Representation Learning for Zero-Shot Anomaly Detection

Johannes Hölker

[1] University of Kassel
[2] Germany
johannes.hoelker@uni-kassel.de

Abstract. The detection of anomalies in time series data is subject of current research in machine learning. Several methods are published in recent years which try to detect anomalies in various ways. Mostly the patterns that represent time series are learned. This is particularly useful for a general knowledge about the data. The concepts used for this include CNNs, Contrastive Learning, Autoencoders and Transformers. This paper provides an overview of the recent developments in anomaly detection and presents methods that try to detect anomalies in time series data. The found methods are evaluated with regard to Zero-Shot Learning and a multivariate input. Methods that are publicly available are investigated further using a completely new dataset with known anomalies provided by SMA. A proof of work in a Zero-Shot scenario without training the models is presented. None of the methods found the correct time point of an anomaly.

Keywords: Representation Learning · Zero Shot Learning · Anomaly Detection · Multivariate Time Series.

1 Introduction

Nowadays sensors can be found everywhere and they become more popular across multiple domains. Gyroscopes, cameras, compasses and accelerometers are integrated in smartphones. Physical machines are tracking their movement through vibration sensors, health care systems in hospitals visualize the heart beat of a patience and voltmeters measure the generated power in a solar plant. Everytime sensor values are collected, time series data (TSD) is produced.

In some scenarios the measurements of different sensors are combined. Physical machines sometimes track vibration and motor rotations, health care systems visualize the heart beat and body temperature and solar plants measure voltage and current. Collections of different sensor measuring at a common time window produce Multivariate Time Series Data (MVTSD).

Applications that produce MVTSD may evaluate the data and further decisions depend on a correct analysis. Normally the data is consistent and values change constantly in repetetive patterns. This is when the machine, the patience health or the solar plant is functioning like it is supposed to. But sometimes the values change unpredicted because of differing surroundings or other influences. This can lead to serious situations. Machine measurements detect a potential fault which may break the machine. When the patient's heart beat changes its pattern the health of the patient is seriously endangered. And a solar plant may detect a decline in the generated power which should further influence the power consumption for a better efficiency.

Recognising and reacting to these changes in MVTSD can therefore be very important. But these interruptions occur in different forms. They can be recognised as outliers or they are hidden and not obviously seen as anomalies. In some cases they form shapes which never occured before. This raises the demand for a tool to detect anomalies in time series data without any further knowledge of the anomaly.

A systematic literature review concerning the topic is conducted and the best choices are implemented on a test data set. First basic necessary terms are explained in section 2. In section 3 the basic methodology used in this paper is described and the main research questions are formulated. In section 4 the found papers and their methodologies are presented and explained. The methodologies are compared and evaluated for usability in the context of Zero-Shot Anomaly Detection in section 5. The implementation of suitable techniques is provided in section 6. Finally the results are discussed and concluded in section 7

2 Definitions and Conventions

The basic expressions used in this paper are explained in the following chapter. First Representation Learning is defined and the different approaches to find representations in data are explained. Afterwards Zero Shot Learning as well as Anomaly Detection are described and explained.

2.1 Representation Learning

Variations in data are not always visible for a human and even less possible to label them accordingly. It is important for artificial intelligence to detect representations in data by machines. A machine should be able to extract information hidden in the low-level sensor measurings and continue working with the representations instead of the raw data. Being able using a representation as an input to a supervised predictor is the main requirement for a good representation [5].

Representation Learning (RL) tries to detect meaningful interconnections in data relevant for further data analysis. These interconnections represent abstract information, so called background knowledge [27].

In neural networks representations are learned in every layer whereas the representations in hidden layers are incomprehensible to humans. They are produced by weights and biases and build so called neural representations. At a higher level of abstraction, these neural representations can be understood as spatial representations within a conceptual space, where concepts are represented as points or regions. When these spatial representations are transformed into language, they become symbolic representations, which are used to convey meaning in a human-understandable form. Together, neural, spatial, and symbolic representations build cognitive repesentations [18].

To extract representations a knowledge discovery process with different methods of machine learning and data mining methods are used. During the process representations are learned by the model. RL methods are divided into Propositionalization as symbolic representations and Embeddings as spatial representations [27, p. 4].

RL occurs in several machine learning areas. Depending on the underlying concept, different strategies to extract representations can be found. They work different in detecting patterns and store them in different ways [6].

In [15, p. 525] a general detailed description of representation learning is given. They summarize that representations should make the subsequent learning tasks easier. This implies that to find the best fitting representation and the underlying representation learning technique, we need to know the task it should perform afterwards.

Concepts The most straight-forward approach to detect representations are Multi Layer Perceptrons (MLP). An input vector is processed by interconnected artificial neurons. The neurons build layers starting with an input layer, followed by hidden layers with a final output layer. The produced output layer typically classifies the input and predicts the label. The difference between predicted and labeled output indicates the performance of the network. Adjusting the interconnections using weights and biases of each neuron enables a learning process. [39]

Convolutional Neural Networks (CNN) are a variation of the MLP building subsets of the input vector. This is mainly used in image processing.

Recurrent neural networks (RNN) are another variation of the traditional feed-forward MLP. Every neuron has an additional input containing the previous state. This is especially useful for time series data.

Traditional neural networks like MLP, RNN, and CNN have limitations in learning robust, generalizable, and semantically meaningful representations, especially with limited labeled data [53].

Learning representations in time series data is done in several different ways. One solution according to [72] is contrastive learning (CL). Pairs of data points are labeled as similar and dissimilar. These data points are put into a feature space where the distance between the two represents their similarity. Similar data points are grouped together and dissimilar data points are distant from each other. With a contrastive loss function and a label of similarity between two points, the model is trained by putting the similar data points together and separating dissimilar points. Using this method groups of similar data points are formed [53].

Autoencoders are another important method in representation learning. An autoencoder is a framework implemented by neural networks. It is used to learn efficient codings of input data in an unsupervised manner. It consists of an encoder that compresses the input into a latent-space representation and a decoder that reconstructs the input from this representation. The goal is to minimize the difference between the real and the reconstructed input.

Transformers, initially developed for natural language processing tasks, have become a powerful tool in representation learning. They use self-attention mechanisms to weigh the significance of each part of the input data differently, enabling the model to capture long-range dependencies. The dependencies represent abstract and valuable information [61].

Based on the transformer architecture, Large Language Models (LLM) are developed and used increasingly in different applications. Known as chatbots they can help in language specific tasks. Beside that they can be used in anomaly detection and forecasting.

In summary, representation learning can be achieved using different techniques, each suitable for different types of data and tasks. From neural networks and autoencoders to transformers, these methods provide the tools necessary to transform raw data into meaningful representations that facilitate further analysis and learning.

2.2 Anomalies in Time Series

Several definitions of anomalies in data can be found in literature. In this paper the definition of [17, p. 54] is used. It seperates anomaly and novelty detection as different tasks. Anomalies can be understood as outliers from the regular class. But these anomalies can vary in their cause. If there is a specific cause and the anomalies occur in its own cluster, they form a novelty. If instead the outliers randomly occur with no specific root cause, they are called noise. The cause for noise then is of a different kind and cannot be classified. Figure Figure 1 visualizes the different

Instead of dividing anomalies by their cause the shape of anomalies can vary in several ways. In real measurement data of any shape is possible and it is totally unpredictable [52]. For training purposes anomaly injection is crucial. Then the anomalies are simulated as point anomalies or subsequence anomalies. Point anomalies occur once and can be global or contextual. Subsequence anomalies on the other hand change the values in a given time window or on long term. They can be divided in seasonal, shapelet and trend anomalies (see Figure 2). Seasonal and shapelet anomalies change the values in a limited time window, trend anomalies are changing all following values [9, p. 9].

In this paper we want to focus on single time events, which are in any case anomalies because they cannot form its own cluster. This defines our goal as an Anomaly Detection (AD) task.

2.3 Zero Shot Learning

In this paper the definition made by [42] is used. They seperate Single Task Learning, where every model is trained seperately for each task, from Multi Task

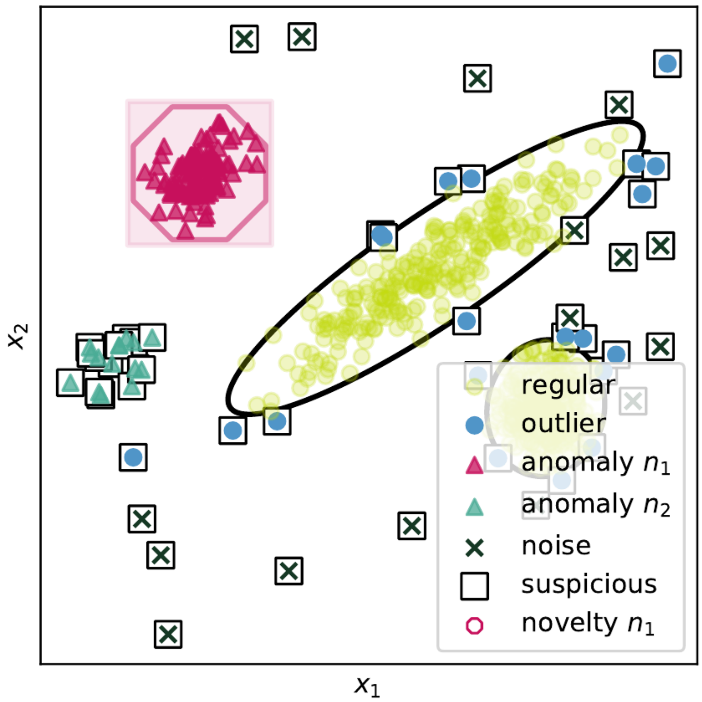

Fig. 1. Classification of outliers [17, p. 54]

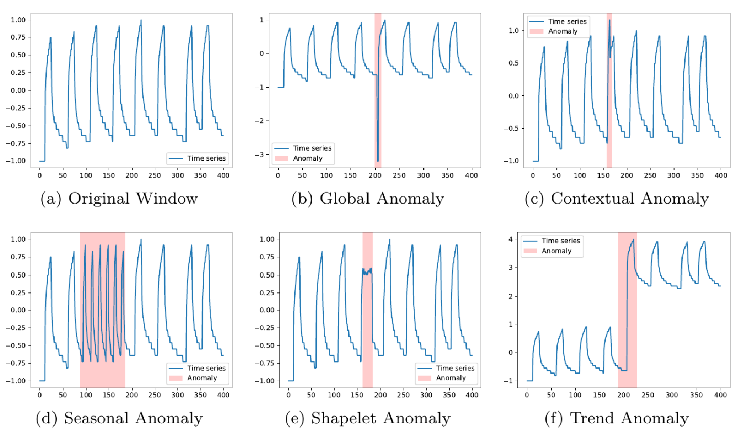

Fig. 2. Classification of time series anomaly types [9]

Learning (MTL) where one model is trained and evaluated on several tasks. For Zero Shot Learning (ZSL) in comparison the model is trained on several tasks like in MTL but tested on completely new ones.

Zero Shot Learning is therefore an extreme form of transfer learning. While transfer learning is the concept of transferring the knowledge and weights gained at one task using them at solving another task, Zero-Shot Learning means there are no samples for the other task. The transformation of knowledge can help solving tasks where there are few or no samples available. The gained knowledge is normally stored as representations of data. Representations which are abstract enough to not see a specific item but information about items. This also means that ZSL is only possible because additional information has been discovered during training [15, p. 536].

[43] were the first to implement a successful Zero-Shot AD followed by [55] who used semantic word vector representations to classify words in groups with a fully unsupervised model.

Zero-shot learning involves training a model on certain classes and then testing its ability to recognize new, unseen classes without any retraining. In the context of AD, this means the model should be able to detect types of anomalies it has not encountered during the training phase.

3 Systematic Literature Review

A literature review to contribute in the development of an AD tool is presented in this paper. It provides an overview on the latest trends in representation learning and extracts the possible solutions addressing the problem described in section 1. The review conforms to the methodology presented by [25]. First the research questions are formulated. Finally Inclusion and Exclusion Criteria are formulated in order to filter the found literature for the application. The search process and the websites used are listed.

Further analysis with a systematic quality assessment and data collection like in [25] are excluded.

3.1 Research question

The covered topic includes different areas of machine learning, all being further developed in recent years. In order to break it down into separate concerns the following research questions are formulated:

- RQ1: How can representations be learned using machine learning?
- RQ2: Which RL concepts can be used for multivariate time series?
- RQ3: How to use RL for AD?
- RQ4: Which methods are useful for ZSL scenarios?

These question form a path for further chapters. RQ1 and RQ2 are explained in section 2. Answering RQ3 involves a literature review in section 4 which presents useful methods. RQ4 is answered in section 5. The research questions build a basis for the formulation of the following Criteria.

3.2 Inclusion and Exclusion Criteria

This paper focuses on published methods for AD in Zero Shot Scenarios on MVTSD. In order to structure the search for and selection of relevant articles, the necessary guidelines are formulated below. Articles that are considered in more detail must meet the following inclusion criteria:

- IC1: Methods using a representation learning concept
- IC2: Methods handling time series data
- IC3: Methods used for AD
- IC4: Published in recent years ($<$ 6 years)

The chosen articles are examined in more detail. They are described and explained in section 4. Using the gained knowledge all described articles are filtered by the following exclusion criteria in section 5.

- EC1: Methods not tested on Zero-Shot Learning
- EC2: Methods designed for univariate data
- EC3: Multiple publications reporting the same methodologies
- EC4: Methods with restricted availability

Using these exclusion criteria ensures to find methodologies that meet the desired use case described in the research questions.

EC1 excludes methods that are not tested in a Zero-Shot Learning scenario. The second exclusion criteria filter for methods handling multiple input variables only. EC3 avoids duplicated papers and EC4 ensures that the method is publicly available and does contain a description on how to implement and reproduce the outcomes.

3.3 Search process

A manual search of specific conference proceedings and journal papers was made. Considering the pace on which new developments emerge in the area of machine learning the help of research tools was needed. Specifically in the field of AD the publications are made in recent years. This makes it difficult to assure finding every relevant paper.

The main tool used to find papers was Consensus, which is an academic search engine. They use large language models (LLMs) and purpose-built search technology. The chatbot is based on ChatGPT 4.0 and should answer questions based on papers including their reference. For reassuring the existence of the papers conventional bibliographies are used.

4 Representation Learning Methods

In this chapter any found paper proposing a RL strategy used for time series data with adaptability on AD tasks is presented. For the literature research the inclusion criteria as described in subsection 3.2 are applied.

The different RL strategies are explained focusing on compliance of the exclusion criteria. The strategies are organized by their underlying concept. First straight-forward methods which are based on one concept are presented and the complexity increases throughout the chapter. In the end combinations of different concepts are presented.

MLP Using a simple MLP is a straight-forward way to learn representations and to detect anomalies in time series data [39]. The input variable for the MLP are time points and the output variable represents the value at these time points. The model is trained to learn this mapping. With the trained model, the values in a live scenario are predicted and the difference to the actual values is measured. If the difference exceeds a certain threshold, an anomaly is found. A method called INRAD, Implicit Neural Representation of time series Data is using this concept. The method takes multiple variables as input and the model is trained with data including anomalies. It is not suitable for ZSL [21].

RNN [57] propose a method called OmniAnomaly for AD in MVTSD using a Stochastic Recurrent Neural Network to model the temporal dependencies. The key advantage of this method is its robustness to noisy and high-dimensional data according to the authors. The model learns to represent normal patterns in time series and identifies deviations from these patterns as anomalies. Since OmniAnomaly depends on having access to representative normal data to learn patterns, it is not suitable for zero-shot scenarios.

CNN Methods based on CNNs are normally used for classification of images but in recent papers they are used to detect anomalies in images. [3] develop a Texture AD and achieve a high performance in ZSL. They compare Zero-Shot against Many-Shot Learning in their work. Several image AD tools can be found ([51], [3]). But CNNs perform on time series data as well.

The main idea of using CNNs is to predict a value based on the input frame. If the distance between the predicted and the actual value exceeds a predefined threshold, the anomaly can be detected.

This idea is used to detect cyberattacks in industrial control systems. A recent study uses a dataset from a Secure Water Treatment testbed to identify cyber anomalies by measuring the statistical deviation between predicted and observed values. They explore different deep learning architectures, including CNNs and recurrent networks, and find that one dimensional CNNs perform particularly well for time series prediction tasks. Their approach successfully detects the majority of cyber attacks with minimal false positives, highlighting the effectiveness of CNNs in real-time AD in MVTSD [26]. However, the paper does not discuss the usability on ZSL. In the same area a method detecting unknown cyber-attacks is presented in [73] who use an Autoencoder which is discussed later on.

[20] use Temporal Convolutional Networks (TCN). TCNs restrict the output to be dependent on past and present time steps only. This enables them to capture temporal dependencies. By training on normal patterns, the network learns to predict future values. Significant deviations between these predictions and actual observations indicate potential anomalies. The authors do not mention if the method works in a ZSL scenario but the method only learns normal data, which indicates a usability for ZSL.

Another paper introduces a mask-based self-supervised representation learning approach to extract both short-term local dependencies and long-term global trends. By integrating forecasting and reconstruction-based models, the method effectively captures temporal contexts and feature correlations. An attention mechanism ensures feature importance, leading to better AD performance on various datasets. The method is designed for MVTSD AD but does not explicitly address ZSL scenarios [36].

Contrastive Learning In [72] a framework using CL is proposed which is applied for industrial fault detection. Two data sets that consist of various vibration signals of industrial machines and stiction sensors with multiple variables are used for training. The effectiveness of the proposed framework is demonstrated through its application to these datasets.

CL is also used for AD of time series in [9]. They use CL combined with synthetic anomaly injection. CL enables them to capture patterns in time series data and the framework shows good results on common real world datasets according to the authors. Like in the previous paper, dissimilar pairs, the anomalies, build distant data points and similar data points are close to each other. In order to train the model artificial anomalies are injected which build distant pairs. In the next stage the classification is done by the proximity of the neighbours in the representation space. Additionally anchor points representing the nearest and furthest neighbour are given from each representation as seen in Figure 3. An implementation by the authors can be found [3].

CL-TAD, a method for time series AD that uses CL and reconstruction-based techniques addresses the challenges of temporal dynamics, label scarcity, and data diversity in real-world applications. The method comprises two main components: positive sample generation and contrastive-learning-based representation learning. Positive samples are generated by reconstructing masked parts of the time series data, helping the model learn the underlying normal patterns. These samples, along with the original data, are then fed into a CL framework, which contrasts pairs of similar and dissimilar samples to learn representations [38]. While CL-TAD is not explicitly designed as a ZSL method, its use of CL and reconstruction-based techniques suggests that it could have potential in zero-shot AD scenarios. A tutorial for implementation can be found [4].

To succeed on Zero-Shot AD, One-Class Classification (OCC) can be useful. By gathering all "normal" values into a single class the outliers are directly de-

[3] github.com/zamanzadeh/CARLA
[4] github.com/nguhcv/cl-tad/tree/main

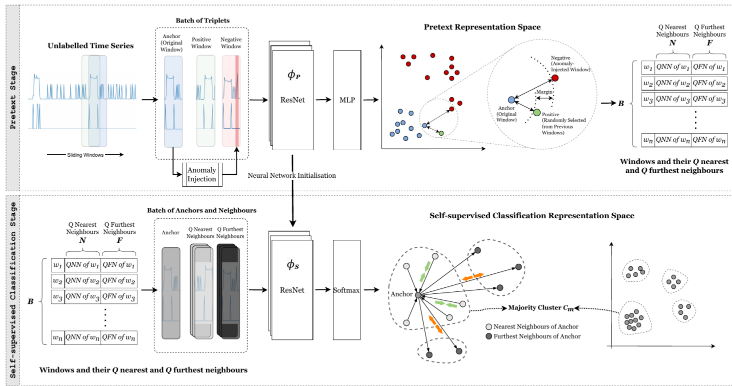

Fig. 3. Architecture of the method CARLA [9]

tected if they are outside of it. The COCA (Contrastive One-Class AD) method combines CL with OCC to improve AD in MVTSD. By treating original and reconstructed representations as positive pairs, it optimizes a contrastive one-class loss function that enhances the detection of anomalies while preventing common issues. Although COCA is designed for self-supervised AD, its ability to learn from unlabeled data suggests potential applicability in ZSL scenarios, though this has not been explicitly tested [64]. An implementation script is provided by the authors [5].

The paper by [28] presents an approach for detecting anomalies also using OCC in industrial time series data, which typically lacks labels for supervised learning. They combine OCC with CL to define a new objective function that can simultaneously learn from both models. This method enhances feature extraction while preserving temporal characteristics. The paper demonstrates the method's effectiveness through high AD performance on datasets with similar normal and anomalous data forms, highlighting its potential in industrial applications.

Unlike traditional OCC methods that map all normal instances into a single hypersphere, the method presented by [7] focuses on local contextual information. By pulling each normal instance towards its recent context window, it aims to better detect context-based anomalies. The model uses a deterministic contrastive loss, which improves the network's ability to distinguish between normal and abnormal data. The authors did not test the method in a ZSL scenario and no public implementation can be found.

[71] introduce TS2Vec, a framework for learning robust and universal time series representations at multiple levels through hierarchical CL. TS2Vec captures characteristics of different time series instances and dynamic temporal patterns within each series. The method is tested on ZSL and is applicable for MVTSD.

[5] github.com/ruiking04/COCA

The instructions on reproducing the outcomes are made publicly available by the authors [6].

Another paper introduces an autonomous system for AD in MVTSD also using CL. The proposed method TimeAutoAD automates model configuration and hyperparameter optimization. It uses self-supervised CL to enhance the model's ability to differentiate normal and anomalous time series by generating pseudo-negative samples. The method is tested on real-world datasets. The paper does not explicitly address zero-shot AD [22].

The ContrastAD framework presented in [30] is a self-supervised method for time series AD that leverages CL with temporal transformations. The key innovation is the use of anomaly-induced transformations to create representations that differentiate between normal and abnormal data. This approach targets both point anomalies and contextual anomalies in MVTSD. By learning representations for normal and anomalous data in the latent space, ContrastAD improves performance on noisy and complex datasets. However, the method is not trained or validated on entirely new types of anomalies.

Another model called DCdetector presented in [68] is an attention-based CL framework designed for time series AD. It utilizes a dual attention asymmetric design to create a permutation-invariant representation, guiding the learning process. This approach enhances the model's ability to discriminate between normal and anomalous data. Extensive experiments demonstrate that DCdetector achieves state-of-the-art performance across multiple benchmark datasets. While the paper focuses on its effectiveness in AD, it does not explicitly address or test the model's applicability to ZSL scenarios [68] In the article a repository containing bash scripts for implementation can be found [7].

[47] presents a novel approach for AD in MVTSD using Contrastive Predictive Coding (CPC). Their method, named time series Representational Learning through Contrastive Predictive Coding (TRL-CPC), aims to capture the temporal dependencies and correlations across multiple variables in time series data. The TRL-CPC framework consists of an encoder, an auto-regressive model, and a non-linear transformation model. These components are optimized to learn the representations of MVTSD by predicting future segments from past segments. The core idea is to maximize the mutual information between the encoded representations of past and future segments, thereby learning robust representations. To detect anomalies, TRL-CPC calculates the prediction error between actual future segments and the predicted segments generated by the CPC model. Anomalies are identified where this prediction error exceeds a certain threshold, enabling unsupervised AD based on the structure of the data itself.

CPC is also used by the method TiCTok presented in [23]. The model proposes an approach to MVTSD AD with a time series token encoder. This encoder converts raw time series data into latent embeddings that capture wide-ranging temporal information. The model employs CL to produce representations, which help distinguish between normal and anomalous data. Additionally, TiCTok in-

[6] github.com/zhihanyue/ts2vec

[7] github.com/DAMO-DI-ML/KDD2023-DCdetector

troduces a new anomaly scoring method based on the contrastive loss used during training. In their paper they do not test the model on ZSL scenarios.

Another paper introduces Multiview Graph CL for detecting anomalies in MVTSD, particularly in IoT systems. The method constructs graph structures to model both temporal context and signal dependencies, while an adaptive data augmentation strategy generates graph views for CL. This approach enhances representation quality and improves performance in AD tasks except ZSL [49]. The method is available on Github [8].

The paper by [58] introduces TriAD (Tri-domain Anomaly Detector), a self-supervised learning method for time series AD. TriAD models features across three domains: temporal, frequency, and residual. Without relying on labeled anomalies. Unlike traditional CL, TriAD uses inter-domain and intra-domain contrastive losses to learn shared attributes among normal data and distinguish them from anomalies. The approach is designed to handle anomalies of varying lengths and shapes [58]. The authors provide a link to their implementation [9].

In summary methods using CL are mostly not tested on ZSL. Their robustness against noisy data is not ensured.

Autoencoder Autoencoders on the other hand are robust to unclean training datasets [1, p. 2487]. Due to their architecture noise is filtered from the input.

[48] introduces an AD method using an autoencoder architecture based on Long Short-Term Memory (LSTM) networks. The core idea is that an LSTM autoencoder learns to compress and reconstruct the input time series data. During training, the model learns the normal patterns in the data by minimizing the reconstruction error. When fed new data, the model attempts to reconstruct it, and any significant reconstruction error signals an anomaly. This approach is particularly effective because LSTMs are well-suited to capture temporal dependencies, making them ideal for time series data. The model works without requiring labeled datasets, making it an unsupervised solution. The authors test the model on both synthetic and real-world data, such as sound event detection. It can effectively detect outliers based on reconstruction error but does not have the capacity for ZSL.

A method called LATAM (Long short-term memory Autoencoder with Temporal Attention Mechanism) also uses LSTMs for AD in MVTSD. Temporal dependencies are combined with dynamic thresholding, which adapts the threshold throughout the evaluation of the model. They train the model on inverter data including failures and evaluate it on other benchmark datasets explicitly in few-shot scenarios [41] The source code is provided by the authors [10].

Other methods that use LSTM networks are found in literature. All of them are designed for MVTSD but are not tested on ZSL. The authors did not provide instructions on how to replicate the outcomes [70] [13] [40].

[8] github.com/shuxin-qin/MGCLAD

[9] github.com/pseudo-Skye/TriAD

[10] github.com/anonymousgit234/FewShot-Anomaly-Detection-using-LATAM

[46] proposes a Multi-Scale Temporal Variational Autoencoder (MST-VAE) for AD in MVTSD. MST-VAE combines short and long-scale convolutional kernels within a 1D CNN and a Variational Autoencoder to capture both short-term and long-term temporal patterns. The method is not directly tested in a Zero-Shot scenario. An implemention is provided by the authors [11].

To detect anomalies in healthcare data a variational recurrent autoencoder (VRAE) is used in [45]. They created an unsupervised framework where the model learns to represent the data and detect anomalies. During training, they add noise to the input data, and the model tries to reconstruct the original, uncorrupted data. This helps the model learn more robust representations of the data. To detect anomalies, they cluster these learned representations and calculate the distance to identify outliers. Their approach was tested on a benchmark electrocardiogram dataset and showed that it could effectively detect unusual heartbeats.

Another approach using VRAE involves creating synthetic anomalies to improve the detection process. First, they distill feature descriptors of normal data points into more robust representations using AEs. These representations are then refined using a VAE that creates a family of distributions. From these distributions, they select those that lie on the outskirts of the normal data as generators of synthetic anomalies. By generating these synthetic anomalies, they train binary classifiers to distinguish between normal and abnormal data. Their hierarchical structure for feature distillation and fusion helps create robust representations, enabling effective AD without needing actual anomalies during training [50].

Kieu et al. propose Variational Quasi-Recurrent Autoencoders (VQRAEs) for unsupervised time series AD, using robust divergences to improve resilience against noisy data. VQRAEs utilize Quasi-Recurrent Neural Networks (QRNNs) for efficient temporal dependency capture, and a bi-directional version (BiVQRAEs) enhances accuracy by processing data in both forward and backward directions. However the method was not explicitly tested on ZSL [24]. An implementation is provided by the authors [12].

A previously mentioned paper addresses the challenge of detecting unknown cyberattacks by applying ZSL. The proposed method maps the features of known attacks to a semantic space using a sparse autoencoder and restores them to the feature space by minimizing reconstruction errors. This technique enables the model to detect previously unseen attacks by generalizing from known attack features. The research highlights the feasibility and effectiveness of ZSL for cybersecurity applications [73].

A method called Fused Sparse Autoencoder and Graph Net (FuSAGNet) is used for AD in MVTSD. The method combines a Sparse Autoencoder (SAE) to learn sparse latent representations and a Graph Neural Network (GNN) to forecast future time series behavior. It captures both temporal dependencies and complex interfeature relationships by learning graph structures from the data.

[11] github.com/tuananhphamds/MST-VAE

[12] github.com/tungk/Bi-VQRAE

This joint optimization approach aims to improve AD performance by fusing reconstruction and forecasting tasks. The method was empirically tested on three real-world datasets related to industrial systems but not in ZSL scenarios [19] [13].

Deep Autoencoding One-Class (AOC), a method for time series AD that combines autoencoder-based reconstruction and one-class classification in a single-stage approach to better capture normal patterns. The method is evaluated on public datasets, outperforming baseline models and proving effective for detecting various types of anomalies in both univariate and MVTSD. However, ZSL is not tested or addressed in the proposed framework [37] [14].

Another paper presents a method named RANSynCoders for detecting anomalies in asynchronous multivariate time series by combining spectral analysis with autoencoders. This approach improves performance on high-dimensional, asynchronous data by reducing false positives and providing better localization of anomalies. The method in the paper does not specifically mention being tested in zero-shot scenarios [1]. An implemention of the method is provided by the authors [15].

Another proposed method is using a Temporal Convolutional Network Autoencoder (TCN-AE) designed for unsupervised AD in MVTSD. It employs dilated convolutions to capture long-range dependencies and compresses time series into representations using an autoencoder. The method detects anomalies by evaluating reconstruction errors [59]. A Repository with a minimal working example can be found provided by the author but no implementation on ZSL is provided [16].

A framework called Multiscale Wavelet Graph AutoEncoder (MEGA) in presented in [63] for AD in multivariate time series. It integrates Discrete Wavelet Transform (DWT) to decompose time series into different frequency components, reconstructing them to highlight anomalies across scales. Additionally, a dynamic graph convolution network is employed to model inter-variable relationships at different scales, enhancing the detection of anomalies caused by changes in variable dependencies. This method is tested in a zero-shot scenario and explicitly takes multivariate time series as input. The code is provided by the authors [17].

The method proposed in [8] introduces DAEMON, an adversarial autoencoder framework designed for unsupervised multivariate time series AD and interpretation. The model employs two training processes: one to align the hidden variable's posterior distribution with a prior and another to minimize the difference between original and reconstructed data. This improves robustness and avoids overfitting while anomalies are detected based on reconstruction errors. There is no mentioning of testing in a zero-shot scenario. The authors of

13 github.com/sihohan/FuSAGNet

14 github.com/alsike22/AOC

15 github.com/eBay/RANSynCoders

16 github.com/MarkusThill/bioma-tcn-ae

17 github.com/jingwang2020/MEGA

the method DAEMON published python scripts for replicating the results on all datasets they used [18].

The method proposed in [14] utilizes a Gated Recurrent Unit-based Autoencoder (GRU-AE) to detect anomalies in time series data by reconstructing sequences and identifying large reconstruction errors as anomalies. It incorporates an attention mechanism to enhance performance and applies multi-timestamp stacking to reduce time steps for better training efficiency. The model is tested on real-world datasets from cellular networks, detecting anomalies at both single-day and multi-day scales. The method is applied to MVTSD, but no mention of a zero-shot scenario is made.

The method proposed in [62] presents an attention-based encoder-decoder network for AD in time series data, which focuses on learning representations in both seperate feature spaces without needing reconstruction. The attention mechanism is applied to rescale convolutional layers, highlighting the most contributive segments of the data for better representation learning. This approach improves AD by calculating belief scores in both spaces using kernel density estimation (KDE). The method is tested on multivariate time series and in a zero-shot scenario, specifically for detecting new faults but no implementation is provided by the authors.

To overcome the challenge of poorly available time series data sets [35], the model family MOMENT tries to learn general patterns on a pile of time series data following the architecture in Figure 4. The pile is a collection of different datasets which they assembled for their pretraining. According to the paper minimal finetuning is needed to perform on multivariate time series tasks like AD. They published the model and made the usage easily accessible with its own python library [19]. The time series datasets the model is trained on consist of domains including weather measurements, sensor values and power consumption datasets. They also included tongue and finger movement of humans. The different tasks which the model is evaluated on are forecasting (long and short horizon), classification, AD and imputation. Except for short-horizon forecasting all tasks are managed well. However it cannot detect anomalies in vertically shifted time series [16].

A variety of methods based on AEs can be found. But AE-based methods have remaining challenges. AEs are able to detect anomalies in general but they cannot identify the type of faulty samples [72].

Transformer Transformers on the other hand, learn the dependence of time points instead of reconstructing the input. This can possibly lead to representations that hold information about the shape of the anomalies.

A method called "Anomaly Transformer", which detects anomalies in time series by comparing a time point's learned associations with a Gaussian-based prior association to compute a Association Discrepancy. The model uses a minimax strategy to amplify the difference between normal and anomalous points

[18] github.com/Sherlock-C/DAEMON

[19] github.com/moment-timeseries-foundation-model/moment/blob/main/tutorials/anomaly_detection.ipynb

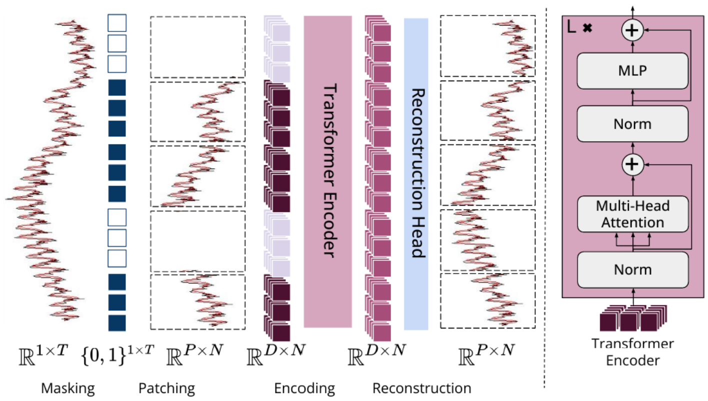

Fig. 4. Architecture of the method MOMENT [16]

based on this discrepancy. The method is not tested in a zero-shot scenario but supports multivariate time series as input [66] [20].

TranAD is a transformer-based model designed for AD in MVTSD. It uses an adversarial training to enhance both accuracy and training stability. The paper does not explicitly mention testing the method in a ZSL scenario [60] [21].

A proposed method, Decompose Auto-Transformer Network (DATN), focuses on time series AD by decomposing the input into seasonal and trend components, followed by an auto-attention mechanism for feature extraction. The method takes multivariate input, where the decomposition and attention mechanisms operate on MVTSD. The paper does not mention ZSL explicitly [65].

Time Series Anomaly Transformer (TiSAT), introduces a transformer-based approach that captures long-range temporal dependencies in time series AD using a probabilistic attention mechanism, which improves computational efficiency by focusing on key query-value pairs. The model takes multivariate input but the authors do not mention ZSL [10] [22].

The TCF-Trans method enhances AD in MVTSD by using a feature fusion decoder that combines shallow and deep layers to capture important anomaly details while resisting noise. It also introduces a temporal context fusion module to adaptively merge predictions for more robust results. Although the method was tested on various datasets, it is not explicitly evaluated in a strict zero-shot scenario [44].

The DCT-GAN method proposed in [33] integrates Dilated Convolutional Networks with Transformers within a GAN framework to enhance the detection of anomalies in time series data, improving generalization and accuracy. There is no specific mention in the document about testing the method on ZSL tasks.

[20] github.com/thuml/Anomaly-Transformer

[21] github.com/imperial-quore/TranAD

[22] github.com/kevaldoshi17/TiSAT

The AnoFormer method is a Transformer-based GAN framework for unsupervised multivariate time series AD, utilizing a two-step masking strategy to improve normal data representation and AD. The model masks parts of the data randomly, then remasks uncertain regions based on entropy to refine its focus. While the method excels in unsupervised learning, it is not explicitly tested in a ZSL scenario [54].

A method detecting anomalies in tabular data using Large Language Models (LLM) is presented in [29]. In order to perform tasks that LLMs are not directly build for they generate synthetic datasets. Using these datasets LLMs and specifically GPT-4 have comparable performance with transductive learning methods [29, p. 6]. Concerning different batches, the method is tested in ZSL. The model is trained on some batches and evaluated on new ones. Due to the fact that the prompt contains data of one variable, the method is not used for multivariate time series.

[56] examine a literature review on how LLMs perform on anomaly detection tasks concerning time series data. LLMs in anomaly detection are specifically useful when the time series data is in the form of words. This can be the case in log analysis. Logs are generated over time and hold a lot of information which can detect errors and system failures. They conclude that LLMs have potential in detecting anomalies but challenges remain. The occurrence of hallucinations and the need for computational efficiency to name a few.

Using Fourier Analysis and the transformer architecture is emplloyed in [69] to detect anomalies in time series data. The encoder of a transformer is used to capture temporal features of time series. They detect frequencies by using Fourier Analysis.

Shapelet Learning [4] address the problem of detecting anomalies in time series data using a novel unsupervised method based on shapelet learning. Their method learns representative features that describe the shape of time series data from the normal class and simultaneously learns to accurately detect anomalies. The objective function encourages the learning of a feature representation in which normal time series lie within a compact hypersphere, while anomalous observations lie outside the decision boundary. The advantage of their approach is that it can efficiently detect anomalies in unseen test data without retraining the model, by reusing the learned feature representation. Experimental results on multiple benchmark datasets demonstrate the robustness and reliability of the method in detecting anomalous time series. They suggest to extend their method on multivariate time series in future works.

In contrast, [2] propose a method combining matrix profiles with shapelet learning to handle streaming time series data. The matrix profile efficiently identifies potential anomalies in real-time, and shapelet learning characterizes these anomalies. This approach is particularly suited for environments requiring immediate AD. The authors provide code examples for implementation in their paper but further open source repositories were not found. As the previous work they didn't use the method for multivariate time series and suggest expanding the method for this.

The IPS (Instance Profile for Shapelet Discovery) method addresses time series classification by using shapelets-discriminative subsequences of time series. This method was tested extensively on various datasets but was not explicitly tested in a zero-shot scenario [31].

The method proposed in [34] focuses on using shapelet-based representations for unsupervised learning of multivariate time series. It employs CL, multi-scale alignment, and a shapelet transformer. The method was tested in scenarios where labeled data is sparse or unavailable, but it does not appear to be tested explicitly in a traditional zero-shot setting [23].

Combinations [32] propose a method for detecting anomalies in multivariate time series by combining clustering and reconstruction techniques. The authors use a sliding window approach to generate subsequences from the multivariate time series, then apply extended fuzzy clustering to reveal the underlying structure of the subsequences. By reconstructing these subsequences with optimal cluster centers, the method detects anomalies based on how well the reconstructed data fits the original subsequences. A confidence index quantifies the level of detected anomalies. The method does not seem to have been explicitly tested in a ZSL scenario.

A Fourier transformation isolates seasonal components in a method called Moving Memory Dynamic Filter (MMDF). While the distance transformation captures both the data values and their temporal dependencies. Anomalies are detected when the center-to-center distance exceeds the threshold. The authors primarily discuss the proposed method in the context of univariate time series AD and do not explicitly mention its applicability to MVTSD [11].

In contrast, the method DCFF-MTAD, a multivariate time series AD method uses a dual-channel feature extraction that combines spatial short-time Fourier transformation for spatial features and a graph attention network for temporal features. These features are then fused using a Gated Recurrent Unit for robust AD. The method is not tested on ZSL and is not available as open source [67].

5 Application on Zero-Shot Anomaly Detection in Multivariate Time Series

The found articles are filtered using the exclusion criteria defined in subsection 3.2. By excluding methods that haven't been tested with multiple input variables we answer RQ2. Methods that are not tested in Zero-Shot Learning Scenarios are also excluded which covers RQ4. In order to achieve a successful implementation in the following chapter only models which are publicly available and well documented are chosen for further examination.

[23] github.com/real2fish/CSL

More precisely we want to know by the defined filter process which of the proposed RL types are best suited for Zero Shot AD in multi-variate time series data. A selection of appropriate methods for MVTSD AD out of section 4 is extracted.

All found representation learning methodologies from section 4 are listed in Table 1. Their evaluation on the exclusion criteria is marked and a short description on the underlying concepts is given.

6 Proof of Concept

The best fitting strategies extracted in section 5 are implemented on a small test data set in order to demonstrate how and if they work. First the used dataset ant the purpose of AD for the specific use case is described. Later the process of implemention and the results are presented.

6.1 Inverter data including Anomalies

While NLP and image processing tasks are common and a variety of data sets exists, time series data sets are not available that much [35]. Thanks to the employees of SMA a multivariate time series dataset is provided. The specific use case and the structure of the chosen data is described in this section.

SMA developes and manufactures inverters for home and commercial use. The inverters convert direct to alternating current or vice versa depending on the use case. They also act as home managers controlling all energy flows in a power plant. Inverters are equipped with several sensors measuring the surroundings and internal states in order to maximize the efficency and to avoid system failures. Sometimes system failures appear still which raises the question if this could have been foreseen by analyzing the gathered sensor data during runtime. The first step to reach such a forecasting tool is to detect the anomalies in recorded sensor data. The implementation of the chosen methods is therefore done on inverter data provided by the company. The dataset is initially presented in [41].

The variables contain measurements of current and voltage of all phases in AC and the generated DC sources. Additionally temperatures, CPU usage, internal parameter settings and many other measurements are included. The total number of features is 137.

The values are collectively stored at a 7 minute interval over several months. 19 inverters that had system failures at some point are taken into account. These failure time stamps are known and added as an additional feature with a binary value of 1. Every other error bit is 0. The data is collected between 2018 and 2020. The locations of the inverters cannot be provided.

6.2 Implemented Methods

The main goal is to detect the labeled anomalies in the SMA dataset. If the methods find the timestamp of the system failure, they perform correctly. First

Table 1. Abbreviations: Transformer (T), Clustering (C), multiple input variables (MV), open source availability (OSA). Legend: yes: ✓, no: ✗

Method Name	Author	Concepts	MV	ZSL	OSA
INRAD	Jeong et al.	MLP	✓	✗	✓
OmniAnomaly	Su et al.	RNN	✓	✗	✓
CNN based method	Kravchik et al.	CNN	✓	✗	✗
CNN based method	He et al.	CNN	✓	✗	✗
SLMR	Miao et al.	CNN	✓	✗	✗
Debiased CL	Zhang et al.	CL	✓	✓	✗
CARLA	Darban et al.	CL	✓	✓	✓
CL-TAD	Ngu et al.	CL	✓	✗	✓
COCA	Wang et al.	CL, OCC	✓	✗	✓
CL based method	Lee et al.	CL, OCC	✓	✓	✗
CL based method	Chen et al.	CL	✓	✗	✗
TS2Vec	Yue et al.	CL	✓	✓	✓
TimeAutoAD	Jiao et al.	CL	✓	✗	✗
ContrastAD	Li et al.	CL	✓	✗	✗
Dddetector	Yang et al.	CL	✓	✗	✓
TRL-CPC	Pranavan et al.	CPC	✓	✗	✗
TiCTok	Kang et al.	CPC	✓	✗	✗
MGCLAD	Qin et al.	CL	✓	✗	✓
TriAD	Sun et al.	CL	✗	✗	✓
AE based method	Provotar et al.	AE, LSTM	✓	✗	✗
LATAM	Nivarthi et al.	AE,LSTM	✓	✗	✓
LSTM-based VAE-GAN	Niu et al.	AE, LSTM	✓	✗	✗
TSMAE	Gao et al.	AE, LSTM	✓	✗	✗
VRAE based method	Pereira et al.	VRAE	✓	✓	✗
AE based method	Ramirez et al.	VRAE	✓	✓	✗
Bi-VRQRAE	Kieu et al.	VRAE	✓	✗	✓
FuSAGNet	Han et al.	AE, GNN	✓	✗	✓
MSTVAE	Pham et al.	VAE, CNN	✓	✗	✓
deep AOC	Mou et al.	AE, OCC	✓	✗	✓
AE based method	Zhang et al.	AE	✗	✓	✗
RANSynCoders	Abdulaal et al.	AE	✓	✗	✓
TCN-AE	Thill et al.	AE, TCN	✓	✓	✗
MEGA	Wang et al.	AE, GCN	✓	✓	✓
DAEMON	Chen et al.	AE	✓	✓	✓
GRU-AE	Gong et al.	AE, GRU	✓	✗	✗
MSCVAE	Yokkampon et al.	VAE	✓	✓	✗
AE based method	Wang et al.	AE	✓	✓	✗
MOMENT	Goswami et al.	AE	✓	✓	✓
TranAD	Tuli et al.	T	✓	✗	✓
AnomalyTransformer	Xu et al.	T	✓	✗	✓
TiSAT	Doshi et al.	T	✓	✗	✓
TCF-Trans	Peng et al.	T	✓	✗	✗
DCT-GAN	Li et al.	T, GAN	✓	✗	✗
AnoFormer	Shin et al.	T, GAN	✓	✗	✗
LLM based method	Li et al.	T, LLM	✗	✓	✗
DATN	Wu et al.	T, LLM	✓	✗	✗
Transformer based method	Ye et al.	T, Fourier	✓	✗	✗
SL based method	Beggel et al.	SL	✗	✗	✗
SL based method	Alshaer et al.	SL	✗	✗	✗
IPS	Li et al.	SL	✗	✗	✗
CSL	Liang et al.	SL, CL, T	✓	✗	✓
Clustering based	Li et al.	C	✓	✗	✗
MMDF	Duan et al.	MMDF	✗	✗	✗
DCFF-MTAD	Xu et al.	GAN, Fourier	✓	✗	✗

the models are taken as is and a forward pass is applied. The given output is evaluated if the model already detects the failure time points in the inverter time series data. If possible, the model is trained afterwards using a set of 12 inverters and tested on the remaining 7 inverters. This way the Zero-Shot scenario like in section 2 is created. The code for the implementions can be found on link [24].

CARLA An implemention of the model CARLA is done by the authors using Python available on footnote 3. The repository contains scripts for replicating the results, but no further information on how to adapt the model on new datasets is given.

TS2Vec An implemention of the model TS2Vec is done by the authors using Python which is available in footnote 6. A function for preprocessing the data to fit as an input for the model is available in footnote 24. Using the function a successful forward pass was made. The learned representations can be used for further evaluation.

MEGA According to [63] the model MEGA is trained and evaluated on three different datasets. The code for reproduction is provided by the authors. They implement the model using Python on footnote 17. No further information on how to adapt the model on new datasets is given but the scripts are adapted to run successfully with SMA data as multivariate input. The output was not further reviewed.

DAEMON The authors of the method DAEMON published python scripts for replicating the results on all datasets they used [8]. However, no explanation on how to use the method for entirely new datasets can be found in footnote 18. Functions for data preprocessing and a forward pass throught the model are written.

MOMENT A detailed instruction on how to reproduce the outcomes of MOMENT presented by [16] is provided on this GitHub footnote 19. The method takes a three dimensional input tensor, containing batches, channels and time points. The maximal number of points is 512. It reconstructs the given input and returns a tensor containing the reconstructions.

For the implementation on SMA data the provided tutorial is used as a guide. The SMA data is preprocessed to match the requested input tensor. The reconstructions are further compared with the inputs by calulating the Mean Squared Error (MSE):

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1}$$

[24] github.com/johanneshoelker/Smart-Systems-Paper/tree/main/Implementation

The error is further normalized using the Min-Max Normalization:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{2}$$

Every timestamp that exceeds a certain threshold then represents the found anomalies.

One sensor needed to be excluded because the values didn't change. The threshold was decreased until 0.005 but with an untrained model no anomalies are found.

7 Discussion and Future Work

The presented paper has a few limitations which are discussed in this chapter. Additionally ideas for further research are given.

Some of the found methods are not tested in Zero-Shot Scenarios in the article they are presented. The authors didn't focus on Zero-Shot Learning and didn't use appropriate datasets for this use case. This doesn't mean they can not perform well on ZSL. Further research in generating ZSL scenarios with benchmark datasets need to be done with these methods.

In comparison the adaptability of univariate methods for multivariate data is restricted. If a method designed for time series data with a single input variable needs to be used for multiple input variables a redesign is necessary. This can be extensive or rather simple depending on the architecture. To know how much redesign is necessary every method has to be reviewed in further research. However, nearly all of the found methods are trained and tested with MVTSD.

Some models like MOMENT are handling input variables seperately. The interconnection between the different channels is not considered directly. The influence of one variable on the other can provide informations that may be important for AD and the learned representations in general. A successful approach including the correlations between multiple input variables can be found in [47], [63] and [19].

Another important evaluation point is the selection of datasets. The transferability between time series datasets is difficult due to the fact that the data between domains is huge [35]. Time series can have a similar shape across different domains but that does not have to be true in general. For a good model fitting to the specfic context, it should be trained on similar data. This way robust representations can be learned that hold information about the specific domain.

The model selection in this paper could have been more systematically. Several models were excluded that could possibly be adapted to the presented use case. A model selection process with generating synthetic anomalies simplifies the search for an appropriate dataset. Such a model selection process for zero shot AD is presented in [12].

Not only the methods available as open source should be implemented and tested. Methods without a code example provided by the authors can be implemented based on the architecture presented in the paper. This needs to be done in future development.

Testing pretrained models that are not trained on the domain of inverter data is an interesting attempt, which could potentially have an advantageous outcome. This way domains that have no data available can be tested on anomalies. However in this case it is not ensured to find all anomalies successfully. For robust representations that contain information about the data, they need to be trained and finetuned for the specfifc use case in future works.

During the research on how to use the methods for entirely new datasets, several difficulties occured. The code provided by the authors was mainly provided for replicating the found results. Given scripts took the main benchmark datasets that were mentioned in the paper as an input. Preparing entirely new datasets in order to train and use the models was tricky because of the missing knowledge of their software architecture. Providing additional data preprocessing guidelines for generalization is necessary for further development. However, preprocessing SMA inverter data for some methods and a forward pass through the models was successful. This work can be used for further development.

Additionally the implemention in this paper is done without further analysis of the results. The correctness of the detection can be rated and compared in future works.

References

1. Abdulaal, A.: Practical Approach to Asynchronous Multivariate Time Series Anomaly Detection and Localization (2021)
2. Alshaer, M., Garcia-Rodriguez, S., Gouy-Pailler, C.: Detecting Anomalies from Streaming Time Series using Matrix Profile and Shapelets Learning. In: 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI). pp. 376–383. IEEE, Baltimore, MD, USA (Nov 2020). https://doi.org/10.1109/ICTAI50040.2020.00066, https://ieeexplore.ieee.org/document/9288261/
3. Aota, T., Tong, L.T.T., Okatani, T.: Zero-shot versus Many-shot: Unsupervised Texture Anomaly Detection. In: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 5553–5561. IEEE, Waikoloa, HI, USA (Jan 2023). https://doi.org/10.1109/WACV56688.2023.00552, https://ieeexplore.ieee.org/document/10030870/
4. Beggel, L., Kausler, B.X., Schiegg, M., Pfeiffer, M., Bischl, B.: Time series anomaly detection based on shapelet learning. Computational Statistics **34**(3), 945–976 (Sep 2019). https://doi.org/10.1007/s00180-018-0824-9, http://link.springer.com/10.1007/s00180-018-0824-9
5. Bengio, Y., Courville, A., Vincent, P.: Representation Learning: A Review and New Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(8), 1798–1828 (Aug 2013). https://doi.org/10.1109/TPAMI.2013.50, http://ieeexplore.ieee.org/document/6472238/
6. Bishop, C.M.: Pattern recognition and machine learning. Information science and statistics, Springer, New York (2006)
7. Chen, K., Feng, M., Wirjanto, T.S.: Time-series Anomaly Detection via Contextual Discriminative Contrastive Learning (Apr 2023), http://arxiv.org/abs/2304.07898, arXiv:2304.07898 [cs]
8. Chen, X., Deng, L., Zhao, Y., Zheng, K.: Adversarial Autoencoder for Unsupervised Time Series Anomaly Detection and Interpretation. In: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining. pp. 267–275. ACM, Singapore Singapore (Feb 2023). https://doi.org/10.1145/3539597.3570371, https://dl.acm.org/doi/10.1145/3539597.3570371
9. Darban, Z.Z., Webb, G.I., Pan, S., Aggarwal, C.C., Salehi, M.: CARLA: Self-supervised Contrastive Representation Learning for Time Series Anomaly Detection (Apr 2024), http://arxiv.org/abs/2308.09296, arXiv:2308.09296 [cs]
10. Doshi, K., Abudalou, S., Yilmaz, Y.: TiSAT: Time Series Anomaly Transformer (Mar 2022), http://arxiv.org/abs/2203.05167, arXiv:2203.05167 [cs, eess, stat]
11. Duan, J., Xu, X., Wang, Y.: Unsupervised Time Series Anomaly Detection using Moving Memorial Dynamic Filter. In: 2021 40th Chinese Control Conference (CCC). pp. 3403–3408. IEEE, Shanghai, China (Jul 2021). https://doi.org/10.23919/CCC52363.2021.9549428, https://ieeexplore.ieee.org/document/9549428/
12. Fung, C., Qiu, C., Li, A., Rudolph, M.: Model Selection of Zero-shot Anomaly Detectors in the Absence of Labeled Validation Data (Feb 2024), http://arxiv.org/abs/2310.10461, arXiv:2310.10461 [cs]
13. Gao, H., Qiu, B., Barroso, R.J.D., Hussain, W., Xu, Y., Wang, X.: TSMAE: A Novel Anomaly Detection Approach for Internet of Things Time Series Data Using Memory-Augmented Autoencoder. IEEE Transactions on Network Science and Engineering **10**(5), 2978–2990 (Sep 2023). https://doi.org/10.1109/TNSE.2022.3163144, https://ieeexplore.ieee.

org/document/9744555/?arnumber=9744555, conference Name: IEEE Transactions on Network Science and Engineering

14. Gong, X., Liao, S., Hu, F., Hu, X., Liu, C.: Autoencoder-Based Anomaly Detection for Time Series Data in Complex Systems. In: 2022 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS). pp. 428–433 (Nov 2022). https://doi.org/10.1109/APCCAS55924.2022.10090260, https://ieeexplore.ieee.org/document/10090260/?arnumber=10090260
15. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. Adaptive computation and machine learning, The MIT Press, Cambridge, Massachusetts (2016)
16. Goswami, M., Szafer, K., Choudhry, A., Cai, Y., Li, S., Dubrawski, A.: MOMENT: A Family of Open Time-series Foundation Models (May 2024), http://arxiv.org/abs/2402.03885, arXiv:2402.03885 [cs]
17. Gruhl, C.M.: Novelty Detection for Multivariate Data Streams with Probalistic Models (2022). https://doi.org/10.17170/KOBRA-202205106160, https://kobra.uni-kassel.de/handle/123456789/13902, publisher: Universität Kassel
18. Gärdenfors, P.: Conceptual spaces: the geometry of thought. MIT Press, Cambridge, Mass (2000)
19. Han, S., Woo, S.S.: Learning Sparse Latent Graph Representations for Anomaly Detection in Multivariate Time Series. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 2977–2986. ACM, Washington DC USA (Aug 2022). https://doi.org/10.1145/3534678.3539117, https://dl.acm.org/doi/10.1145/3534678.3539117
20. He, Y., Zhao, J.: Temporal Convolutional Networks for Anomaly Detection in Time Series. Journal of Physics (2019)
21. Jeong, K.J., Shin, Y.M.: Time-Series Anomaly Detection with Implicit Neural Representation (Jan 2022), http://arxiv.org/abs/2201.11950, arXiv:2201.11950 [cs]
22. Jiao, Y., Yang, K., Song, D., Tao, D.: TimeAutoAD: Autonomous Anomaly Detection With Self-Supervised Contrastive Loss for Multivariate Time Series. IEEE Transactions on Network Science and Engineering **9**(3), 1604–1619 (May 2022). https://doi.org/10.1109/TNSE.2022.3148276, https://ieeexplore.ieee.org/document/9705079/
23. Kang, M., Lee, B.: TiCTok: Time-Series Anomaly Detection With Contrastive Tokenization. IEEE Access **11**, 81011–81020 (2023). https://doi.org/10.1109/ACCESS.2023.3301140, https://ieeexplore.ieee.org/document/10201844/
24. Kieu, T., Yang, B., Guo, C., Cirstea, R.G., Zhao, Y., Song, Y., Jensen, C.S.: Anomaly Detection in Time Series with Robust Variational Quasi-Recurrent Autoencoders. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE). pp. 1342–1354 (May 2022). https://doi.org/10.1109/ICDE53745.2022.00105, https://ieeexplore.ieee.org/document/9835268/?arnumber=9835268, iSSN: 2375-026X
25. Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., Linkman, S.: Systematic literature reviews in software engineering – A systematic literature review. Information and Software Technology **51**(1), 7–15 (Jan 2009). https://doi.org/10.1016/j.infsof.2008.09.009, https://linkinghub.elsevier.com/retrieve/pii/S0950584908001390
26. Kravchik, M., Shabtai, A.: Detecting Cyberattacks in Industrial Control Systems Using Convolutional Neural Networks (Dec 2018), http://arxiv.org/abs/1806.08110, arXiv:1806.08110 [cs]
27. Lavrač, N., Podpečan, V., Robnik-Šikonja, M.: Representation Learning: Propositionalization and Embeddings. Springer International Publishing, Cham (2021).

https://doi.org/10.1007/978-3-030-68817-2, https://link.springer.com/10.1007/978-3-030-68817-2
28. Lee, Y., Byun, Y., Baek, J.G.: Time Series Anomaly Detection Using Contrastive Learning based One-Class Classification. In: 2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC). pp. 330–335. IEEE, Bali, Indonesia (Feb 2023). https://doi.org/10.1109/ICAIIC57133.2023.10067089, https://ieeexplore.ieee.org/document/10067089/
29. Li, A., Zhao, Y., Qiu, C., Kloft, M., Smyth, P., Rudolph, M., Mandt, S.: Anomaly Detection of Tabular Data Using LLMs (Jun 2024), http://arxiv.org/abs/2406.16308, arXiv:2406.16308 [cs]
30. Li, B., Müller, E.: Contrastive Time Series Anomaly Detection by Temporal Transformations. In: 2023 International Joint Conference on Neural Networks (IJCNN). pp. 1–8 (Jun 2023). https://doi.org/10.1109/IJCNN54540.2023.10191358, https://ieeexplore.ieee.org/document/10191358/, iSSN: 2161-4407
31. Li, G., Choi, B., Xu, J., Bhowmick, S.S., Mah, D.N.y., Wong, G.L.: IPS: Instance Profile for Shapelet Discovery for Time Series Classification. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE). pp. 1781–1793 (May 2022). https://doi.org/10.1109/ICDE53745.2022.00179, https://ieeexplore.ieee.org/document/9835498/?arnumber=9835498, iSSN: 2375-026X
32. Li, J., Izakian, H., Pedrycz, W., Jamal, I.: Clustering-based anomaly detection in multivariate time series data. Applied Soft Computing **100**, 106919 (Mar 2021). https://doi.org/10.1016/j.asoc.2020.106919, https://linkinghub.elsevier.com/retrieve/pii/S1568494620308577
33. Li, Y., Peng, X., Zhang, J., Li, Z., Wen, M.: DCT-GAN: Dilated Convolutional Transformer-Based GAN for Time Series Anomaly Detection. IEEE Transactions on Knowledge and Data Engineering **35**(4), 3632–3644 (Apr 2023). https://doi.org/10.1109/TKDE.2021.3130234, https://ieeexplore.ieee.org/document/9626552/?arnumber=9626552, conference Name: IEEE Transactions on Knowledge and Data Engineering
34. Liang, Z., Zhang, J., Liang, C., Wang, H., Liang, Z., Pan, L.: A Shapelet-based Framework for Unsupervised Multivariate Time Series Representation Learning (Aug 2024). https://doi.org/10.14778/3632093.3632103, http://arxiv.org/abs/2305.18888, arXiv:2305.18888 [cs]
35. Ma, Q., Liu, Z., Zheng, Z., Huang, Z., Zhu, S., Yu, Z., Kwok, J.T.: A Survey on Time-Series Pre-Trained Models (May 2023), http://arxiv.org/abs/2305.10716, arXiv:2305.10716 [cs]
36. Miao, Q., Xu, C., Zhan, J., Zhu, D., Wu, C.: An Unsupervised Short- and Long-Term Mask Representation for Multivariate Time Series Anomaly Detection (Aug 2022), http://arxiv.org/abs/2208.09240, arXiv:2208.09240 [cs]
37. Mou, X., Wang, R., Wang, T., Sun, J., Li, B., Wo, T., Liu, X.: Deep Autoencoding One-Class time Series Anomaly Detection. In: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5 (Jun 2023). https://doi.org/10.1109/ICASSP49357.2023.10095724, https://ieeexplore.ieee.org/document/10095724/?arnumber=10095724, iSSN: 2379-190X
38. Ngu, H.C.V., Lee, K.M.: CL-TAD: A Contrastive-Learning-Based Method for Time Series Anomaly Detection. Applied Sciences **13**(21), 11938 (Oct 2023). https://doi.org/10.3390/app132111938, https://www.mdpi.com/2076-3417/13/21/11938
39. Nielsen, M.: Neural Networks and Deep Learning p. 224 (2015)

40. Niu, Z., Yu, K., Wu, X.: LSTM-Based VAE-GAN for Time-Series Anomaly Detection. Sensors **20**(13), 3738 (Jul 2020). https://doi.org/10.3390/s20133738, https://www.mdpi.com/1424-8220/20/13/3738
41. Nivarthi, C.P., Sick, B.: Towards Few-Shot Time Series Anomaly Detection with Temporal Attention and Dynamic Thresholding. In: 2023 International Conference on Machine Learning and Applications (ICMLA). pp. 1444–1450. IEEE, Jacksonville, FL, USA (Dec 2023). https://doi.org/10.1109/ICMLA58977.2023.00218, https://ieeexplore.ieee.org/document/10459893/
42. Nivarthi, C.P., Vogt, S., Sick, B.: Unified Autoencoder with Task Embeddings for Multi-Task Learning in Renewable Power Forecasting. In: 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA). pp. 1530–1536. IEEE, Nassau, Bahamas (Dec 2022). https://doi.org/10.1109/ICMLA55696.2022.00240, https://ieeexplore.ieee.org/document/10068974/
43. Palatucci, M., Pomerleau, D., Hinton, G.E., Mitchell, T.M.: Zero-shot Learning with Semantic Output Codes (2009)
44. Peng, X., Li, H., Lin, Y., Chen, Y., Fan, P., Lin, Z.: TCF-Trans: Temporal Context Fusion Transformer for Anomaly Detection in Time Series. Sensors **23**(20), 8508 (Oct 2023). https://doi.org/10.3390/s23208508, https://www.mdpi.com/1424-8220/23/20/8508
45. Pereira, J., Silveira, M.: Learning Representations from Healthcare Time Series Data for Unsupervised Anomaly Detection. In: 2019 IEEE International Conference on Big Data and Smart Computing (BigComp). pp. 1–7. IEEE, Kyoto, Japan (Feb 2019). https://doi.org/10.1109/BIGCOMP.2019.8679157, https://ieeexplore.ieee.org/document/8679157/
46. Pham, T.A., Lee, J.H., Park, C.S.: MST-VAE: Multi-Scale Temporal Variational Autoencoder for Anomaly Detection in Multivariate Time Series. Applied Sciences **12**(19), 10078 (Oct 2022). https://doi.org/10.3390/app121910078, https://www.mdpi.com/2076-3417/12/19/10078
47. Pranavan, T., Sim, T., Ambikapathi, A., Ramasamy, S.: Contrastive predictive coding for Anomaly Detection in Multi-variate Time Series Data (Feb 2022), http://arxiv.org/abs/2202.03639, arXiv:2202.03639 [cs]
48. Provotar, O.I., Linder, Y.M., Veres, M.M.: Unsupervised Anomaly Detection in Time Series Using LSTM-Based Autoencoders. In: 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT). pp. 513–517 (Dec 2019). https://doi.org/10.1109/ATIT49449.2019.9030505, https://ieeexplore.ieee.org/document/9030505/?arnumber=9030505
49. Qin, S., Chen, L., Luo, Y., Tao, G.: Multiview Graph Contrastive Learning for Multivariate Time-Series Anomaly Detection in IoT. IEEE Internet of Things Journal **10**(24), 22401–22414 (Dec 2023). https://doi.org/10.1109/JIOT.2023.3303946, https://ieeexplore.ieee.org/document/10214266/
50. Ramirez Rivera, A., Khan, A., Bekkouch, I.E.I., Sheikh, T.S.: Anomaly Detection Based on Zero-Shot Outlier Synthesis and Hierarchical Feature Distillation. IEEE Transactions on Neural Networks and Learning Systems **33**(1), 281–291 (Jan 2022). https://doi.org/10.1109/TNNLS.2020.3027667, https://ieeexplore.ieee.org/document/9228891/
51. Sabokrou, M.: Deep-anomaly_ Fully convolutional neural network for fast anomaly detection in crowded scenes. Computer Vision and Image Understanding (2018)
52. Schwartz, E., Arbelle, A., Karlinsky, L., Harary, S., Scheidegger, F., Doveh, S., Giryes, R.: MAEDAY: MAE for few and zero shot AnomalY-Detection (Feb 2024), http://arxiv.org/abs/2211.14307, arXiv:2211.14307 [cs]

53. Shi, Z., Chen, J., Li, K., Raghuram, J., Wu, X., Liang, Y., Jha, S.: The Trade-off between Universality and Label Efficiency of Representations from Contrastive Learning (Feb 2023), http://arxiv.org/abs/2303.00106, arXiv:2303.00106 [cs]
54. Shin, A.H., Kim, S.T., Park, G.M.: Time Series Anomaly Detection Using Transformer-Based GAN With Two-Step Masking. IEEE Access **11**, 74035–74047 (2023). https://doi.org/10.1109/ACCESS.2023.3289921, https://ieeexplore.ieee.org/document/10164104/?arnumber=10164104, conference Name: IEEE Access
55. Socher, R., Ganjoo, M., Sridhar, H., Bastani, O., Manning, C.D., Ng, A.Y.: Zero-Shot Learning Through Cross-Modal Transfer (Mar 2013), http://arxiv.org/abs/1301.3666, arXiv:1301.3666 [cs]
56. Su, J., Jiang, C., Jin, X., Qiao, Y., Xiao, T., Ma, H., Wei, R., Jing, Z., Xu, J., Lin, J.: Large Language Models for Forecasting and Anomaly Detection: A Systematic Literature Review (Feb 2024), http://arxiv.org/abs/2402.10350, arXiv:2402.10350 [cs]
57. Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., Pei, D.: Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2828–2837. ACM, Anchorage AK USA (Jul 2019). https://doi.org/10.1145/3292500.3330672, https://dl.acm.org/doi/10.1145/3292500.3330672
58. Sun, Y., Pang, G., Ye, G., Chen, T., Hu, X., Yin, H.: Unraveling the "Anomaly" in Time Series Anomaly Detection: A Self-supervised Tri-domain Solution (Nov 2023), http://arxiv.org/abs/2311.11235, arXiv:2311.11235 [cs]
59. Thill, M., Konen, W., Wang, H., Bäck, T.: Temporal convolutional autoencoder for unsupervised anomaly detection in time series. Applied Soft Computing **112**, 107751 (Nov 2021). https://doi.org/10.1016/j.asoc.2021.107751, https://linkinghub.elsevier.com/retrieve/pii/S1568494621006724
60. Tuli, S., Casale, G., Jennings, N.R.: TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data (May 2022), http://arxiv.org/abs/2201.07284, arXiv:2201.07284 [cs]
61. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is All you Need (2017)
62. Wang, B., Tsung, F., Yan, H.: Attention-based Representation Learning for Time Series with Principal and Residual Space Monitoring. In: 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE). pp. 1833–1839. IEEE, Mexico City, Mexico (Aug 2022). https://doi.org/10.1109/CASE49997.2022.9926721, https://ieeexplore.ieee.org/document/9926721/
63. Wang, J., Shao, S., Bai, Y., Deng, J., Lin, Y.: Multiscale Wavelet Graph AutoEncoder for Multivariate Time-Series Anomaly Detection. IEEE Transactions on Instrumentation and Measurement **72**, 1–11 (2023). https://doi.org/10.1109/TIM.2022.3223142, https://ieeexplore.ieee.org/document/9954430/?arnumber=9954430, conference Name: IEEE Transactions on Instrumentation and Measurement
64. Wang, R., Liu, C., Mou, X., Gao, K., Guo, X., Liu, P., Wo, T., Liu, X.: Deep Contrastive One-Class Time Series Anomaly Detection (Apr 2023), http://arxiv.org/abs/2207.01472, arXiv:2207.01472 [cs]
65. Wu, B., Fang, C., Yao, Z., Tu, Y., Chen, Y.: Decompose Auto-Transformer Time Series Anomaly Detection for Network Management. Electronics **12**(2), 354 (Jan 2023). https://doi.org/10.3390/electronics12020354, https://www.mdpi.com/2079-9292/12/2/354

66. Xu, J., Wu, H., Wang, J., Long, M.: Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy (Jun 2022), http://arxiv.org/abs/2110.02642, arXiv:2110.02642 [cs]
67. Xu, Z., Yang, Y., Gao, X., Hu, M.: DCFF-MTAD: A Multivariate Time-Series Anomaly Detection Model Based on Dual-Channel Feature Fusion. Sensors **23**(8), 3910 (Apr 2023). https://doi.org/10.3390/s23083910, https://www.mdpi.com/1424-8220/23/8/3910
68. Yang, Y., Zhang, C., Zhou, T., Wen, Q., Sun, L.: DCdetector: Dual Attention Contrastive Representation Learning for Time Series Anomaly Detection. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 3033–3045 (Aug 2023). https://doi.org/10.1145/3580305.3599295, http://arxiv.org/abs/2306.10347, arXiv:2306.10347 [cs]
69. Ye, Y., He, Q., Zhang, P., Xiao, J., Li, Z.: Multivariate Time Series Anomaly Detection with Fourier Time Series Transformer (2023)
70. Yokkampon, U., Mowshowitz, A., Chumkamon, S., Hayashi, E.: Robust Unsupervised Anomaly Detection With Variational Autoencoder in Multivariate Time Series Data. IEEE Access **10**, 57835–57849 (2022). https://doi.org/10.1109/ACCESS.2022.3178592, https://ieeexplore.ieee.org/document/9783083/
71. Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y., Xu, B.: TS2Vec: Towards Universal Representation of Time Series (Feb 2022), http://arxiv.org/abs/2106.10466, arXiv:2106.10466 [cs]
72. Zhang, K., Cai, R., Zhou, C., Liu, Y.: Debiased Contrastive Learning for Time-Series Representation Learning and Fault Detection. IEEE Transactions on Industrial Informatics **20**(5), 7641–7653 (May 2024). https://doi.org/10.1109/TII.2024.3359409, https://ieeexplore.ieee.org/document/10443248/
73. Zhang, Z., Liu, Q., Qiu, S., Zhou, S., Zhang, C.: Unknown Attack Detection Based on Zero-Shot Learning. IEEE Access **8**, 193981–193991 (2020). https://doi.org/10.1109/ACCESS.2020.3033494, https://ieeexplore.ieee.org/document/9239385/

8 Originality Statement

I hereby declare that the content of this paper is written on my own and sources from literature are declared as such.

Kassel, 23.09.2024

9 AI Assistance Statement

Artificial intelligence is used for learning the foundations to write this paper. During literature research the GPT "Consensus" helped in finding and summarizing relevant papers. The output was reviewed and checked for correctness. The AI-based translator "DeepL" was used to formulate sentences and to find a proper wording.

Kassel, 23.09.2024