important to note, however, that these meta-features can only be generated if the training dataset contains labels that categorize the state of the data at different points in time. In the following, x_i represents the value of the time series at time i, where $x_i = 1$ indicates an anomalous data point and $x_i = 0$ indicates a normal data point. N is the total number of data points in the time series.

- **Total anomaly ratio** indicates the proportion of anomalous data points relative to the total length of the time series, providing a measure of the overall prevalence of anomalies.

$$TAR = \frac{\sum_{i=1}^{N} x_i}{N} \tag{9}$$

- **Point anomaly ratio** represents the ratio of individual data points identified as anomalies, providing insight into the frequency of isolated abnormal observations.

$$PAR = \frac{\sum_{i=1}^{N} x_i}{N} \tag{10}$$

- **Collective anomaly ratio** quantifies the ratio of anomalies that occur collectively, indicating instances where anomalies occur in groups or clusters within the time series.

$$CAR = \frac{\sum_{i=1}^{N} x_i}{N} \tag{11}$$

- **Anomaly durations** refer to the lengths of time anomalies persist in the time series, providing information about the temporal extent of abnormal patterns. To calculate Anomaly Durations, we identify consecutive sequences of anomalous data points and measure the duration of each such sequence. For example, if $x_i = 1$ indicates an anomalous data point and $x_i = 0$ indicates a normal data point, the duration of an anomaly sequence starting at time i and ending at time j would be $j - i + 1$ time units.
- **Non-anomaly durations** represent the length of time intervals between anomalous occurrences, providing insight into the periods of normalcy or absence of anomalies in the time series. To calculate Non-Anomaly Durations, we identify the consecutive sequences of normal data points (0s) and measure the duration of each such

sequence. For example, if $x_i = 1$ indicates an anomalous data point and $x_i = 0$ indicates a normal data point, the duration of a non-anomaly sequence starting at time i and ending at time j would be $j - i + 1$ time units.

Additionaly, further sets of meta-features (Table 3) were extracted using the PyMFE library (Alcobaça et al., 2020a).

Experimental evaluation

The automotive manufacturing process encompasses a spectrum of highly automated welding procedures. For instance, clusters of robots are employed to weld various types of studs onto the car body to facilitate subsequent assembly processes. In this context, substantial volumes of data are generated and gathered to monitor and control robots and equipment. The studs' material, length, diameter, and welding position vary depending on the car model. Consequently, different welding programs are executed. Depending on the production schedule, model changes may occur more or less frequently, resulting in a broad array of process parameter curves. Figure 8 illustrates the same two signals for different combinations of studs and bodies, each identified by a unique ProcessID.

When it comes to implementing PdM using machine learning techniques, the critical question is: Which algorithm is optimal for each process? What if there's a need to establish fault prediction for numerous types of robots? What if additional processes and technologies are introduced, each with different variants and components? How does the monitoring adapt when the workload on a machine changes? And how does replacing a machine component affect the accuracy the prognostic model? What happens to sensor readings as environmental conditions change? The crux of the matter is that any change in these conditions may require an adjustment or modification to the algorithm used to monitor the process. In such scenarios, an automated approach to selecting appropriate algorithms would save time and resources.

In the following experiments, we investigate whether a reliable selection of algorithms is possible with the presented procedure. The performance is compared with benchmark methods, the importance of the individual meta-features is examined, and the computing times of the candidate algorithms are considered.

Meta-algorithm configuration

In a preliminary study, several meta-learning algorithms were evaluated in conjunction with different feature encoding methods using fivefold cross-validation based on the MSE. In particular, the Random Forest Regressor (RFR) algorithm showed robust performance and was therefore chosen as the meta-learning algorithm. It works by generating numerous decision trees during the training phase, each constructed using a random subset of the training data and features to ensure diversity among the trees. During the prediction phase, RFR combines the outputs of individual trees to produce the final prediction. We use the sklearn.multioutput module, which extends the capabilities of this originally single-output regression model to handle multiple target variables simultaneously. The problem involves learning a mapping function f that predicts F1-Scores (Y') based on a matrix X of meta-features. The objective is to minimize the Mean Squared Error loss function, $L(Y, Y')$, where Y represents the actual F1-Scores. The RFR is trained to optimize its parameters, resulting in a model f^* used for predicting F1-Scores for new meta-features. Considering a dataset collection with N rows, each dataset is characterized by M meta-features. Let X represent the matrix of meta-features ($N \times M$), where X_{ij} is the j-th meta-feature for the i-th dataset. The target Y, is a matrix of the actual F1-Scores ($N \times 1$), where Y_i is the F1-Score for the i-th algorithm.

$$L(Y, Y') = \frac{1}{DN} \sum_{i=1}^{D} \sum_{j=1}^{N} (Y_{ij} - Y'_{ij})^2 \tag{12}$$

The optimal regression model f^* minimizes this loss:

$$f^* = \arg\min_f L(Y, f(X)) \tag{13}$$

The resulting model f^* serves for predicting F1-Scores for each candidate algorithm for the test datasets which are not known to the meta-model. To evaluate the Meta-M-Sel method, the parameters were set to $\sigma_l = 0.2$ and $\sigma_u = 1$ using a grid search.

Benchmarks

The literature lacks a standardized evaluation procedure for meta-learning approaches, leading to a diversity of evaluation methodologies without established guidelines. To fill this gap, this study employs a comparative analysis by introducing five benchmark methods for evaluating the selection strategies. This approach aims to provide an informative assessment of the meta-model's performance relative to the following baselines:

- **Optimum:** Selection of the candidate algorithm with the best performance on the test data, representing the ideal selection strategy. It is assumed that all results are known and the best one for a data set is used to calculate the F1-Score.

Table 3 Overview of meta feature groups extracted from PyMFE

Group	Description
Clustering	Various indices and metrics related to clustering quality, such as INT index, normalized relative entropy, Pearson correlation between class matching and instance distances, number of clusters with size smaller than a given size, mean silhouette value, Davies and Bouldin Index, and Dunn Index
Complexity	Measures related to the complexity of the dataset, including entropy of class proportions, imbalance ratio, clustering coefficient, average density of the network, Fisher's discriminant ratios, volume of overlapping region, feature efficiency, hub score, error distances, OVO subsets error rate of linear classifier, non-linearity of a linear classifier, cardinality, fraction of borderline points, nearest neighbor distances, and more
Concept	Features related to the variability and density of class labels among examples, including variations of weighted distance and concept variation
General	Features providing general information about the dataset, such as ratios between attributes, categorical and numeric features, relative frequency of each distinct class, instances and attributes, total number of attributes, number of binary and categorical attributes, distinct classes, instances (rows), numeric features, and numerical and categorical features
Info-theory	Information theory related measures including concentration coefficients, Shannon's entropy for predictive attributes, mutual information, and other attributes' entropy, noisiness of attributes, and number of equivalent attributes for a predictive task
Itemset	Features related to itemset mining, including one itemset and two itemset meta-features
Landmarking	Performance measures of different classifiers and models, such as decision trees, nearest neighbor classifiers, linear discriminant classifier, and Naive Bayes classifier
Model-based	Features derived from decision tree models, including the number of nodes (leaf and non-leaf), size of branches, leaves corroboration, homogeneity, proportion of leaves per class, tree shape, depth of nodes, features importance, and various ratios
Statistical	A wide range of statistical measures such as correlations, covariances, eigenvalues, means, interquartile range, kurtosis, skewness, sparsity, variance, and various tests and traces for statistical properties of the dataset. This group encompasses many different statistical aspects of the data

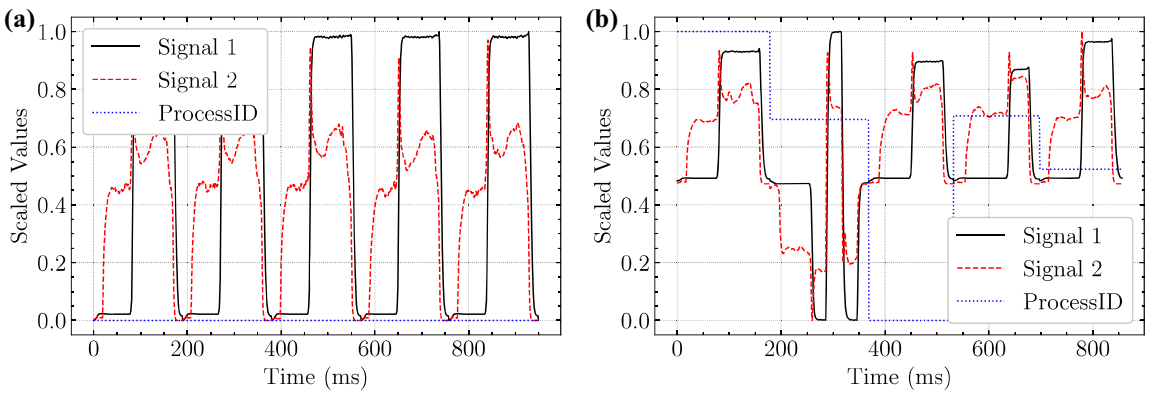


Fig. 8 Curves of process parameters in stud welding. **a** Process parameters for identical stud and car body combination (ProcessID). **b** Variation in process parameters across different combinations of studs and car bodies

- **Te-Sel:** Selection of the algorithm with the best overall prediction result on the test data. Again, it is assumed that all outcomes are known. Instead of using an individual algorithm for each new data set, the algorithm with the best overall result for all data sets is used to determine the F1-Score.
- **Tr1-Sel:** Selects the algorithm with the best performance on the training data. Here, the results on the training data are used as the selection function. The algorithm with the best result on a subset of the data is also used to predict the evaluation data.
- **Tr2-Sel:** Forms an ensemble of the two most effective algorithms on the training data. In contrast to Tr1-Sel, not a single, but a combination of the results of the two best algorithms on a part of the data is used for the evaluation part of the data.
- **Tr3-Sel:** Forms an ensemble of the three best performing algorithms on the training data. Like Tr2-Sel, just a combination of the three best algorithms.

The result of the formed ensembles is determined by simple mean value calculation. The results of predicting whether or not an anomaly is present are added and divided by the number of candidates in the ensemble. The F1-Score is then calculated for each data set.

Prognostic performance

Figure 9 provides a comparative overview of the performances of the proposed meta-models and benchmark methods. It displays F1-Score predictions for previously unknown test datasets, simulating the productive use of the offline-constructed meta-models. The meta-model must make selection decisions for a single algorithm (Meta-S-Sel) or form an ensemble (Meta-M-Sel) based on incoming data set meta-features. Additional approaches include Te-1-Sel (application of the best overall algorithm on the test data), Tr-1-Sel (selection of the overall best algorithm for training data, then applied to test data), Tr-2-Sel (ensemble of the best two algorithms for training data), and Tr-3-Sel (ensemble of the best three algorithms for training data). The box plots are ordered by medians, with the optimal outcome, achievable by individually selecting the single best algorithm from the candidate set for each dataset, positioned at the far right (Optimum). Corresponding numerical values are available in Table 4, sorted by median in ascending order.

The findings suggest that when it comes to selecting a single algorithm for an unknown dataset, Meta-S-Sel demonstrates superior performance. Following closely, Meta-M-Sel performs well, while Te-1-Sel, Tr-2-Sel, Tr-3-Sel, and Tr-1-Sel exhibit significantly lower performances. The median difference between Meta-S-Sel and the optimal value is minimal (0.023), and the maximum value matches the optimum, highlighting its efficiency. In contrast, the discrepancies with the optimum are more substantial for Meta-M-Sel (0.082), Te-1-Sel (0.144), Tr-2-Sel (0.148), Tr-3-Sel (0.148), and Tr-1-Sel (0.166).

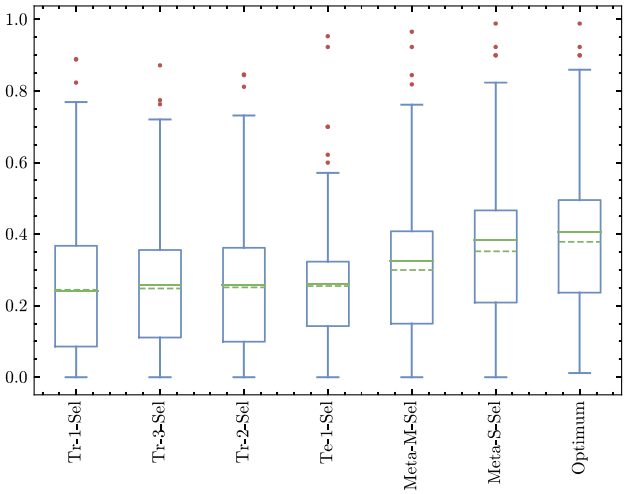


Fig. 9 Box plots illustrating predictive performance for baseline and meta-selection strategies

In summary, the results strongly confirm the effectiveness of the meta-learning approach in selecting the most appropriate algorithm or ensemble based on meta-data. The method even outperforms all strategies that rely on evaluations of parts of considered data sets, demonstrating its superior performance. Of particular note is its ability to outperform the selection of the overall best algorithm, the most common approach in industrial settings. The minimal difference from the optimum indicates that the chosen meta-features encapsulate indispensable information that is critical for making and informed selection decisions. Regarding the superiority of Meta-S-Sel over Meta-M-Sel, it is suspected that this is due to the fact that the meta model can clearly identify the best suitable algorithm for a data set. Thus, model selection yields better results than a combination of the best and the second-best candidate algorithms. However, there is always the possibility of overfitting the - even if large - selection of datasets. In this sense, the Meta-M-Sel may be a more robust solution for model selection in practice if the data structures are very different from those used here.

Meta-feature importance

Examining the importance of different meta-features for predicting the performance of individual candidate algorithms through the meta-model provides deep insights into the effectiveness of the proposed approach. In total, 155 meta-features were obtained for each dataset, 145 of which were computed from the PyMFE collection, apart from the ten meta-features

Table 4 Statistical summary of predictive performances for baseline and meta-selection strategies

	Mean	SD	Min	25%	50%	75%	Max
Tr-1-Sel	0.244	0.196	0.000	0.086	0.240	0.368	0.889
Tr-3-Sel	0.248	0.176	0.000	0.111	0.258	0.356	0.872
Tr-2-Sel	0.251	0.186	0.000	0.099	0.258	0.362	0.846
Te-1-Sel	0.255	0.189	0.000	0.143	0.262	0.323	0.953
Meta-M-Sel	0.300	0.204	0.000	0.150	0.324	0.408	0.966
Meta-S-Sel	0.352	0.211	0.000	0.209	0.383	0.466	0.989
Optimum	0.378	0.213	0.011	0.236	0.406	0.495	0.989

presented in this paper (see Sect. 3.4). Permuation Feature Importance was used as the method to determine the results. This is an analysis method that measures the contribution of each feature to the statistical performance of an adapted model. To do this, the values of each feature are randomly shuffled and the change in model performance is observed (Breiman, 2001). In this way, it can be used to analyze how much a feature contributes to the prediction of a target variable.

Table 5 ranks the results in descending order of median, with Transfer Entropy Maximum and Mutual Information Mean emerging as the most important features by a significant margin. Interestingly, the Collective Anomaly Ratio is also among the top features, indicating that the collective anomaly ratio is a critical criterion for selecting anomaly detection algorithms. This confirms the hypothesis that the newly introduced meta-features help to describe the similarity of data sets and capture relevant criteria for the selection of anomaly detection algorithms. The results suggest that the relationships between time series provide valuable insights, justifying the use of meta-features, in particular Transfer Entropy Maximum and Mutual Information Mean. These features allow the meta-model to identify relevant patterns and structures in the data, facilitating informed decision making. However, it is important to note that the availability of labels, especially for the Collective Anomaly Ratio, may not be guaranteed in all applications. Therefore, there may be situations where certain features cannot be used due to a lack of label information. Furthermore, it should be emphasized that the relevance of features is highly dependent on the specific circumstances and requirements of the application domain. What is meaningful in one context may be less relevant in another. In summary, the results of the feature importance analysis confirm the effectiveness of the newly introduced meta-features and underline their role in describing data and capturing crucial criteria for the selection of anomaly detection algorithms. These findings are essential for adapting the meta-model to specific use cases and optimizing its performance.

Computing times

The computational time of various candidate anomaly detection algorithms is examined by measuring the time each algorithm takes to predict each data point in a dataset. The results show clear differences in the computation times of the models considered. HBOS, PCA, and CBLOF proved to be significantly faster algorithms than LODA, COPOD, LOF, and OCSVM. Conversely, IFOREST, KNN, FB, and COF proved to be relatively slow. The slowest group not only showed generally longer computation times, but also a remarkable variation in the measured values. This variation could indicate an increased sensitivity to data characteristics, especially for complex datasets with many attributes and instances. Observing these differences in computation time raises questions about the practical applicability of the algorithms. The faster models, such as HBOS and PCA, could be advantageous in applications that require real-time response to anomalies. The longer computational times of the slower algorithms could be problematic in certain contexts, particularly where quick decisions are required. The high variance within this group indicates sensitivity to different data structures, leading to significant performance variations in complex scenarios. It is important to note that the proposed method does not use computational time as a criterion for algorithm selection, but focuses solely on predictive performance. In practice, however, the choice of an anomaly detection algorithm should be carefully considered, taking into account the specific requirements of the domain.

Conclusion

In this paper, a meta-learning approach to algorithm selection was presented to predict the predictive performance of different candidate algorithms for anomaly detection in manufacturing systems. Subsequently, the best algorithm is selected or a dynamic ensemble is formed based on the estimated predictive performance. Experiments evaluating the method demonstrate its superiority over five benchmark approaches. In practice, even when algorithms are selected

Table 5 Importance of meta-features for algorithm selection

Meta-feature	Importance	Group
Transfer Entropy Maximum	0.28258	Multivariate Time Series
Mutual Information Mean	0.22014	PyMFE
Collective Anomaly Ratio	0.03380	Anomaly Feature
Naive Bayes Standard Deviation	0.02897	PyMFE
Nodes Repeated Standard Deviation	0.02755	PyMFE
Transfer Entropy Mean	0.02580	Multivariate Time Series
W Lambda	0.02370	PyMFE
Joint Entropy Mean	0.02354	PyMFE
Tree Shape Mean	0.02126	PyMFE
Transfer Entropy Standard Deviation	0.01919	Multivariate Time Series
Max Power Frequency Mean	0.01768	Multivariate Time Series
Cross Correlation Standard Deviation	0.01578	Multivariate Time Series
LH Trace	0.01449	PyMFE
Roy Root	0.01378	PyMFE
Cross Correlation Mean	0.01353	Multivariate Time Series
P Trace	0.01347	PyMFE
Class Entropy	0.01200	PyMFE
Leaves Homogeneity Standard Deviation	0.01082	PyMFE
Elite Nearest Neighbor Mean	0.01076	PyMFE
Max Power Frequency Maximum	0.00921	Multivariate Time Series
Best Node Standard Deviation	0.00879	PyMFE
Autoregressive Standard Deviation	0.00854	Multivariate Time Series

Table 6 Statistical summary of prediction times for candidate algorithms across all datasets

	mean	std	min	25%	50%	75%	max
HBOS	0.000	0.000	0.000	0.000	0.000	0.001	0.003
PCA	0.001	0.001	0.000	0.000	0.000	0.002	0.004
CBLOF	0.001	0.001	0.000	0.001	0.001	0.001	0.014
LODA	0.007	0.007	0.001	0.002	0.003	0.008	0.078
COPOD	0.007	0.006	0.001	0.002	0.005	0.011	0.038
LOF	0.026	0.037	0.001	0.004	0.015	0.033	0.545
OCSVM	0.169	0.363	0.000	0.003	0.023	0.084	3.028
IForest	0.049	0.026	0.026	0.030	0.041	0.052	0.192
KNN	0.136	0.156	0.004	0.020	0.067	0.200	1.441
FeatureBagging	0.221	0.321	0.005	0.018	0.099	0.319	4.770

individually based on validation errors, the meta-learning approach improves predictive performance. Moreover, the success of the method lies in its ability to make a better selection than choosing the overall best algorithm for the test data. The minimal differences between the meta-learning method and the optimal strategy suggest that the meta-features are well suited to describe multivariate datasets and capture relevant properties for anomaly detection.

However, these promising results should be treated with caution. Although efforts were made to diversify the data sets, the generalizability of the method cannot be assumed. Such generalizability would require further investigation in different application domains. In addition, it is unclear how improved anomaly detection affects the manufacturing system, especially with respect to the optimization of the algorithmic evaluation metrics. In this paper, different loss functions are used in different places. First, the individual loss functions of the candidate algorithms, and second, the loss function of the meta-regression model, where the mean square error (MSE) is used to evaluate the performance of the meta-model in estimating the candidate algorithms. In addition, the F1-Score is used as a criterion for comparing different candidates and selection methods to evaluate how well anomalies can be identified. With respect to the individ-

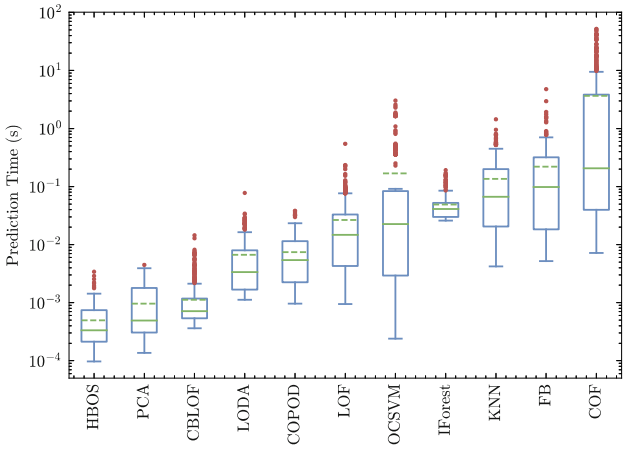


Fig. 10 Boxplot of the distribution of prediction times for each candidate algorithm across all training data sets

ual loss functions of the candidates, it can be assumed that better performance leads to better anomaly detection. Better anomaly detection can in turn lead to better failure prevention and thus to a higher overall effectiveness of the manufacturing system. This correlation exists in general, but the strength of the correlation depends on the structure of the manufacturing system. More research is needed to make more accurate statements. Overall, under the conditions of this study, the initial hypothesis is confirmed: the individual selection of robust algorithms for anomaly detection leads to a better predictive performance for multivariate data sets, such as those found in machine and operational data in manufacturing. The automated applicability of the method provides an opportunity to increase the efficiency of manufacturing systems by minimizing unplanned downtime through condition-based maintenance and enabling efficient maintenance planning.

Acknowledgements This work was supported by the German Research Foundation (DFG) under grant number FR 3658/4-1 as part of the Collaborative Research Initiative on Smart Connected Manufacturing.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability The data that support the findings of this study are openly available. The repositories are listed including the DOI and Download Source in Table 2.

Declarations

Competing interest The authors declare that they have no known competing financial interests or personal relationships that could appear to have influenced the work reported in this paper.

Declaration of generative AI and AI-assisted technologies in the writing process During the preparation of this work the authors used generative AI in order to improve language and readability. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

Aggarwal, C. C. (2017). *Outlier analysis*. Springer International Publishing. https://doi.org/10.1007/978-3-319-47578-3

Alcobaça, E., Siqueira, F., Rivolli, A., Garcia, L. P. F., Oliva, J. T., & de Carvalho, A. C. P. L. F. (2020). Mfe: Meta-feature extraction. *Journal of Machine Learning Research, 21*(111), 1–5.

Alcobaça, E., Siqueira, F., Rivolli, A., Garcia, L. P. F., Oliva, J. T., & de Carvalho, A. C. P. L. F. (2020). Mfe: Towards reproducible meta-feature extraction. *Journal of Machine Learning Research, 21*(111), 1–5.

Ali, A. R., Gabrys, B., & Budka, M. (2018). Cross-domain meta-learning for time-series forecasting. *Procedia Computer Science, 126*, 9–18. https://doi.org/10.1016/j.procS.2018.07.204

Bahri, M., Salutari, F., Putina, A., & Sozio, M. (2022). Automl: State of the art with a focus on anomaly detection, challenges, and research directions. *International Journal of Data Science and Analytics, 14*(2), 113–126. https://doi.org/10.1007/s41060-022-00309-0

Bart, E., & Ullman, S. (2005). Cross-generalization: Learning novel classes from a single example by feature replacement. In*2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, *1*, 672–679 vol. 1. https://doi.org/10.1109/CVPR.2005.117

Baxter, J. (2019). Learning internal representations. *CoRR, 8*, 311–320.

Bengio, Y. (2012). Deep learning of representations for unsupervised and transfer learning. In I. Guyon, G. Dror, V. Lemaire, G. Taylor & D. Silver (Eds.), *Proceedings of icml workshop on unsupervised and transfer learning* (pp. 17–36, Vol. 27). PMLR.

Bensusan, H., & Kalousis, A. (2001). Estimating the predictive accuracy of a classifier. In L. De Raedt & P. Flach (Eds.), *Machine learning: Ecml 2001* (pp. 25–36). Springer.

Bhattacharya, P., & Burman, P. (2016). 13 - time series. In P. Bhattacharya & P. Burman (Eds.), *Theory and methods of statistics* (pp. 431–489). Academic Press. https://doi.org/10.1016/B978-0-12-802440-9.00013-8

Brazdil, P., & da Costa, J. P. (2003). Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results. *Machine Learning, 50*, 251–277.

Breiman, L. (2001). Random forests. *Machine Learning, 45*, 5–32.

Cacoveanu, S., Vidrighin, C., & Potolea, R. Evolutional meta-learning framework for automatic classifier selection. In 2009, pp 27–30. https://doi.org/10.1109/ICCP.2009.5284790

Caruana, R. (1994). Learning many related tasks at the same time with backpropagation. In *Proceedings of the 7th International Conference on Neural Information Processing Systems*, pp. 657–664.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection. *ACM Computing Surveys, 41*(3), 1–58. https://doi.org/10.1145/1541880.1541882

Colledani, M., & Tolio, T. (2012). Integrated quality, production logistics and maintenance analysis of multi-stage asynchronous

manufacturing systems with degrading machines. *CIRP Annals, 61*(1), 455–458.

Cook, A. A., Misirli, G., & Fan, Z. (2020). Anomaly detection for iot time-series data: A survey. *IEEE Internet of Things Journal, 7*(7), 6481–6494. https://doi.org/10.1109/JIOT.2019.2958185

Fei-Fei, L., Fergus, R., & Perona, P. (2006). One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*(4), 594–611. https://doi.org/10.1109/TPAMI.2006.79

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems.* (Vol. 28). Curran Associates, Inc. Curran Associates, Inc.

Fink, M. (2004). Object classification from a single example utilizing class relevance metrics. In L. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in neural information processing systems.* (Vol. 17). MIT Press.

Fu, S., Gao, X., Zhang, H., Liu, M., Li, J., & Xu, J. (2022). Anomaly detection for power dispatching data based on meta-learning dynamic ensemble selection. *Dianwang Jishu/Power System Technology, 46*(8), 3248–3256. https://doi.org/10.13335/j.1000-3673.pst.2022.0017

Fürnkranz, J., & Petrak, J. (2001). An evaluation of landmarking variants.

Fusi, N., Sheth, R., & Elibol, M. (2017). Probabilistic matrix factorization for automated machine learning. *Neural Information Processing Systems, 1*, 3.

Gabbay, D. M., Siekmann, J., Brazdil, P., Giraud-Carrier, C., Soares, C., & Vilalta, R. (2009). *Metalearning*. Springer. https://doi.org/10.1007/978-3-540-73263-1

Gomes, T. A., Prudêncio, R. B., Soares, C., Rossi, A. L., & Carvalho, A. (2012). Combining meta-learning and search techniques to select parameters for support vector machines. *Neurocomputing, 75*(1), 3–13. https://doi.org/10.1016/j.neucom.2011.07.005

Hawkins, D. M. (1980). *Identification of outliers*. Springer. https://doi.org/10.1007/978-94-015-3994-4

Hutter, F., Kotthoff, L., & Vanschoren, J. (2019). *Automated machine learning*. Springer International Publishing. https://doi.org/10.1007/978-3-030-05318-5

Karmaker, S. K., Hassan, M. M., Smith, M. J., Xu, L., Zhai, C., & Veeramachaneni, K. (2022). Automl to date and beyond: Challenges and opportunities. *ACM Computing Surveys, 54*(8), 1–36. https://doi.org/10.1145/3470918

Köpf, C., Taylor, C., Ag, D. *& box, P* (2000). From data characterisation for meta-learning to meta-regression: Meta-analysis.

Kotlar, M., Punt, M., Radivojevic, Z., Cvetanovic, M., & Milutinovic, V. (2021). Novel meta-features for automated machine learning model selection in anomaly detection. *IEEE Access, 9*, 89675–89687. https://doi.org/10.1109/ACCESS.2021.3090936

Leite, R., & Brazdil, P. (2005). Predicting relative performance of classifiers from samples, pp. 497–503. https://doi.org/10.1145/1102351.1102414

Lemke, C., Budka, M., & Gabrys, B. (2015). Metalearning: A survey of trends and technologies. *Artificial intelligence review, 44*(1), 117–130. https://doi.org/10.1007/s10462-013-9406-y

Lemke, C., & Gabrys, B. (2010). Meta-learning for time series forecasting and forecast combination. *Neurocomputing, 73*(10–12), 2006–2016. https://doi.org/10.1016/j.neucom.2009.09.020

Li, L., Wang, Y., Xu, Y., & Lin, K.-Y. (2022). Meta-learning based industrial intelligence of feature nearest algorithm selection framework for classification problems. *Journal of Manufacturing Systems, 62*(4), 767–776. https://doi.org/10.1016/j.jmsy.2021.03.007

Lopez, F., Saez, M., Shao, Y., Balta, E. C., Moyne, J., Mao, Z. M., Barton, K., & Tilbury, D. (2017). Categorization of anomalies in smart manufacturing systems to support the selection of detection mechanisms. *IEEE Robotics and Automation Letters, 2*(4), 1885–1892. https://doi.org/10.1109/LRA.2017.2714135

Lughofer, E., & Sayed-Mouchaweh, M. (2019). *Predictive maintenance in dynamic systems*. Springer International Publishing. https://doi.org/10.1007/978-3-030-05645-2

Maforte Dos Santos, P., Ludermir, T. B., & Prudêncio, R. B. C. Selection of time series forecasting models based on performance information. In: 2005, pp. 366–371.

Mantovani, R. (2018). *Use of meta-learning for hyperparameter tuning of classification problems* [Doctoral dissertation].

Papastefanopoulos, V., Linardato, P., & Kotsiantis, S. (2021). Unsupervised outlier detection: A meta-learning algorithm based on feature selection. *Electronics (Switzerland), 18*, 10. https://doi.org/10.3390/electronics10182236

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Poulakis, Y., Doulkeridis, C., & Kyriazis, D. Autoclust: A framework for automated clustering based on cluster validity indices. In: *2020-November*. 2020, 1220–1225. https://doi.org/10.1109/ICDM50108.2020.00153

Reif, M., Shafait, F., Goldstein, M., Breuel, T., & Dengel, A. (2014). Automatic classifier selection for non-experts. *Pattern Analysis and Applications, 17*(1), 83–96. https://doi.org/10.1007/s10044-012-0280-z

Ridd, P., & Giraud-Carrier, C. (2014). Using metalearning to predict when parameter optimization is likely to improve classification accuracy. In *Proceedings of the 2014 International Conference on Meta-Learning and Algorithm Selection-Volume 1201*, pp. 18–23.

Sanders, S., & Giraud-Carrier, C. (2017). Informing the use of hyperparameter optimization through metalearning. *IEEE International Conference on Data Mining (ICDM), 2017*, 1051–1056. https://doi.org/10.1109/ICDM.2017.137

Schmidl, S., Wenig, P., & Papenbrock, T. (2022). Anomaly detection in time series. *Proceedings of the VLDB Endowment, 15*(9), 1779–1797. https://doi.org/10.14778/3538598.3538602

Schneider, P., & Xhafa, F. (2022). Chapter 3-anomaly detection: Concepts and methods. In P. Schneider & F. Xhafa (Eds.), *Anomaly detection and complex event processing over iot data streams* (pp. 49–66). Academic Press. https://doi.org/10.1016/B978-0-12-823818-9.00013-4

Smith-Miles, K. A. (2008). Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys, 41*(1), 1–25. https://doi.org/10.1145/1456650.1456656

Sun, Q., Pfahringer, B., & Mayo, M. (2013). Towards a framework for designing full model selection and optimization systems. *International Workshop on Multiple Classifier Systems*, pp. 259–270.

Tao, Fei, Qi, Qinglin, Liu, Ang, & Kusiak, Andrew. (2018). Data-driven smart manufacturing. *Journal of Manufacturing Systems, 48*, 157–169. https://doi.org/10.1016/j.jmsy.2018.01.006

Tavares, G. M., & Junior, S. B. (2021). Process mining encoding via meta-learning for an enhanced anomaly detection. *Communications in Computer and Information Science, 1450*, 157–168. https://doi.org/10.1007/978-3-030-85082-1_15

Thrun, S., & Mitchell, T. Learning one more thing. In *Proceedings of 14th international joint conference on artificial intelligence (ijcai '95)*. Morgan Kaufmann, 1995, pp. 1217–1223.

Vanschoren, J. (2010). Understanding machine learning performance with experiment databases (het verwerven van inzichten in leerperformantie met experiment databanken).

Vanschoren, J. (2018). Meta-learning: A survey.

Vichare, N., Rodgers, P., Eveloy, V., & Pecht, M. G. (2004). In situ temperature measurement of a notebook computer: A case study in health and usage monitoring of electronics. *IEEE Transactions*

on Device and Materials Reliability, 4(4), 658–663. https://doi.org/10.1109/TDMR.2004.838403

Wang, Y., Perry, M., Whitlock, D., & Sutherland, J. W. (2022). Detecting anomalies in time series data from a manufacturing system using recurrent neural networks. *Journal of Manufacturing Systems, 62*, 823–834. https://doi.org/10.1016/j.jmsy.2020.12.007

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67–82. https://doi.org/10.1109/4235.585893

Yu, Y.-C., Chuang, S.-W., Shuai, H.-H., & Lee, C.-Y. Fast adaption for multi motor anomaly detection via meta learning and deep unsupervised learning. In *2022-June*. 2022, pp. 1186–1189. https://doi.org/10.1109/ISIE51582.2022.9831559

Zhao, Y., Rossi, R., & Akoglu, L. (2020). Automating outlier detection via meta-learning. *arXiv preprint* arXiv:2009.10606.

Zhao, Y., Nasrullah, Z., & Li, Z. (2019). Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research, 20*(96), 1–7.

Zhao, Y., Rossi, R. A., & Akoglu, L. (2021). Automatic unsupervised outlier model selection. *Advances in Neural Information Processing Systems, 6*, 4489–4502.

Zöller, M.-A., & Gabrys, B. (2020). Avatar-machine learning pipeline evaluation using surrogate model. *Advances in Intelligent Data Analysis XVIII: 18th International Symposium on Intelligent Data Analysis, IDA*, p. 352.