PREPRINT

Toward a Time Series-Specific Machine Learning Life Cycle: Challenges and Requirements

Christoph Schrade[1], Theo Zschörnig [2] and Bogdan Franczyk[1,2,3]
[1]Information Systems Institute, Leipzig University, Grimmaische Str. 12, 04109 Leipzig, Germany
[2]Institute for Applied Informatics (InfAI), Goerdelerring 9, 04109 Leipzig, Germany
[3]Business Informatics Institute, Wrocław University of Economics, ul. Komandorska 118-120, 53-345 Wrocław, Poland
schrade@wifa.uni-leipzig.de, zschoernig@infai.org, franczyk@wifa.uni-leipzig.de

Keywords: Time Series, Machine Learning Life Cycle, Data Processing

Abstract: Machine learning (ML) life cycles differ from conventional software development cycles because ML model behaviour depends on data, not only on code. However, the concrete form of an ML life cycle also depends on the data modality that drives applications. This article argues that time series data is distinctive and often particularly challenging. We compare key data handling activities, especially data collection and data processing, for time series with typical practices for text and image data. Based on these differences, we derive requirements for a time series specific ML life cycle. The paper therefore provides a first step toward a future reference model for developing and operating time series-based ML systems.

1 INTRODUCTION

The machine learning (ML) life cycle is usually defined as the end-to-end process within which an ML system is developed and operated; from problem definition, data preparation, model training and evaluation to deployment, monitoring and ongoing improvement (Schaffererin 2025). The use of data as a core resource for ML-based software applications brings with it special requirements that make this ML life cycle substantially different from the usual software development life cycle (Sculley et al. 2015; Silva and Alahakoon 2022). However, these requirements also depend on the type of data used. Typical types of data used in ML applications and found in the context of computer vision and language models are image and text data. Contrary to this, in many domains such as the Internet of Things (IoT) or energy industry, time series data, i.e., data indexed by timestamps, is of great importance (cf. Hu et al. 2023). It has been shown that specific aspects and difficulties arise when handling time series data in the context of ML (Conrad et al. 2024). These distinguish time series from text and image data, which are treated as prototypical data modalities in machine learning because a substantial share of AI research output is organized around natural language processing and visual computing (Maslej et al. 2023). Against this background, this paper aims to highlight these time series specifics in order to derive relevant requirements for the ML life cycle. The goal is to demonstrate that meeting these requirements is necessary to ensure a long-term robust performance of time series-based models. The requirements identified could later be used as a basis for developing a reference model for a time series specific ML life cycle.

The remainder of this paper is structured as follows: In Section 2, we present related work in the literature that deals with time series-related peculiarities for the ML life cycle. Section 3 will then describe specific challenges in handling time series data in the ML context. It will address the typical main phases of data preparation in the ML life cycle as well as other aspects that cannot be directly assigned to these phases. The special aspects of time series data handling will be presented in comparison to image and text data. In Section 4, we derive from these considerations what requirements should be placed on the ML life cycle when working with time series data. Finally, Section 5 summarizes our findings and describes future work.

2 RELATED WORK

The literature on the ML life cycle for time series data can be roughly divided into two different categories: On the one hand, there are works that consider the entire life cycle for specific application areas and focus primarily on automation potential. On the other hand, there is research that targets individual steps within the overall life cycle.

One source that provides an overview of the state of research in the development of automatic ML applications for time series forecasting is *Meisenbacher et al. 2022*. The authors in *ibid.* investigate all stages of the ML life cycle. However, only time series with equidistant time intervals between data points are considered from the outset, which, in our opinion, greatly simplifies data preprocessing, which is further elaborated in section 3. In addition, the authors describe feature engineering as a complex, highly application-specific problem, which is why they suggest further research in this area. *Conrad et al. 2024* test several AutoML frameworks for time series forecasting tasks in production engineering, but also generally note that "addressing the time series aspect in an automated ML (AutoML) framework poses unique challenges." As far as data handling is concerned, the authors state that automated feature engineering is not as high-quality as that performed by humans; further evidence of the use case-specific nature of feature engineering for time series.

Time series research on individual sub-steps within the ML life cycle primarily focuses on data preprocessing, feature engineering, and model development.

A recent paper that provides a good overview of preprocessing methods for time series is *Tawakuli et al. 2025*. However, again data with a fixed frequency is used, which, in our opinion, omits what is a very typical and difficult case in terms of preprocessing, namely irregularities with regard to the time axis.

Even though feature engineering for time series is a highly complex and use case-dependent task, programming libraries are proposed that implement frequently used feature families and transformation steps (Barandas et al. 2020; Cerqueira et al. 2024). They thus support partial standardization, but do not provide a universal solution for all domain-specific requirements.

There are also several papers that deal specifically with the topic of model selection and development, i. e. with the optimization of model architectures and hyperparameters, in the context of time series-driven ML software (Sergio et al. 2016; Rätz et al. 2019; Lowther et al. 2020; Deng et al. 2022). While these articles deal in detail with typical ML algorithms in the time series context, they have no direct influence on the higher-level structure of the ML life cycle, which is why they are not of primary importance for the present work.

The literature review conducted has shown that the consideration of the ML life cycle for time series-driven applications is an active field of research. However, it also became clear that difficulties in dealing with time series arise repeatedly, which are related to the very nature and structure of the data. The remainder of this article will therefore focus on these difficulties in order to derive general requirements for a time series specific ML life cycle. Such a holistic view of the life cycle with special requirements for time series has not yet been found in the literature.

3 SPECIFICS OF TIME SERIES DATA PROCESSING

In order to deduce requirements for the ML life cycle later on, this section discusses specific challenges in handling time series data in the ML context. To this end we consider process steps within the ML life cycle that are directly related to the data *before* it is transferred to a model - at this stage, we therefore exclude possible subsequent life cycle phases such as deployment and monitoring from the scope of analysis. The considered data related steps usually consist of *data collection* and following actions of data processing, such as *data preprocessing*, *data augmentation* and *feature engineering*. Across the literature, these steps are generally structured in a largely consistent manner, with variation occurring predominantly in the degree of granularity at which they are specified. For instance, *Shankar et al. 2024* distinguish data collection and data processing, which they subsume under the broader term of data preparation. *Mumuni and Mumuni 2025*, in turn, further decompose data processing into data preprocessing, data augmentation, and feature engineering.

In summary four steps, which – following *Shankar et al. 2024* - are subsumed under the term data preparation, will be explored:

P1. Data Collection
P2. Data Preprocessing
P3. Data Augmentation
P4. Feature Engineering

Table 1 compares main aspects within these phases between time series data and image and text data.

Next, we will take a closer look at the aspects that are only roughly outlined in the table.

3.1 Data Collection

Data collection in the ML context refers to identifying and acquiring relevant raw data from sources such as sensors, logs, databases, or external providers, including the definition of what is recorded and at which granularity. It establishes the dataset basis for subsequent processing.

A key difference between time series and image or text data concerns the existence of standardized training data sets. In the case of images and text, the field is strongly influenced by curated reference datasets and benchmarks, which enable uniform evaluation protocols and comparable performance measurements (Raji et al. 2021). Even if the data sets cannot be used to train models directly, pretraining is often possible (cf. e. g. Ridnik et al. 2021). Examples in the case of image data include the ImageNet dataset and the ILSVRC benchmarking practice (Russakovsky et al. 2015); for text data, there are very large open source corpora such as Common Crawl (Common Crawl - Open Repository of Web Crawl Data 2026). Such standards are often lacking for time series, and even when data sets exist, they are often domain-specific and difficult to transfer. In this regard the authors of *Qiu et al. 2024* explicitly state that “Time series from different domains may exhibit diverse characteristics” and that “insufficient coverage of data domains” in the existing data sets prevents fair benchmarks for model performance. In *Cui et al. 2024*, an attempt is made to address the aforementioned domain specificity by constructing a multi-domain dataset in which additional contextual information in the form of text data is added for each domain. The authors refer to the result as a multimodal multi-domain text-time-series dataset.

3.2 Data Preprocessing

Data preprocessing refers to the entirety of steps used to prepare raw data so that it can be reliably used for ML models. This should be clearly distinguished from data augmentation (that is discussed in the next subsection), in which the training data is specifically expanded or varied in order to make the model more robust and improve generalization.

One of the first steps in preprocessing time series is often to examine the temporal resolution of the

Table 1: Comparison of data preparation phases for ML models between image and time series data.

Phase	Image/Text Data	Time Series Data
P1 **Data Collection**	There are many large datasets available for (pre-)training models for various tasks.	Data usually needs to be collected on a domain-specific basis.
P2 **Data Preprocessing**	Data preprocessing usually follows a standard procedure.	Data preprocessing depends on the specific use case and the structure of the raw data.
P3 **Data Augmentation**	There are many obvious augmentation methods that follow geometric or semantic intuition.	Augmentation methods are usually very abstract.
P4 **Feature Engineering**	There are many generally applicable geometric or text-related features.	Definition of features depends heavily on the use case.

data. Irregular data frequencies and granularities may occur, i.e., the time intervals between transmitted data values may vary. This can occur, for example, due to different data frequency configurations between different data sources when multiple sources are used as input. However, this can also occur within individual data sources. A typical example of this in the context of IoT measurement infrastructures is device failures or the disruption of individual transmission paths within the infrastructure (Jeffery et al. 2020). In addition, configurations of the transmission frequency of measured values are common in which a value is only sent if it deviates sufficiently from the previous one, which also inevitably leads to irregularities in the time intervals.

However, many analyses and models require time series with a fixed data frequency as input. In these cases, fluctuating time intervals between the points in the raw data require resampling to a common time grid. This may require interpolation, whereby the chosen method can influence the resulting patterns in the data and thus also the model quality. The resampling and interpolation methods must be selected according to the specific application.

In the case of multiple data sources, it is also typical for values to be in different orders of magnitude. This makes robust statistical standardization mandatory, as different scales in the values lead to systematic distortions that are difficult to diagnose in later model operation. However, standardization (as well as similar methods as mean subtraction, scaling and normalization) are very common methods that are almost always used in preprocessing, regardless of which kind of data is utilized. Furthermore, this type of preprocessing always runs the same way and is therefore use case-independent. This is also the reason why preprocessing in the case of image data often follows a repetitive standard procedure, as standardization is primarily used (Goodfellow et al. 2016). Preprocessing text data is a much more complex task than for image data. Nevertheless, several methods have become established over time that can be applied repeatedly regardless of the context (Siino et al. 2024). The field of preprocessing for text data can therefore be considered much more standardized than for time series data.

3.3 Data Augmentation

As mentioned above, augmentation is the targeted expansion or variation of training data to achieve greater generalization of a model. Artificially modified training examples are generated, for example, by applying label preserving transformations or generating new plausible samples. This is typically a training-specific aspect, i.e. it is often only used in training, not in inference.

In many computer vision tasks, it is possible to augment the underlying image data used for training by applying simple geometric transformations such as rotations, translations, or reflections (Goodfellow et al. 2016). The geometric nature of the data alone therefore provides a whole range of obvious augmentation variants.

As with preprocessing, data augmentation for text data is often more complex than for image data, as text is discrete and small changes can alter the semantics. Nevertheless, there are numerous approaches that use text-specific transformations and are based on assumptions about semantic equivalence, which are often guided by human linguistic intuition (Li et al. 2022).

While augmentation methods derived from geometric or semantic intuition exist for image and text data, the field is more complicated for time series data. The overview article *Iglesias et al. 2023* essentially describes four different classes of approaches, all of which are rather abstract:

- Jittering (resp. adding noise to the samples)
- Approaches using dynamic time warping
- Approaches using variational auto encoders
- Approaches using Generative Adversarial Networks

The last three approaches in particular are highly complex, which is why implementing data augmentation in this way involves greater effort. Furthermore, it is more difficult to decide which augmentation method is suitable for which application. On the one hand, this is due to the abstract nature of the methods. On the other hand, it is again due to the already mentioned high domain specificity of time series. In this regard the authors of *ibid.* explicitly write: “Each dataset of time series is very different and needs special attention on how it is being augmented.”

3.4 Feature Engineering

Feature engineering is the transformation of raw or preprocessed data into informative input variables that a model can learn from. This is also a special aspect when handling time series as it is difficult to answer the questions of what constitutes meaningful features or what can be considered learnable structures within the data. For images and text, it is plausible to identify which local building blocks can carry information. In images, for example, edges, textures, color areas, or object parts can be understood as descriptive structures. In text, tokens and their sequences provide a natural basic unit, supplemented by syntactic and semantic regularities. Time series, on the other hand, are often less object-related and more relational, which makes feature engineering more abstract and less intuitive. Relevant information can be expressed in temporal patterns that are not tied to a single point in time, but rather to dynamics and transitions. What this relevant information looks like exactly can again depend heavily on the use case (Lubba et al. 2019). In this context, the choice of measurement scale and resolution can also play a role. Some patterns are only visible in coarse aggregation, others only in high-frequency signals.

As said, features and characteristics in time series data are often dynamic over time, meaning that they do not refer to individual data points, but rather to time windows or entire time periods. However, the length of these time windows depends heavily on the use case and the concrete features that are relevant to the context. In order to be able to adequately represent key features in the model input, the question of the length of the time periods that are directly transferred to the model is therefore critical. The investigation of the optimal length of time windows, also known as "windowing", is therefore an issue that must be reanalyzed with each new application.

The temporal dynamics of features have another consequence if time series need to be analyzed in real time: If a feature refers to a period of time, this period must first be waited out in a live analysis before it can be detected in the data.

3.5 Other special aspects of time series data

In addition to the special aspects of time series data (in terms of structure and handling), which can be directly assigned to the four data preparation phases (P1-4), three extra aspects should be taken into account:

A1. Data Shuffling: To avoid overfitting, ML models are often trained multiple times with different sample orders in the training batches. These are often shuffled randomly. Since the temporal relationship between data points usually plays an important role in time series, shuffling is rarely permissible.

A2. Dimensionality of Data: Especially in the energy or IoT sector, it is typical for time series that come from a single source (as for example a smart meter) that they are univariate and therefore do not have a high information density. For this reason, combining multiple data sources is a common scenario. The resulting data heterogeneity brings with it several difficulties. The handling of different data frequencies and size scales in the data has already been discussed in the section on data preprocessing. Another problem is establishing semantic relationships and correlations between values from different sources when the temporal resolutions differ greatly from one another.

A3. Domain Shift: Time series are often subject to trends and seasonal effects that lead to changes in the distribution of the underlying data (Du et al. 2021). These effects are particularly significant in the energy sector, where weather conditions have a major influence on consumption and generation patterns. A trained model is only able to detect learned patterns and regularities during the inference phase if the data distributions during training and inference are identical (Goodfellow et al. 2016). If the distribution changes due to external influences on the domain, the stable performance of the model can no longer be assumed.

4 REQUIREMENTS FOR THE ML LIFE CYCLE IN THE CASE OF TIME SERIES

The outlined specifics and difficulties in data handling will now be used to derive specific requirements for the ML life cycle when dealing with time series data.

The life cycle for the development of ML applications in general is not uniformly modelled in the literature (MLOps: Continuous delivery and automation pipelines in machine learning; IBM-Whitepaper 2015; Studer et al. 2021; Best practices by ML lifecycle - Machine Learning Lens 2026). Therefore, the derived time series-specific requirements will not be discussed in relation to a particular model for the ML life cycle. Instead, those requirements can serve as a starting point for developing a prototypical reference model that is specifically applicable to time series. This is an important direction for future work.

To deduce the time series specific requirements, we review the special aspects within the four phases of data preparation (P1-4) as well as the three additional aspects (A1-3):

Consequences from specifics for (P1)-Data Collection:

The need for domain-specific data collection has two major consequences for the operation of ML applications with time series data:

R1. *Initial Phase:* The ML life cycle shall include an initialization phase in which use-case-specific data is collected, during which no training or inference is performed, and during which data labeling is supported if required; the duration of this phase shall be configurable and application-dependent.
R2. *Model Performance Evaluation:* In the absence of standardized training datasets and generally accepted benchmarks, the ML life cycle shall include a mandatory, application-specific performance evaluation design, in which appropriate evaluation metrics and validation procedures are explicitly defined.

Consequences from specifics for (P2)-Data Preprocessing:

Data preprocessing is an essential part of all ML applications. Normalization and standardization procedures constitute established and widely adopted preprocessing practices.

However, consideration of time granularity is an essential requirement in the preprocessing of time series data. The analysis and, if necessary, adjustment of the time resolution must be reconsidered and carried out for each new use case.

R3. *Analysis and Adjustment of Time Resolution:* The ML life cycle shall include a dedicated step for assessing and, where required, transforming the temporal sampling characteristics of time series data, including (i) detection of irregular time intervals, (ii) selection of an application-specific interpolation strategy for gap handling (particularly for large gaps), and (iii) evaluation and specification of an appropriate temporal resolution (e. g., down sampling) to support pattern visibility, optimize signal-to-noise ratio and achieve computational feasibility for training and inference.

Consequences from specifics for (P3)-Data Augmentation:

The complexity and abstract nature of data augmentation methods for time series come with greater effort and the need to tailor them to each individual case. Therefore, another requirement is:

R4. *Custom Data Augmentation:* The ML life cycle shall treat data augmentation as an optional, case-dependent step and, when augmentation is considered, shall require a cost-benefit assessment that compares complex techniques (e. g., dynamic time warping, variational autoencoders, generative adversarial networks) against simpler alternatives (e. g., jittering) in terms of expected model performance gains, implementation effort and computational overhead.

Consequences from specifics for (P4)-Feature Engineering:

The individuality of features in time series data means that there are no universal routine procedures for feature engineering of time series. It must therefore be reworked and implemented from context to context. This represents a further special requirement for the ML life cycle:

R5. *Custom Feature Engineering:* The ML life cycle shall include a mandatory, use-case-dependent feature engineering step, in which temporal feature construction is explicitly addressed; where relevant, this shall include defining and tuning an appropriate windowing scheme (i.e. window length and stride) to capture time-dependent feature dynamics.

Consequences from specifics for (A3)-Domain Shift:

Due to the problem that shifts in the data domain are often to be expected when using time series, consequently the following requirements must be placed on the ML life cycle:

R6. *Regular Retraining*: The ML life cycle shall support regular model retraining to maintain robust performance under expected changes in the inference data distribution, including the capability to execute retraining runs and manage resulting model versions.
R7. *Implementation of a Suitable Trigger Logic for Retraining:* The ML life cycle shall require an explicit operational definition of distribution change and retraining criteria and shall support both (i) indicator-based approaches that monitor statistical data descriptors over time and initiate retraining when deviations exceed defined thresholds and (ii) schedule-based retraining at predefined intervals independent of detected shift.
R8. *Continuous Data Collection during the Inference Phase:* The ML life cycle shall ensure continuous retention of a rolling window of recent historical data sufficient to retrain models on the current data distribution.

No specific requirements for an ML life cycle can be derived from **(A1-Data Shuffling)** and **(A2-Dimensionality of Data)**: First, this follows from the fact that the potential inadmissibility of shuffling training data in the case of time series does not have a direct implication for the architecture of an ML life cycle. Moreover, no specific consequence for the ML life cycle can be derived from the need to robustly transform data originating from multiple heterogeneous sources - which often have to be consolidated due to the low dimensionality of individual sources - to a uniform scale, as this constitutes a standard procedure applied across data modalities.

5 CONCLUSIONS

This article examines the extent to which the ML life cycle for time series data needs to be considered specifically when compared to the situation with other types of data. To this end, we first provide an overview of current research on the topic. While many studies highlight specific features in individual parts of the life cycle in the case of time series, there is no sufficient discussion in the literature about a holistic general structure for the time series ML life cycle. In this regard, we first examine the specifics of time series in data handling (compared to image and text data) and derive a proposal for necessary requirements for the life cycle afterwards. Concerning this we identify key time series-specific challenges and considerations across four core phases of data preparation: *data collection* (P1), *data preprocessing* (P2), *data augmentation* (P3), and *feature engineering* (P4). Beyond these phase-aligned aspects, we also highlight three cross-cutting issues that cannot be cleanly assigned to one of those phases: *data shuffling* constraints (A1), *data dimensionality* characteristics (A2) and susceptibility to *domain shift* (A3). Building on these time-series-specific characteristics, we derive eight requirements for an ML life cycle: Implementation of an *Initial Phase* for collection of domain specific data (R1), Definition of a robust use case specific *Model Performance Evaluation* (R2), *Analysis and Adjustment of Time Resolution* as one of the main steps in preprocessing time series data (R3), Definition of *Custom Data Augmentation* strategies (R4), Definition of *Custom Feature Engineering* (R5), Implementation of *Regular Retraining* (R6), *Implementation of a Suitable Trigger Logic for Retraining* (R7) and *Continuous Data Collection during the Inference Phase* for retraining jobs (R8). Together, these requirements translate the identified time series-specific challenges into concrete life cycle design implications for time-series applications. Future work could use (R1-8) as conceptual foundation for developing a reference model for a time series ML life cycle. Such a reference model would serve as a blueprint for engineering and operating time series-based ML systems with robust long-term performance.

ACKNOWLEDGEMENTS

The work presented in this paper is co-funded by the Federal Ministry for Economic Affairs and Energy, co-financed by tax revenue based on the budget approved by the Saxon state parliament and co-funded by the European Union.

REFERENCES

Barandas M, Folgado D, Fernandes L, Santos S, Abreu M, Bota P, Liu H, Schultz T, Gamboa H, 2020. TSFEL: Time Series Feature Extraction Library. SoftwareX, 11, 100456.

Best practices by ML lifecycle - Machine Learning Lens, 2026, 04.02.2026. Accessed on 04.02.2026, https://docs.aws.amazon.com/wellarchitected/latest/machine-learning-lens/best-practices-by-ml-lifecycle-phase.html

Cerqueira V, Moniz N, Soares C, 2024. VEST: automatic feature engineering for forecasting. Machine Learning, 113 (7), 4523–4545.

Common Crawl - Open Repository of Web Crawl Data, 2026, 02.02.2026. Accessed on 03.02.2026, https://commoncrawl.org/

Conrad F, Mälzer M, Lange F, Wiemer H, Ihlenfeldt S, 2024. AutoML Applied to Time Series Analysis Tasks in Production Engineering. Procedia Computer Science, 232, 849–860.

Cui J, Kamarthi H, Kong L, Liu H, Prakash B, Sasanur A, Sharma M, Wen Q, Xu S, Zhang C, Zhao Z, 2024. Time-MMD: Multi-Domain Multimodal Dataset for Time Series Analysis, 77888–77933.

Deng D, Karl F, Hutter F, Bischl B, Lindauer M, 2022. Efficient Automated Deep Learning for Time Series Forecasting, 13715, 664–680.

Du Y, Wang J, Feng W, Pan S, Qin T, Xu R, Wang C (eds.), 2021. AdaRNN: Adaptive Learning and Forecasting of Time Series.

Goodfellow I, Bengio Y, Courville A, 2016. Deep learning. The MIT Press, Cambridge, Massachusetts, London, England, 775 p.

Hu C, Sun Z, Li C, Zhang Y, Xing C, 2023. Survey of Time Series Data Generation in IoT. Sensors (Basel, Switzerland), 23 (15).

IBM-Whitepaper, 2015. Foundational Methodology for Data Science, IBM Analytics White Paper (2015) “Foundational Methodology for Data Science”, [online],

Iglesias G, Talavera E, González-Prieto Á, Mozo A, Gómez-Canaval S, 2023. Data Augmentation techniques in time series domain: a survey and taxonomy. Neural Computing and Applications, 35 (14), 10123–10145.

Jeffery SR, Alonso G, Franklin MJ, Hong W, Widom J, 2020. Declarative Support for Sensor Data Cleaning. ICT Express, 3968 (3), 83–100.

Li B, Hou Y, Che W, 2022. Data augmentation approaches in natural language processing: A survey. AI Open, 3, 71–90.

Lowther AP, Fearnhead P, Nunes MA, Jensen K, 2020. Semi-automated simultaneous predictor selection for regression-SARIMA models. Statistics and Computing, 30 (6), 1759–1778.

Lubba CH, Sethi SS, Knaute P, Schultz SR, Fulcher BD, Jones NS, 2019. catch22: CAnonical Time-series CHaracteristics. Data Mining and Knowledge Discovery, 33 (6), 1821–1852.

Maslej N, Fattorini L, Brynjolfsson E, Etchemendy J, Ligett K, Lyons T, Manyika J, Ngo H, Niebles JC, Parli V, Shoham Y, Wald R, Clark J, Perrault R, 2023. Artificial Intelligence Index Report 2023, unpublished.

Meisenbacher S, Turowski M, Phipps K, Rätz M, Müller D, Hagenmeyer V, Mikut R, 2022. Review of automated time series forecasting pipelines. WIREs Data Mining and Knowledge Discovery, 12 (6).

MLOps: Continuous delivery and automation pipelines in machine learning. Accessed on 04.02.2026, https://docs.cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

Mumuni A, Mumuni F, 2025. Automated data processing and feature engineering for deep learning and big data applications: A survey. Journal of Information and Intelligence, 3 (2), 113–153.

Qiu X, Hu J, Zhou L, Wu X, Du J, Zhang B, Guo C, Zhou A, Jensen CS, Sheng Z, Yang B, 2024. TFB: Towards Comprehensive and Fair Benchmarking of Time Series Forecasting Methods. Proceedings of the VLDB Endowment, 17 (9), 2363–2377.

Raji ID, Bender EM, Paullada A, Denton E, Hanna A, 2021. AI and the Everything in the Whole Wide World Benchmark, unpublished.

Rätz M, Javadi AP, Baranski M, Finkbeiner K, Müller D, 2019. Automated data-driven modeling of building energy systems via machine learning algorithms. Energy and Buildings, 202, 109384.

Ridnik T, Ben-Baruch E, Noy A, Zelnik-Manor L, 2021. ImageNet-21K Pretraining for the Masses, unpublished.

Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L, 2015. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 115 (3), 211–252.

Schaffererin, 2025. Concepts - Machine learning operations (MLOps) for AI and machine learning workflows - Azure Kubernetes Service, 16.12.2025. Accessed on 05.02.2026.

Sculley D, Holt G, Golovin D, Davydov E, Phillips T, Ebner D, Chaudhary V, Young M, Crespo J-F, Dennison D, 2015. Hidden Technical Debt in Machine Learning Systems. Advances in Neural Information Processing Systems, 28.

Sergio AT, Lima TP de, Ludermir TB, 2016. Dynamic selection of forecast combiners. Neurocomputing, 218, 37–50.

Shankar S, Garcia R, Hellerstein JM, Parameswaran AG, 2024. "We Have No Idea How Models will Behave in Production until Production": How Engineers Operationalize Machine Learning. Proceedings of the ACM on Human-Computer Interaction, 8 (CSCW1), 1–34.

Siino M, Tinnirello I, La Cascia M, 2024. Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers. Information Systems, 121, 102342.

Silva D de, Alahakoon D, 2022. An artificial intelligence life cycle: From conception to production. Patterns (New York, N.Y.), 3 (6), 100489.

Studer S, Bui TB, Drescher C, Hanuschkin A, Winkler L, Peters S, Müller K-R, 2021. Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology. Machine Learning and Knowledge Extraction, 3 (2), 392–413.

Tawakuli A, Havers B, Gulisano V, Kaiser D, Engel T, 2025. Survey:Time-series data preprocessing: A survey and an empirical analysis. Journal of Engineering Research, 13 (2), 674–711.