IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 11 February 2024, accepted 26 February 2024, date of publication 29 February 2024, date of current version 7 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3371891

APPLIED RESEARCH

Detecting Anomalies in Time Series Using Kernel Density Approaches

ROBIN FREHNER[1], KESHENG WU[2], (Senior Member, IEEE), ALEXANDER SIM[2], (Senior Member, IEEE), JINOH KIM[3], (Senior Member, IEEE), AND KURT STOCKINGER[1]

[1]School of Engineering, Zurich University of Applied Sciences, 8401 Winterthur, Switzerland
[2]Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
[3]Department of Computer Science and Information Systems, Texas A&M University–Commerce, Commerce, TX 75428, USA

Corresponding author: Robin Frehner (frehner.robin@gmail.com)

This work was supported in part by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-05CH11231; in part by the National Energy Research Scientific Computing Center (NERSC); and in part by the School of Engineering, Zurich University of Applied Sciences.

ABSTRACT This paper introduces a novel anomaly detection approach tailored for time series data with exclusive reliance on normal events during training. Our key innovation lies in the application of kernel-density estimation (KDE) to scrutinize reconstruction errors, providing an empirically derived probability distribution for normal events post-reconstruction. This non-parametric density estimation technique offers a nuanced understanding of anomaly detection, differentiating it from prevalent threshold-based mechanisms in existing methodologies. In post-training, events are encoded, decoded, and evaluated against the estimated density, providing a comprehensive notion of normality. In addition, we propose a data augmentation strategy involving variational autoencoder-generated events and a smoothing step for enhanced model robustness. The significance of our autoencoder-based approach is evident in its capacity to learn normal representation without prior anomaly knowledge. Through the KDE step on reconstruction errors, our method addresses the versatility of anomalies, departing from assumptions tied to larger reconstruction errors for anomalous events. Our proposed likelihood measure then distinguishes normal from anomalous events, providing a concise yet comprehensive anomaly detection solution. The extensive experimental results support the feasibility of our proposed method, yielding significantly improved classification performance by nearly 10% on the UCR benchmark data.

INDEX TERMS Time series anomaly detection, machine learning, neural networks, autoencoder, kernel density estimation.

I. INTRODUCTION

Anomaly detection in time series is a crucial task in many applications including network monitoring, medical diagnosis, financial fraud and database applications detection [1], [2], [5], [6], [7]. One key challenge in anomaly detection is the scarcity of the anomalies [1], [8], [9]. Many well-known techniques for differentiating one type of events from another would not be effective when there is a significant imbalance between different types of events [2]. Among the techniques specifically developed for anomaly detection, there is no clear winner [2], [10], [11].

This study is concerned with developing an effective machine learning-based approach for anomaly detection in time series data, with a focus on the modeling of normal behavior. We explore a reconstruction-based approach that incorporates several elements of common approaches, including data augmentation, encoding, distance metrics, and statistical analysis [12]. Our goal is to develop a model that can accurately distinguish between normal and malicious events by measuring the degree of deviation from the learned normal behavior. In addition, we investigate techniques

The associate editor coordinating the review of this manuscript and approving it for publication was Dost Muhammad Khan.

for enhancing the performance of the model using data augmentation. Overall, our study aims to contribute to the field of anomaly detection by providing insights into the development of effective machine learning-based approaches for capturing normal behavior and detecting anomalies in time series data, demonstrating its effectiveness with a set of challenging benchmark data.

The reconstruction-based approach we use employs a machine learning technique known as autoencoder [3]. This autoencoder reduces the dimension of the input time series (typically in short subsequences) and then decodes the compressed representation to produce a model output. The difference between the model output and the original input is measured to identify anomalies. This encoding, decoding, and identification approach is very flexible and could incorporate many techniques from other anomaly detection approaches. We are interested in this approach because it has an important feature for addressing challenging anomaly detection problems where only normal events are available for training, which is an extreme form of scarcity of anomalies. In such a scenario, anomaly detection approaches often rely heavily on a time-consuming technique known as *data augmentation* to increase the diversity of the training data and improve the model's ability to generalize to unseen data [13], [14], [15].

The reconstruction-based approach is appropriate for the recent benchmark from University of California, Riverside (UCR) [16], where the training samples are all normal events and anomalies are only present in the testing set. This UCR benchmark is particularly challenging by construction and is a good testing data set for anomaly detection algorithms.

Overall, the main contributions of this paper can be outlined as follows:

1) We propose a novel approach that *classifies anomalies based on kernel-density estimation* to replace the commonly used thresholding approach. The kernel-density estimation is applied on the reconstruction errors to produce a distribution of training errors, which is then used to produce a score for measuring the normality of an event using the estimated density.
2) We present a *data augmentation approach based on a variational autoencoder*. This autoencoder is trained on anomalous samples to augment data by encoding and projecting them onto the latent space. The generated samples are then smoothed in the augmentation step to remove outliers causing spikes.
3) We share the evaluation results and our observations made from the extensive experiments. Detailed experiments on the UCR benchmark data show that our *distribution-based classification approach improves classification accuracy by nearly 10% and makes certain expensive data augmentation unnecessary*.

The paper is structured as follows. In Section II, we discuss related work. In Section III, we provide an overview of the taxonomy of anomalies in time series and various anomaly detection paradigms, while also establishing the necessary background information. The aforementioned problems are further examined in Section IV within the context of reconstruction-based methods. In Section V, the proposed methodology, including the reconstruction process, training process, and augmentation techniques, are presented in detail. The detailed experimental setup is described in Section VI. The results of our anomaly detection experiments are presented and analyzed in Section VII. Finally, the paper concludes with a summary of the findings and future considerations in Section IX.

II. RELATED WORK

In general, (variational) autoencoder approaches have been extensively investigated in combination with deep learning architectures that can effectively handle sequential data. For instance, [17] proposed an LSTM-based autoencoder (LSTM-AE) and demonstrated its efficacy in the context of multivariate time series anomaly detection. Notably, this approach effectively incorporates the temporal dependencies in the data to improve anomaly detection performance. In a similar vein, [18] successfully employed an LSTM-based variational autoencoder to identify anomalies in sensor data. This approach represents a promising direction in the field of anomaly detection and has garnered significant attention in recent years. An additional approach that leverages reconstruction errors to identify anomalies is TadGAN, as introduced by [19]. This method is a generative adversarial network (GAN) based approach that focuses on identifying anomalous patterns in the data by comparing the reconstruction errors of the real and generated data.

To underscore the versatility and contemporary relevance of autoencoder-based anomaly detection in time series, noteworthy studies exemplify its successful application across diverse domains. In the realm of cybersecurity, the authors of [20] effectively employ an autoencoder-based approach for intrusion detection. Their methodology relies on establishing a threshold for the reconstruction error in order to flag anomalies indicative of cyber threats. Similarly, [21] demonstrate the effectiveness of an LSTM-based autoencoder in early lameness detection in dairy cattle, utilizing a threshold-based approach on the reconstruction error. Extending the applicability spectrum, [22] utilize autoencoder-based techniques for activity anomaly detection in smart homes. Their approach involves the identification of anomalous activities, such as unusual sequences or executions, with the final decision contingent upon a threshold imposed on the reconstruction error of the learned activity.

In the domain of industrial control and cybersecurity, [23] employ a threshold-based autoencoder to discern anomalous behavior in industrial control scenarios, spanning cyber attacks to faulty hardware. These instances collectively showcase the wide-ranging utility and continued state-of-the-art application of autoencoder-based time

series anomaly detection, emphasizing its adaptability across diverse domains and problem contexts.

For autoencoder-based anomaly detection techniques that rely on probability densities, several approaches have been proposed in the literature. For instance, [24] introduced a novel hybrid approach that leverages density estimation to label anomalies in the hidden layer of an autoencoder. Additionally, [12] proposed an autoencoder-based approach that identifies anomalies by fitting a parametric probability density function that best characterizes the underlying distribution of the reconstruction errors. Notably, the latter approach has been successfully applied to the task of time series anomaly detection. In recent years, publications have presented research employing the University of California at Riverside Anomaly Detection Benchmark (UCR Anomaly Detection), as described in Section VI-A. For instance, [25] investigated the performance of three deep learning-based and three classical machine learning-based approaches on the UCR Anomaly Detection Benchmark, utilizing all 250 subsets. Their findings demonstrated that, in this particular case, two classical machine learning methods (MDI [26] and MERLIN [27]) outperformed the examined deep learning approaches, achieving a UCR Score (which pertains to the adjusted recall@k where $k = 1$ metric elaborated upon in Section VI-B2) of 0.47 and 0.44, respectively. In their work, [28] proposed a novel joint architecture that combines regression, implemented by a Long Short-Term Memory (LSTM) Regressor, with a vanilla autoencoder. The authors demonstrated that this joint architecture achieved superior performance compared to individual approaches. Specifically, they reported an unweighted contextual F1 score of 0.47 on the UCR Anomaly Detection Benchmark.

While previous work on the UCR benchmark analyzed their algorithms on a generic level, we provide a deep-dive on a random sample of 10% of the data set. Moreover, our paper also includes extensive novel lessons learned that are helpful in analyzing other anomaly detection algorithms on that benchmark.

III. PRELIMINARIES

This section introduces the taxonomy of anomalies and gives a brief overview of the existing paradigms for anomaly detection in time series. It provides the necessary background for our work.

A time series $X \equiv \{x_i\}$ is a sequence of values at different time points, where x_i denotes the value at time i. Without loss of generality, we take i to be a non-negative integer in the range of $0, \ldots N-1$. It is common for x_i to be a vector, however, we will describe it as a scalar value to simplify the descriptions. We use the term a data point, or simply a point, to refer to x_i, and a window referring to a subsequence of X that is contiguous in time, $\{x_i, x_{i+1}, \ldots, x_{i+w}\}$, where w is known as the width of the window. An event in X may refer to a particular point or a window.

Commonly used in combination of time series analysis is the *sliding window approach*. This approach involves dividing the time series into fixed-length segments, or windows, and analyzing each window separately. The width of the window, denoted by w, determines the number of data points contained in each segment. By sliding the window along the time series with a step size, we can create overlapping segments and capture more comprehensive information. The size of the window and the step size can be adjusted to suit the nature of the data being studied.

A. TYPES OF ANOMALIES

Anomaly detection in time series data refers to the process of identifying patterns that deviate from the expected or normal behavior in a time-dependent context. Several types of anomalies have been identified in the literature [8] and [9], which are briefly reviewed next.

1) POINT ANOMALIES

In the field of time series anomaly detection, point-wise outliers are denoted as anomalous behaviors at individual time points that deviate substantially from the general pattern or trend of the time series data [9]. These outliers can manifest in the form of spikes, which are extreme values in comparison to the remaining data points, or glitches, which are relatively deviated values in relation to their neighboring points. In real applications, it might be hard to differentiate a point in time from a narrow time [8]. However, we will insist on a point anomaly to only contain a single time point to simplify the terminology.

2) COLLECTIVE ANOMALIES

Collective anomalies in the realm of time series anomaly detection, are referred to by [9] as a set of data points that exhibit a deviation from normal patterns over an extended period of time. While individual data points within this type of anomaly may not appear to be problematic, when analyzed collectively, they reveal a discernible deviation from the typical pattern. The detection of such anomalies is challenging, as they are not immediately apparent, therefore, the examination of long-term context is of particular importance in identifying them. Reference [8] describes this class of anomalies as pattern-wise outliers represented as anomalous time windows, which could be deviations from normal seasonality or trend.

3) CONTEXTUAL ANOMALIES

A contextual anomaly in time series anomaly detection according to [8] refers to a data point or sequence that is observed over a short time window but does not deviate from the normal range in a predefined manner, like that of point-wise anomalies. However, when analyzed within the given context, these data points exhibit deviation from the expected pattern or shape. This definition suggests that they are challenging to detect.

B. APPROACHES TO ANOMALY DETECTION IN TIME SERIES DATA

There is a significant body of research dedicated to anomaly detection in time series data. References [2] and [11] have provided extensive reviews of known methods and categorize them into several distinct groups below. Additionally, we further classify them according on their focus on modeling normal or anomalous events as well as a hybrid of both approaches.

1) STATISTICAL MODEL

These methods use historical data to build a model of the expected behavior of a system. New data is compared to the model, and if it does not fit within the model, it is considered an anomaly. For point anomalies, one example approach is to calculate a moving average and standard deviation of the data, and flag any data points that fall outside a certain number of standard deviations from the mean as anomalies. Typically these methods focus on modeling normal events.

2) PATTERN MATCHING

This method involves direct modeling of the time series data. In a supervised setting, where the characteristics of expected anomalous events are known, the detector compares new observations to a database of labeled anomalous events and flags those that are most similar. In the absence of labeled anomalies, the detector learns the most common historic patterns within the normal data and flags novel sub-sequences that do not match the historic corpus as anomalies. A hybrid approach, where observations are compared to a database of labeled normal and anomalous events, could also be implemented.

3) CLUSTERING

This approach projects the data into a multi-dimensional space and uses the density of resulting clusters. Observations that belong to dense clusters are considered normal, while those that are further away from or do not belong to these clusters are reported as anomalous. Clustering approaches typically try to learn patterns associated with normal behavior.

4) PREDICTION

A regression model is generated based on recent and longer-term trends of the system, predicting the expected value at some future time. When a new observation is received, it is compared to these predicted values. If there is a large difference between the observed and predicted values, the observation is flagged as anomalous.

5) DISTANCE-BASED

This approach defines a distance metric that allows newly received observations to be compared to preceding observations. The assumption is that similar mechanisms will result in smaller distances and will be flagged as normal, while larger distances will indicate the observation was generated by a different mechanism and will be flagged as anomalous. Distance-based pattern matching is a widely used method for anomaly detection, with one notable example being the approach detailed in the seminal work [29], which utilizes dynamic time warping as a measure of dissimilarity.

6) RECONSTRUCTION-BASED METHODS

These methods involve building a model of normal behavior by encoding window of a normal training time series into a low-dimensional latent space, and then using this model to reconstruct window from a test time series. Anomalies in the test series can be detected by comparing the reconstructed subsequences to the original, observed values. A prominent example for this paradigm is the family of autoencoders [3], further described in Section III-C. These approaches can be used to model normal events only (e.g if there are no anomalous events present during training) or only anomalous events (e.g if we know beforehand which anomalies to expect and what they look like).

7) ENSEMBLE

The ensemble approach uses multiple algorithms to observe each data point and a voting mechanism is employed over the outputs from each method. An ensemble can be constructed from a group of similar detectors or a collection of dissimilar detectors. The use of ensemble techniques can improve the overall success of a detection suite, but at the expense of increased complexity and computational time. Since these methods are a collection of multiple different approaches, they can consist of methods focusing on learning normal or anomalous behavior or they could implement a hybrid approach.

C. AUTOENCODERS FOR ANOMALY DETECTION

As an example of reconstruction-based methods, we next describe the use of autoencoders for anomaly detection. Autoencoders, as a well-established family of methods for anomaly detection, have been widely studied, for example, by [4], [30], and [31]. The class of autoencoders could be further sub-divided into two groups: traditional autoencoders and variational autoencoders.

1) TRADITIONAL AUTOENCODER

An autoencoder is a self-supervised approach of machine learning, consisting of an *encoder* $e(x)$, projecting inputs x (i.e., typically a window of a time series) onto a latent space z, with a lower dimension than the dimension of x, and a *decoder* $d(z)$ that reconstructs an approximate version x' of x from the latent space in the original space of x.

By training an autoencoder on only one class (i.e., the normal events), the model learns the underlying structure of normal events by projecting them onto latent space z. Given that an autoencoder is trained solely on normal data, it is assumed that the model would encode the normal events more

effectively than the anomalous events. This is expected to result in a greater reconstruction error for an anomalous event, as compared to a normal event [11].

The reconstruction error is typically defined as the *Mean-Squared-Error* (MSE) between the input and the decoded output of the model. For simplicity, let X represent the time series as an input data record, which could also be a time window from a time series. The MSE is defined as follows (recall that N is the number of time points in X):

$$MSE(X, e, d) = \frac{1}{N} (X - d(e(X)))^2 .$$

2) VARIATIONAL AUTOENCODER

The key idea behind variational autoencoders (VAE) is the introduction of a *prior distribution over the latent space*, typically a standard normal distribution. The encoder is then trained to approximate the true posterior distribution of the latent variables given the input data, while the decoder is trained to reconstruct the input data from the latent representation. This is done by minimizing the difference between the true posterior and the approximated posterior, often defined as the Kullback-Leibler (KL) divergence [32]. Rather than directly projecting the input data onto the d-dimensional latent space to obtain a single vector z denoting the encoded value in the latent space, two separate d-dimensional variables are computed: one representing the mean μ and the other representing the standard deviation σ of the encoded value $z = e(x)$ in the latent space. These two variables are used to sample a vector z from a normal distribution defined by μ and σ.

By adding the KL divergence (D_{KL}) as a regularization parameter to the loss function $\mathcal{L}$, it can be enforced that μ and σ follow a standard normal distribution $\mathcal{N}(0, 1)$. The loss function for the variational autoencoder changes from

$$\mathcal{L}_{AE}(x; e(x), d(z)) = MSE(x; e, d) = \frac{1}{N}(X - d(e(X)))^2$$

to

$$\mathcal{L}_{VAE} = \frac{1-\lambda}{N}(X - d(e(X)))^2 + \lambda * D_{KL}(\mathcal{N}(\vec{\mu}, \vec{\sigma})||\mathcal{N}(0, 1))$$

where

$$D_{KL}(p, q) = \int_{-\infty}^{\infty} p(x) \log(\frac{p(x)}{q(x)})$$

and λ is a tunable hyperparameter that regulates the importance of the MSE-loss term and the Kullback-Leibler divergence.

D. KERNEL DENSITY ESTIMATION

Kernel density estimation (KDE) is a non-parametric method for estimating the probability density function (PDF) of a random variable. It is used to smooth a histogram of data to estimate the underlying probability distribution of the data. The basic idea behind KDE is to place a kernel function, which is a probability density function, at each data point. The kernels are then summed to give an estimate of the overall probability density of the data. The kernel density estimation is described by the following formula [33]:

$$f(x) = \frac{1}{nh} \sum_{i=1}^{n} K \left(\frac{x - x_i}{h} \right)$$

The kernel function - K in the above equation - is typically a symmetric, bell-shaped curve such as a Gaussian distribution. The width - parameter h - of the kernels is determined by a parameter called the bandwidth, which controls the smoothness and the amount of bias-variance trade-off of the estimated density function. A small bandwidth will produce a more jagged estimate, while a large bandwidth will produce a smoother estimate. The variable x denotes the value for which a density estimate is calculated, and the variable x_i refers to the sampled values forming the underlying histogram of the n samples. There are several methods for selecting the bandwidth:

1) SILVERMAN'S RULE OF THUMB

It provides a rough estimate of the optimal bandwidth. It is based on the standard deviation of the data and the sample size [34].

2) SCOTT'S RULE

It is similar to Silverman's rule of thumb, but it is based on the interquartile range of the data rather than the standard deviation [35].

The selection of the bandwidth depends on the characteristics of the data and should be selected with respect to the goals of the analysis. Additionally, it is worth to mention that this is a difficult problem and sometimes there is no optimal solution and it depends on the goal of the KDE analysis. One of the main advantages of KDE is that it does not rely on any assumptions about the underlying distribution of the data. This is in contrast to parametric density estimation methods, which assume that the data follows a specific distribution (such as a normal distribution).

IV. CHALLENGES IN ANOMALY DETECTION

There are several challenges in anomaly detection, especially, the type of test cases from the UCR benchmark [16], formally introduced in Section VI-A. Next, we illustrate two of these challenges with examples from the UCR benchmark [10].

A. NO PRIOR KNOWLEDGE ABOUT ANOMALIES

Typically, anomaly detection tasks need to face imbalances between normal and anomalous cases, which creates challenges for anomaly detection. However, the test scenarios from the UCR benchmark is more challenging in that the training cases are always normal events. Thus, the training process is not expected to be able to learn anything about the anomalous events.

One family of methods for better using the data at hand is *augmentation* which is well adapted in computer vision [14], [15] and is a core part in contrastive learning [36].

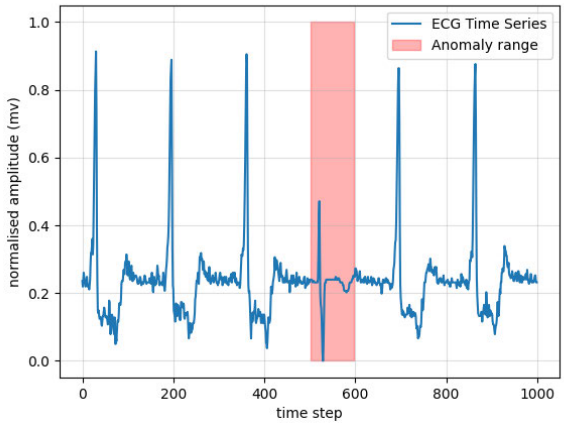

FIGURE 1. Example data (UCR data set nr. 121) of a patient's ECG with an anomaly representing a heartbeat from another patient's ECG.

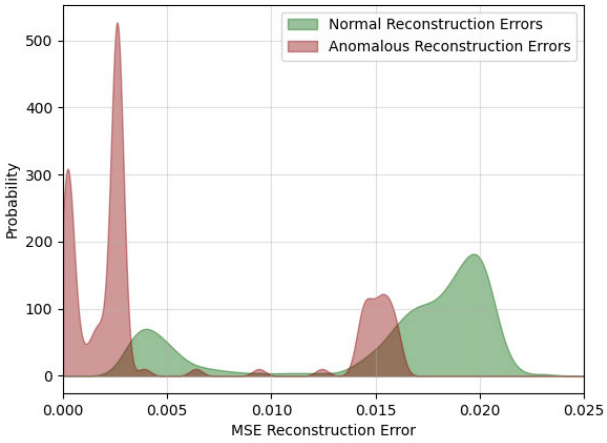

FIGURE 2. Reconstruction error densities for normal and anomalous errors of data set nr 121, depicted in Figure 1. The example showcases the deficiency of the large reconstruction error assumption for anomalous events.

However, while its applicability to time series has been researched [13], research is indecisive of the best performing approach. According to [13], methods range from simply adding Gaussian noise, over reversing the time series to more sophisticated methods such as generating samples leveraging GAN's or VAE's.

B. POLYVALENCE OF ANOMALIES

Given that the training would be only performed on the normal events, the anomalies are defined as those different from the expected. This heavily impacts one of the major assumptions for traditional autoencoder-based approaches where anomalies are supposed to result in larger reconstruction errors [11]. While it seems reasonable to expect an unseen anomalous event to have a greater reconstruction error than the normal events the autoencoder is trained with, it is important to note that this assumption implies its universality across all possible anomalies, which requires further examination.

For example consider the time series in Figure 1 depicting an electrocardiogram (ECG) with the anomalous time window highlighted in red. The anomalous section here is a heart beat of another patient's ECG, which was used to replace the original one. Encoding and decoding different windows of the data using an autoencoder trained on the normal events (heart beats) in this time series results in the histograms depicted in Figure 2. The reconstruction errors in red correspond to windows that contain anomalous events and green for the normal ones. It can be seen that the majority of the anomalous events can be reconstructed with relatively small errors, while the normal events appear to be harder to reconstruct.

In this particular case a traditional classification based on reconstruction errors exceeding a predefined threshold would not work properly because the anomalous events have smaller reconstruction errors. If we set up a threshold to be larger than most of the reconstruction errors from training on the normal events, the reconstruction errors of the anomalous event would never pass the threshold.

V. METHODOLOGY: AUTOENCODER-BASED ANOMALY DETECTION

This section provides a description of our proposed technique, the Variational Autoencoder-Based Anomaly Detection with Density Estimation for Time Series, herein referred to as VABADd. The theoretical framework underpinning our approach is drawn from architectural paradigms in published works [12], [17], [24]. These pioneering works seamlessly integrate the (Variational) Autoencoder framework with the computational prowess of Recurrent Neural Networks (RNNs) to address the dual challenges of encoding and decoding temporal sequences. In the instantiation of our VABADd, we conscientiously embrace the Long Short-Term Memory (LSTM)-based autoencoder architecture documented in [17]. This LSTM-based architecture serves as the baseline of our research framework. In later studies we refer to them as (Variational) Autoencoder-Based Anomaly Detection with Threshold for Time Series, designated as (V)ABADt.

Our method commences with the preliminary phase of autoencoder training, undertaken exclusively on a corpus of normal data. Subsequently, we embark upon a pivotal analytical step, entailing the application of kernel-density estimation (KDE) to the reconstruction errors accrued during the process of data reconstruction. This operation furnishes us with an empirically derived approximation of the probability distribution characterizing normal events post-reconstruction.

An innovative dimension in our approach is employing non-parametric density estimation techniques to scrutinize the reconstruction errors–a facet that has received comparatively limited attention in antecedent research endeavors. By contrast, extant methodologies have predominantly leaned upon simplistic threshold-based mechanisms for the evaluation of reconstruction errors, as elucidated by [17].

Moreover, we propose a *novel approach for augmenting time series data* by generating similar events through a

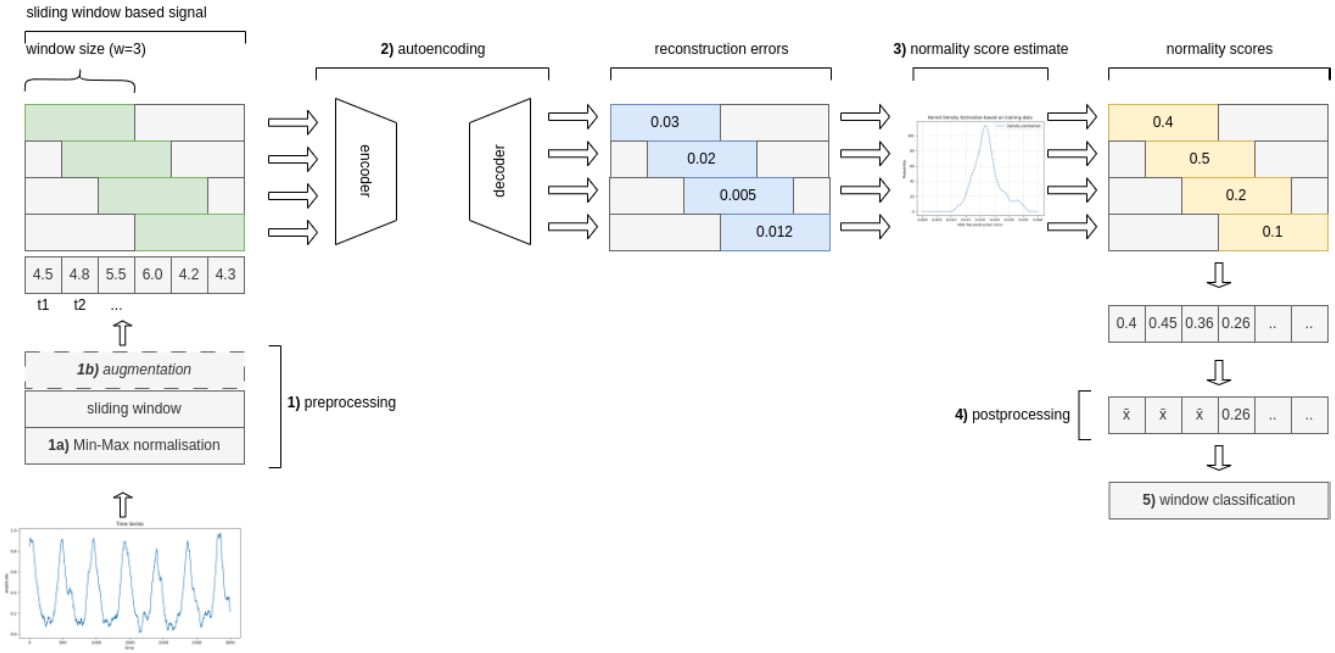

FIGURE 3. Overview of the proposed approach. First, the original data on the left is preprocessed (1). This step includes Min-Max normalization (1a) as well as creating sliding windows of a given size. Optionally, data augmentation (1b) can be performed as part of preprocessing described in Section V-A2. Secondly, each window is autoencoded (2) and the reconstruction error for each window is evaluated on the density estimation (3). The resulting normality scores are postprocessed (4) according to Section V-D and finally the windows are classified (5).

variational autoencoder, followed by a smoothing step to enhance the robustness of the model. After the training phase, new events could be encoded, decoded and evaluated against the estimated density to obtain a notion of normality of the input events.

The use of an autoencoder-based approach hereby addresses the problem mentioned in Section IV, that usually little to no prior knowledge of the anomalies is present during training, by learning a representation of what is considered normal without knowing any anomalous events. By performing a kernel density estimate, the polyvalence of the anomalies described in Section IV-B is addressed by overcoming the assumption that anomalous, unseen events will result in a larger reconstruction error. Instead of obtaining a reconstruction error, we devise a *measure of likelihood of normal* where larger values for normal events and smaller values for anomalous events. In the following section, the different parts are described in more detail and the process is depicted in Figure 3.

A. PREPROCESSING

1) MIN-MAX NORMALIZATION

Given a time series, preprocessing requires data normalization using Min-Max-Scaling defined by

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

Moreover, the data is reshaped in sliding windows of size w_s with a lag of 1. It is worth mentioning that the window sizes are unrelated to the size of the potential anomalous events and are chosen to minimize the reconstruction error of the autoencoder. The preprocessing step is necessary for both training and evaluation.

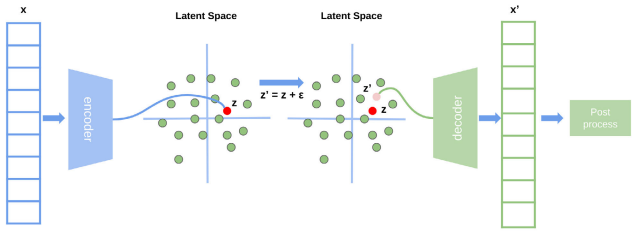

FIGURE 4. Data augmentation using a variational autoencoder.

2) DATA AUGMENTATION

For the optional augmentation, an augmentation technique leveraging variational autoencoders is used. This technique falls under the generational methods discussed in [13].

First, a variational autoencoder is trained on the normal training data and then each record is augmented by encoding and projecting it onto the latent space to obtain z. Adding Gaussian noise to z will result in a new point in the latent space in z's proximity called z'. Afterwards, z' is decoded and a new record similar to the input value is created. As a last step post-processing can be applied. The process is illustrated in Figure 4.

Generating new samples in the aforementioned way results in time series chunks with occasional outliers causing spikes in what should be smooth parts of the series. This

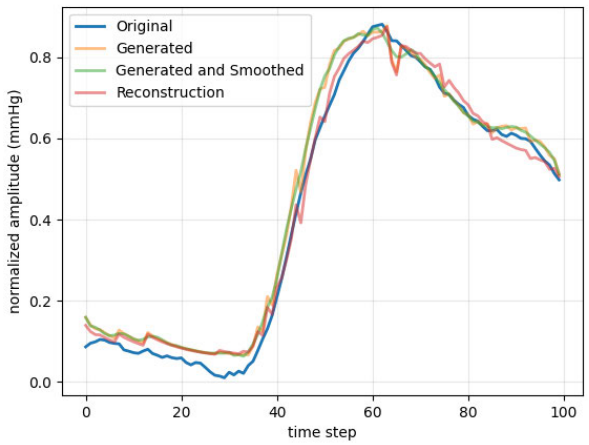

FIGURE 5. Augmentation example for ABP (Arterial Blood Pressure) data using a variational autoencoder showcasing non-smooth subsequences.

phenomenon can be seen in Figure 5 where the reconstructed and generated, non-smoothed data shows zig-zag patterns where the original data does not.

To mitigate this behavior, Savitzky-Golay filtering [37] is applied after decoding the new sample. The Savitzky-Golay filter is a type of smoothing filter that is commonly used to smooth a noisy signal.

The filter works by fitting a polynomial function of a certain degree to a set of data points surrounding the point of interest, and then using the value of that polynomial at the point of interest as the smoothed value.

The advantage of the Savitzky-Golay filter over other types of smoothing filters, such as the moving average filter, is that it can preserve the shape of the peaks in the data while still removing noise [37]. In addition, it has more flexibility in terms of the degree of the polynomial used, allowing for a better representation of the data. The exact parameters used for this work are described in Section VI-D

B. AUTOENCODING

In the second step, a preprocessed event passes through the autoencoder and creates a single reconstruction error for the event. In other reconstruction-base methods, different methods of reconstruction might be used for this step, while this work uses an autoencoder.

C. NORMALITY SCORE ESTIMATION

Given an estimated probability density $f_{KDE}(\epsilon)$ as described in Section III-D computed from the training process, we now need to calculate the normality score for non-anomalous events. In particular, point estimation is performed by looking up the estimated probability for the given reconstruction error ϵ. Since this point estimation is simply using the estimated probability density, larger values could be interpreted as indicating a normal event and smaller ones suggesting anomalous event.

D. POSTPROCESSING

After normality scores are assigned to event windows, postprocessing is applied. For each data point in the series, an individual score is calculated based on the overlapping sliding windows. Assigning a score to a data point in the time series is done by calculating the *mean normality score* of all the windows for which the data point is a part of.

For data points $x_t, t \in [w_s, max(T)-w_s]$, where w_s denotes the size of the window, the mean is taken over w_s number of data points. For values before time $t = w_s$ and after $t = max(T) - w_s$, the mean is less expressive and more prone to outliers in the reconstruction process as there are less overlapping windows for these ranges. Data points in these ranges are assigned the overall mean of the well defined range. As a consequence, anomalies at the beginning (i.e before $t = w_s$) and at the end (i.e. after $t = max(T) - w_s$) cannot be detected. Data points will end up with a low normality score if they consistently are part of windows for which reconstruction errors are uncommon.

E. CLASSIFICATION

In a last step, the actual classification is done and depends on the task and the data provided. In a scenario where anomalous data is present during training, a possible classifier could be implemented by using a threshold-based approach. In the case, a certain point x_t is labeled as anomalous if its corresponding normality score s_t is below a certain threshold t. If no anomalous data is present during training phase, data points could be ranked according to their normality score and the lowest scoring k can be selected.

F. TRAINING

During training, training data is preprocessed according to Section V-A and then a (variational) autoencoder is trained on the normal data only. After training the autoencoder, the normal training data is once again encoded and decoded. The resulting reconstruction errors are then used to fit a Kernel Density Estimator to obtain a density estimation for normal data. A final classifier is fitted after density estimation is done, based on the postprocessed data according to Section V-E.

VI. EXPERIMENTAL SETUP

In this section the data set used for the experiments as well as the evaluation metrics are described in detail. Additionally, the process of classification of an anomaly with respect to the chosen data set is discussed as well as the parameters for the augmentation process and the parameters of the proposed approach.

A. DATA SET

The University of California at Riverside (UCR) Time Series Anomaly Archive [16] is a benchmark data set for time series anomaly detection that was created to address the shortcomings of existing data sets such as the Yahoo S5

TABLE 1. The 25 UCR benchmark subsets used for our study. Column heading "No." is the test case number from the UCR benchmark, the "size" is given by the benchmark as the anomalous event size (in number of data points) and "w" is the window size used in later studies.

Domain	Descripton	No.	Size	w
APB trace	Advanced Peripheral Bus (APB) trace of a healthy walking man	*197*	67	100
ECG (human)	Two-lead human ECG trace that contains some PVC. PVC's are rare occurrences and are present in the training data as well as test data	*70*	200	100
		119	300	80
		121	100	130
		123	300	100
		193	100	70
		221	220	100
		229	10	100
		236	40	100
EPG	Electrical Penetration Graph (EPG) Signal of an Asian Citrus Psyllid insect	173	30	220
acceleration	Acceleration during swim cycles of a Blainville's beaked whale	102	160	70
blood pressure (internal bleeding)	Arterial blood pressure measurements of pigs showing internal bleeding	28	111	80
		33	170	100
		35	109	220
		138	10	170
respiration (breath cycles)	Human breath cycles (deep sleep).	83	100	120
temperature (hourly)	Weather data report from California Irrigation Management Information System (CIMIS) station 44	6	24	120
		114	24	90
walking (3D marker)	Vertical position of a 3D marker on a walking subject on a split-belt treadmill (GaitPhase Database)	22	118	180
		131	104	130
walking (acceleration)	Acceleration data of a walking subject (average of 3D signal)	53	231	100
		54	59	80
		249	30	50
walking (force plate)	Subject walking on a force plate in a biomechanics lab (Huntington's disease, highly asymmetric gait)	59	300	230
		62	110	100

Benchmark [38], PEI's Lab [39] or NASA [40]. This archive is comprised of a diverse range of data sets from various domains, including medicine, sports, entomology, industry, space science, robotics and more. The data sets are designed to provide a spectrum of problems that range from relatively simple to highly complex.

In order to create a data set that is free of the issues commonly found in existing anomaly detection benchmarks, [16] have taken a number of steps. For example, the type of anomalies cover the various different taxonomies introduced in Section III-A, which are supposed to test the generality of proposed algorithms. Additionally, the authors have made a concerted effort to include a wide range of data sets from different domains to prevent the archive from reflecting the authors' own biases and interests.

The UCR Time Series Anomaly Archive is specifically designed to consist of a *single anomaly per data set* and no anomaly in the training data, thus eliminating potential issues with scoring by rendering the discovery of anomalies as a binary event. By providing a large number of data sets, the aggregate of these binary events can be used to calculate a true percentage for accuracy, allowing for meaningful performance comparisons.

In terms of data labeling, the UCR benchmark clearly labels the training data and also clearly indicates the range of the anomalous event within the data set. This allows for testing algorithms in a controlled and consistent manner and making meaningful comparisons to other algorithms. The archive also includes detailed metadata and provenance information for each data set.

Because of computational limitation, a uniformly, randomly sampled subset of 25 data sets is used. The sampled data sets can be found in Table 1 and cover multiple domains with different types of anomalies.

B. EVALUATION METRICS

Reference [16] advocate for treating the classification of anomalies as a binary event, where the central distinction lies in the algorithm's ability to accurately identify the anomaly or not. In cases where the anomaly is successfully identified, a point is awarded to the corresponding subset. The criteria for recognizing an anomaly are contingent upon whether the algorithm effectively designates a data point as anomalous within the predefined range. When employing a window-based methodology, an anomaly is considered detected if it either intersects with the prescribed range or is entirely encapsulated by it.

Furthermore, we emphasize the insightful guidance provided by Eamonn Keogh, the author of the UCR benchmark, who recommends eschewing complex and opaque scoring functions for this benchmark. Instead, he proffers a set of methodological principles for scoring function design, which were expounded upon during a workshop held at KDD millets [41]. These principles have been diligently integrated into our methodological framework:

Unification via a Singular Metric: Our selected metric serves to distill the multifaceted evaluation process into a solitary numerical value, thereby facilitating straightforward comparisons across diverse algorithmic approaches.

Binary Scores for Individual Time Series: We assign binary scores, denoting a value of 1 for accurate predictions and 0 for inaccuracies, to each discrete time series. Subsequently, these binary scores are amenable to aggregation, culminating in a comprehensive evaluation score for the entire dataset.

Convergence Towards Zero for a "Random Dart" Algorithm: Our metric is meticulously structured to yield a score approximating zero for an algorithm resembling a stochastic dart-throwing process, while trending towards unity for an algorithm that attains perfection in the realm of anomaly detection.

Moreover, the author of the UCR benchmark asserts that, within the confines of this specific collection of datasets, the sole meaningful metric assumes the form of "n out of N." This metric signifies the proportion of accurately identified anomalies per dataset within a total of N datasets. In strict adherence to this proposition, we, as the authors, have meticulously implemented this prescribed methodology.

Given the novelty of the aforementioned scoring methodology, it is prudent to establish a connection with established and widely recognized metrics, specifically adjusted precision@k and adjusted recall@k. The comprehensive elucidation of these metrics will be provided in subsequent sections for a more thorough understanding.

1) PRECISION@K

In the field of anomaly detection, precision@k is a measure of performance used to evaluate the quality of a model's ranking of anomalous instances [41]. It is a variation of the more commonly used precision metric, which measures the proportion of true positive results among all positive predictions (i.e anomalous prediction) made by a model. By comparing the ranking of anomalous instances produced by a model to the true ranking of those instances, precision@k allows for a more nuanced evaluation of a model's performance.

$$p@k = \frac{\text{Number of anomalous instances @k}}{k}$$

Precision@k is computed according to above equation and calculates the ratio of true positives among the top k model-ranked instances. In our approach, instances are ranked by ascending *normality scores*, while in traditional reconstruction error-based methods, rankings are determined by descending *reconstruction error*. The rationale is that larger reconstruction errors typically correspond to anomalous events. For example, correctly identifying 10 out of the top 15 anomalous instances yields a precision@15 of 0.67, indicating a 67% precision rate.

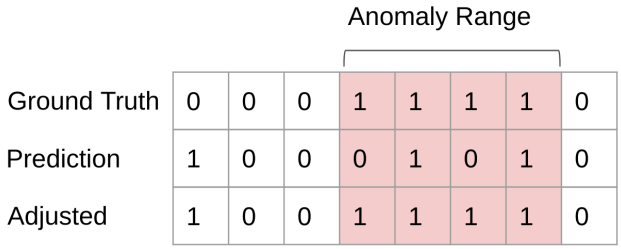

FIGURE 6. Adjustment method proposed in [31] for anomaly detection.

Specifically, our investigation centers on precision@1, an evaluative measure that appraises the accuracy of index labeling within the context of the lowest k indices relative to their anticipated "normality" scores. By confining k to a value of 1, we instantiate a binary classification framework, characterized by the attributes advanced in [41] and elaborated in Section VI-B. Within this framework, predictive outcomes are distinctly dichotomous: they are deemed either accurate (precision@1 $=$ 1, denoting that the lowest scoring index pertains to the anomalous range) or inaccurate (precision@1 $=$ 0, signifying that the lowest scoring index does not align with the anomalous range).

2) ADJUSTED RECALL@K

In the realm of anomaly detection, recall@k serves as a crucial evaluation metric for assessing the performance of a model in identifying all pertinent instances of abnormal behavior or events within a data set. In this context, recall@k is utilized to evaluate the ability of a model to identify all instances of abnormal behavior within the top k predictions. The metric is mathematically defined as the ratio of the number of true positive instances (anomalies) identified in the top k predictions to the total number of true positive instances in the entire data set.

For this particular problem an adjusted recall@k is used which builds on the idea of [31], where they introduce a simple adjustment method for anomaly detection. They consider a anomalous subsequence as detected and label all points in an anomalous segment as anomalous if at least one instance in this range was correctly labeled as anomalous, illustrated in Figure 6.

Similarly we modify the recall@k metric which is defined as

$$Recall@k = \frac{\text{Number of anomalous instances @k}}{\text{total number of anomalous instances}}$$

into

$$Recall_{adjusted}@k = \mathbb{1}\{\text{anomalous instance is present @k}\}$$

k	1	2	3	4	5	6	7	8	...	n
Normality score per instance (ascending)	0.2	0.3	0.5	0.5	0.6	0.8	1.1	1.3	...	8.5
Ground Truth per instance	0	0	1	0	1	1	0	1	...	0
Recall@k	0	0	1/p	1/p	2/p	3/p	3/p	4/p	...	1
Adjusted Recall@k	0	0	1	1	1	1	1	1	1	1

FIGURE 7. Example data set with *p* number of anomalous instances showcasing the difference of the adjusted recall@k metric and recall@k.

where $\mathbb{1}$ denotes the indicator function returning 1 if a true anomalous instance is present within the top k considered instances.

The adjusted recall@k metric serves as a valuable evaluation tool for assessing the performance of a model in identifying instances of abnormal behavior or events within a data set. A value of 1 for the adjusted recall@k for low values of k implies that the true instances of anomalous behavior are effectively identified and ranked highly by the model. Conversely, if the recall@k switches from 0 to 1 for a high value of k, it suggests that the instances of abnormal behavior are not easily identifiable, and a large number of normal data points are considered more suspicious by the model. An illustration of the adjusted recall can be seen in Figure 7.

Ideally, the adjusted recall@k should return a value of 1 for k = 1, as this implies that at least one true instance of anomalous behavior is ranked the highest by the model. In cases where the adjusted recall@k starts returning a value of 1 for a low value of k ($k > 0$), it may indicate the presence of outliers which are considered more anomalous than true anomalies, despite the latter being suspicious as well.

This metric can be averaged over a set of data sets and it provides an indication of how well the true anomalous data points are scored by the model. If for increasing values of k, the averaged adjusted recall@k shows a steep increase, it can be inferred that the true anomalies generally rank high but there are a few normal instances which are even more suspicious. On the other hand, if the averaged recall@k stagnates or increases slowly, the model may fail to assign a high level of suspicion to true anomalous instances, implying that there are a large number of normal data points that are regarded as more concerning by the model.

It is worth noting, that the indicator function utilized for the adjusted recall@k metric corresponds to the identification of a binary event in a manner that is consistent with the intended use of the UCR benchmark.

C. ANOMALY CLASSIFICATION

Following the methodology outlined in Section V-C for point estimation and subsequent postprocessing as detailed in Section V-D, we derive an *anomalous score* for each timestep x_i. This score represents an average of the reconstruction likelihood estimates for all overlapping time windows in which x_i was involved. Given that a timestep within an anomalous segment of the time series is expected to yield a low normality score, the resulting average reflects this tendency. Consequently, we identify the timestep x_m with the lowest score as anomalous, aligning with the precision@k metric where $k = 1$ and the ordering is based on normality scores in ascending order, as explicated in Section VI-B1.

D. AUGMENTATION

In experiments that incorporate augmentation, the methodology outlined in Section V-A2 is employed. This involves leveraging a variational autoencoder to project a given input, specifically a time series window, onto a latent space. Subsequently, Gaussian noise is injected to obtain a neighboring point within the vicinity of the projected input, and this neighbor is then decoded and smoothed.

The weight of the Kullback-Leibler Divergence (parameter λ in Section III-C2) is set to 1e−5. The generators are pretrained and stay the same for every experiment to evaluate the performance gain without any variation due to re-computation with fluctuating generator performance. For each of the 25 subsets the best generator according to the loss metric described in Section III-C2 out of 10 runs is selected.

Augmentations are performed during the preprocessing step after data has been normalized. For all augmentations the noise added to the latent variable z to obtain z' follows a Gaussian distribtution with $\mu = 0$ $and \sigma = 0.1$. For smoothing with the Savitzky-Golay filter, the filter window is set to 40 and the order of the polynomial to 39. This parameters were chosen based on a hyperparameter sweep.

E. SYSTEM PARAMETERS

In this section, we expound upon the parameters and configurations employed in the various constituent elements of the proposed methodology. All autoencoders in our study were subjected to optimization via the ADAM algorithm, utilizing its default hyperparameters, and were charged with the objective of minimizing the Mean Squared Error (MSE) loss function, as elucidated in Section III-C1. Default parameters were consistently applied unless explicitly modified. The implementation of the autoencoder was realized through the PyTorch framework [42].

1) AUTOENCODER

In this configuration, both encoder and decoder utilize Long Short-Term Memory (LSTM) networks. The preference for LSTMs over Gated Recurrent Unit (GRU) networks arises from the lack of a unanimous consensus on their superiority, as noted in various studies [43], [44], [45]. While GRUs generally offer faster computation, LSTMs tend to better capture intricate patterns in time series data [43]. Given the emphasis on handling complexity rather than time efficiency in our work, especially with a benchmark designed to contain complex anomalies and time series intricacies, we opt for the use of LSTM networks for encoding and decoding data.

Figure 8 illustrates our autoencoder setup. Encoding is based on one LSTM layer with one hidden layer. The output

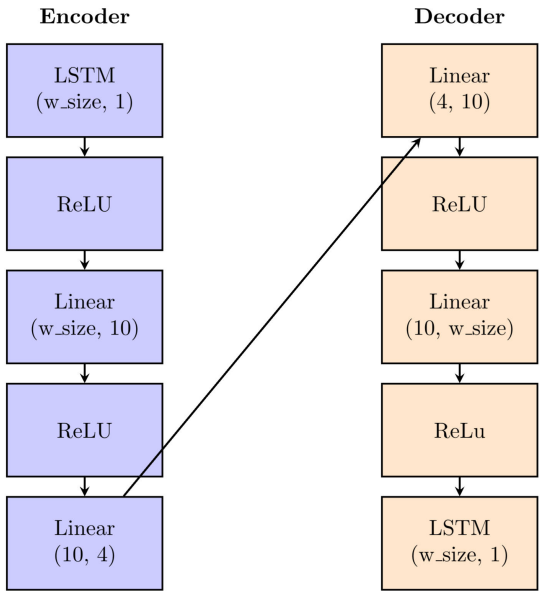

FIGURE 8. The overarching architectural framework for the employed autoencoder, with W_{size} denoting the window size pertinent to the respective time series as documented in Table 1.

is then applied to a ReLU activation function followed by a linear layer with output-dimension 10 with ReLU activation. Finally, an additional linear layer is added projecting the previous layer onto the latent space of dimension 4 without activation.

Decoding first applies a linear layer to the latent variable z with output dimension 10 followed by a ReLU activation. As a second step, another linear layer maps the previous values to the output size equal to the chosen window size, described in Table 1, in combination with a ReLU activation function. Lastly, one layer of LSTM with a hidden size of 1 is used to reconstruct the time series.

The batch size is set to 512 and the autoencoder is trained for 500 epochs. The number of time steps considered depends on the data set described in Table 1 and were chosen based on a hyperparameter-sweep.

2) VARIATIONAL AUTOENCODER

The setup for the variational autoencoder is consistent with the one for the traditional autoencoder described in the previous section except for the variational part for which the Kullback-Leibler divergence coefficient λ is set to $1e{-}5$. Additionally, the last linear layer in the encoder for the traditional autoencoder, responsible for projecting the output of the previous layer onto the latent space of dimension 4, is replaced with two separate linear layers with output dimension 4. One layer represents μ and the other layer represents σ of the latent distribution.

3) KERNEL DENSITY ESTIMATION

To perform the density estimation, the KDEpy library [46] was used. For choosing an appropriate bandwidth (parameter h in the formula in Section III-D), Silvermans method is applied in combination with a Gaussian Kernel.

VII. EXPERIMENT

The subsequent experiments aim to validate the effectiveness of the proposed methodology and evaluate the impact of data augmentation in the field of anomaly detection. The principal objective is to conduct a thorough comparison between our approach and conventional reconstruction- and threshold-based methodologies, emphasizing the differentiation of their respective performances. It is imperative to highlight that the primary focus is not achieving state-of-the-art performance on the employed benchmark. Rather, the experiments are systematically designed to identify potential enhancements introduced by our proposed approach to existing autoencoder methodologies.

A. DENSITY-BASED VS. RECONSTRUCTION ERROR-BASED

With this experiment the impact of the density-based approach (V)ABAD**d** shall be tested over the more traditional approach of classifying events as anomalous based on the larger reconstruction error assumption, which we call (V)ABAD**t** where **t** stands for the traditional threshold-based approach.

1) RECONSTRUCTION ERROR-BASED USING TRADITIONAL AUTOENCODER (1) (ABADT)

This setup utilizes the traditional autoencoder discussed in Section III-C1. Anomalous data points are identified based on their highest reconstruction error following postprocessing, as detailed in Section V-D. Unlike other approaches, no further density estimation is conducted after autoencoding; instead, the reconstruction errors are employed exclusively for postprocessing and scoring. This method is consistent with the work of [17].

2) DENSITY-BASED USING TRADITIONAL AUTOENCODER (OUR APPROACH) (2) (ABADD)

For this setup, a traditional autoencoder, described in Section III-C1, is used and the data point which displays the lowest normality score is labeled as anomalous. Here, a density estimation is done for reconstruction errors after autoencoding. Afterwards, the reconstruction errors are post-processed according to Section V-D. To ensure the differences in performance originating from the density-based scoring and not from the underlying autoencoder, the exact same trained instance as in setup 1 is used.

3) RECONSTRUCTION ERROR-BASED USING VARIATIONAL AUTOENCODER (3) (VABADT)

This configuration is congruent with Setup 1, as delineated in Section VII-A1, with the sole distinction being the utilization of a Variational Autoencoder, as expounded upon in Section III-C2, in lieu of the traditional autoencoder. This particular instantiation aligns with the established baseline methodology described in [17].

4) DENSITY-BASED USING VARIATIONAL AUTOENCODER (OUR APPROACH) (4) (VABADD)

This setup is equal to setup 2 described in Section VII-A2 except a variational autoencoder is used instead of a traditional one. Again, the same trained variational autoencoder instance is used as for setup 3 to ensure that any differences in performance are only due to the density estimation.

B. DATA AUGMENTATION

The impact of data augmentation using the method described in Section V-A2 is tested. For every time window w_i, we employ the methodology outlined and generate its augmented counterpart w_i'. This process is applied to each time window in the training dataset, effectively doubling the training data. The augmented dataset comprises one-half of the original data, while the other half consists of the newly generated augmented instances.

Given that augmentation effectively doubles the volume of training data, we establish a control group to evaluate the efficacy of augmentation methods, as demonstrated in [47]. This control group involves the replication of the training data, ensuring an equivalent number of training data samples to that of the augmented approach. Consequently, each time window is present twice in the control group, mirroring the approach in [47].

An improvement in performance of the control group in comparison to the group without augmentation would imply that a longer training period would have sufficed. Conversely, a decline in performance of the control group could indicate over-fitting. An enhancement in performance of the augmented data group compared to the control group would demonstrate the advantages of utilizing augmented training data. The impact of augmentation is tested on the proposed approach using density-based classification and a variational autoencoder (VABADd) as well as on the baseline reconstruction error-based using a variational autoencoder (VABADt) approach.

VIII. RESULT

Subsequently, we discuss the findings of the experiments detailed in Section VII. Firstly, the impact of relying on a density estimate for normality is evaluated. Secondly, a direct comparison of the underlying autoencoder architectures is conducted. Lastly, the outcomes of data augmentation for both reconstruction error-based and density-based approaches, using a variational autoencoder are reported and analyzed.

A. RECONSTRUCTION ERROR-BASED VS. DENSITY-BASED

Table 2 depicts the overall average precision@1 for all the 25 subsets described in Table 1 based on 10 runs. It can readily be seen that the **density-based approach outperforms the reconstruction-based method by 8.4% using a traditional autoencoder (ABADd vs. ABADt) and by 9.6% when used in combination with a variational autoencoder (VABADd vs. VABADt)**. This suggests that the assumption that anomalous events, unseen during training, does not necessarily lead towards larger reconstruction errors.

TABLE 2. Comparison of the benefit of density-based anomaly detection over reconstruction-based methods as well as the benefit of variational autoencoders over traditional ones.

	reconstruction error-based (existing)	density-based (ours)	improvement using density estimation
average precision@1 ABAD(t/d)	0.332	0.416	+8.4%
average precision@1 VABAD(t/d)	0.364	0.46	+9.6%
improvement over auto-encoder	+3.2%	+4.4%	

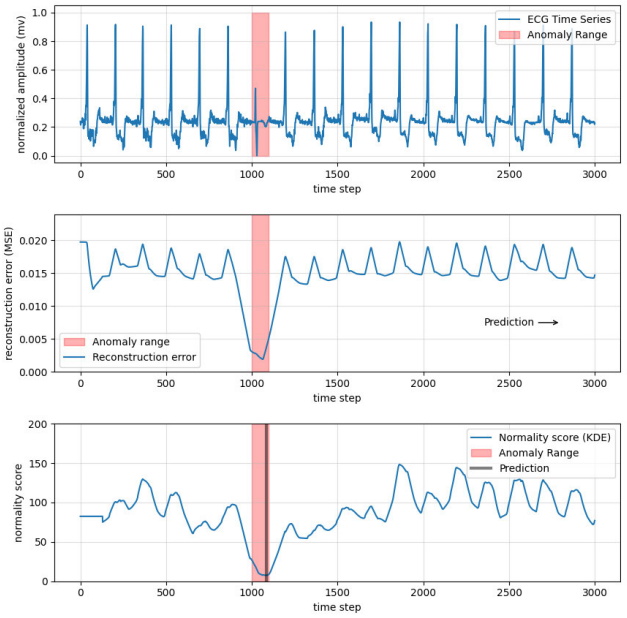

FIGURE 9. The first figure depicts the ECG time series where the anomaly marked in red originates from substituting another patients ECG. In the second figure the reconstruction error at each time step t is shown. The true anomaly could not be detected by labelling the largest reconstruction error as such. The last figure displays the normality scores for each reconstruction error at time t and shows that the true anomaly could be detected based on the lowest normality score.

This is further supported by Figure 9 where in the first plot the original signal is depicted with the anomalous range highlighted in red, the second plot represents the reconstruction error at each time-step, based on the described process and the third plot corresponds to the estimated normality scores. The anomaly in this ECG stems from replacing a certain range with a second, different ECG. A close up of this anomaly can be seen in Figure 1.

The reconstruction error appears to be uncommonly small for the values in the anomalous range, as shown in Figure 10. Consequently, it cannot be detected by selecting the data point which maximizes the reconstruction error. However, it can be detected with the density based approach indicated by a large dip in normality score around the defined anomaly range.

TABLE 3. Precision@1 comparison of the reconstruction error-based and density-based approach using a traditional autoencoder (ABADt vs. ABADd) and variational autoencoder (VABADt vs. VABADd) on the various UCR subsets (sorted by time series type and increasing difference of the density based approach over reconstruction based). "No." refers to the UCR data set number and "Δ" refers to the difference of the density-based approach (VABADd) over the reconstruction error-based (VABADt).

	average precision@1				average precision@1		
No.	ABADt (existing)	ABADd (ours)	Δ	No.	VABADt (existing)	VABADd (ours)	Δ
197	**0.4**	**0.4**	0.0	197	**0.6**	**0.6**	0.0
193	**0.2**	0.1	-0.1	193	**0.5**	0.3	-0.2
70	**1.0**	**1.0**	0.0	70	**1.0**	**1.0**	0.0
119	**1.0**	**1.0**	0.0	119	**1.0**	**1.0**	0.0
221	**0.3**	**0.3**	0.0	221	**0.3**	**0.3**	0.0
229	**0.0**	**0.0**	0.0	229	**0.0**	**0.0**	0.0
236	**0.0**	**0.0**	0.0	236	**0.0**	**0.0**	0.0
123	0.0	**0.2**	0.2	123	0.0	**0.3**	0.3
121	0.3	**0.9**	0.6	121	0.2	**1.0**	0.8
173	0.0	**0.1**	0.1	173	*0.2*	0.1	-0.1
102	**0.5**	**0.5**	0.0	102	**0.6**	**0.6**	0.0
28	**0.1**	**0.1**	0.0	28	**0.0**	**0.0**	0.0
33	**0.5**	**0.5**	0.0	33	0.4	**0.6**	0.2
138	0.6	**0.7**	0.1	35	0.2	**0.4**	0.2
35	0.2	**0.6**	0.4	138	0.3	**0.6**	0.3
83	**0.0**	**0.0**	0.0	83	**0.1**	0.0	-0.1
6	**0.4**	**0.4**	0.0	6	**0.6**	**0.6**	0.0
114	0.4	**0.5**	0.1	114	0.5	**0.6**	0.1
131	**1.0**	0.8	-0.2	131	**1.0**	0.7	-0.3
22	**0.2**	**0.2**	0.0	22	**0.6**	**0.6**	0.0
249	**0.0**	**0.0**	0.0	249	**0.0**	**0.0**	0.0
54	0.9	**1.0**	0.1	54	0.7	**1.0**	0.3
53	0.0	**0.8**	0.8	53	0.0	**0.7**	0.7
59	**0.2**	**0.2**	0.0	62	**0.2**	**0.2**	0.0
62	**0.1**	**0.1**	0.0	59	0.1	**0.3**	0.2
	0.332	**0.416**	0.084		0.364	**0.46**	0.096

A detection approach relying solely on reconstruction errors and the assumption that anomalies result in larger values, would predict the anomaly to be around $t = 13500$ where the ECG time series shows a small outlier in amplitude. This outlier could arise from the fact that this data set contains Premature Ventricular Contractions (PVCs) which appear irregularly in this particular data set and are rare events, too.

Considering Figure 10, it is evident that the majority of data points corresponding to the anomalous range for dataset nr. 121 result in small reconstruction errors. While the density estimation depicted as the grey area represents normal data well, there is a light shift to smaller reconstruction errors. However, it suffices to clearly separate anomalous from normal events. This showcases that using an estimation of normality can overcome the deficiencies of assuming larger reconstruction errors for anomalies.

An in-depth analysis of the performance on the various subsets, as depicted in Table 3, reveals that the utilization of a density-based approach results in a significant improvement in performance for subsets 53 and 121. These subsets

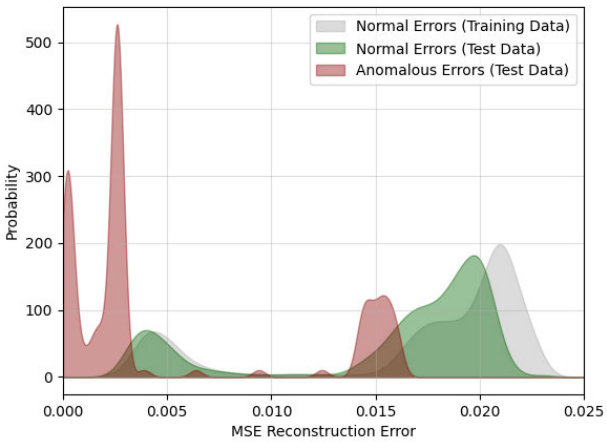

FIGURE 10. ECG reconstruction error histogram including density.

TABLE 4. Overall performance of augmented, non-augmented and control group for reconstruction error and density-based approaches using a variational autoencoder (VABADt and VABADd).

	no augmentation	control group	augmented	performance increase over control group
average precision@1 VABADt (existing)	0.404	0.432	0.452	+2%
average precision@1 VABADd (ours)	0.448	0.5	0.496	-0.4%

exemplify the phenomenon previously discussed, in which the reconstruction errors for anomalous instances are notably low.

Furthermore, Table 3 shows that density-based anomaly detection performs at least as good as reconstruction errors in combination with traditional autoencoders in 23 out of 25 instances and in combination with a variational autoencoder in 21 instances. However, it should be noted that the density-based approach demonstrates the least favorable performance for subset 131, with a difference in average precision@1 of -0.2 and -0.3 for traditional and variational autoencoders, respectively.

B. AUTOENCODER VS. VARIATIONAL AUTOENCODER

Comparing the underlying autoencoder architecture shows consistent improvement for using variational autoencoders on the selected subsets in disregard whether a density-based or reconstruction-based approach is used. In case of a reconstruction-error-based approach, **variational autoencoders outperform traditional ones by 3.2% and for a density-based approach by 4.4%**.

C. IMPACT OF DATA AUGMENTATION

The impact of the proposed augmentation method on variational autoencoder-based approaches can be found in Table 4. **When classification is based on reconstruction error (VABADt), the augmentation method can improve the performance by 2% over the control group**. If used in combination with normality estimates (VABADd), the performance decreases slightly by 0.4%. However, in both cases it is evident that simply replicating training data increases performance by around 2.8% for the error-based method and by 5.2% for the density-based approach.

Considering the precision@k measures for both approaches it can be seen that when using an error-based approach, the metric initially increases before it starts to decline. In this case, instead of classifying the window with the largest reconstruction error, selecting a k_s that maximizes the p@k and randomly sample from the top k_s windows uniformly, would, on average, result in a better performance than labelling a window anomalous with the largest reconstruction error. Augmentation in this case helps shifting this k_s towards $k_s = 1$, suggesting that augmentation generally helps to assign true anomalous windows a larger reconstruction error.

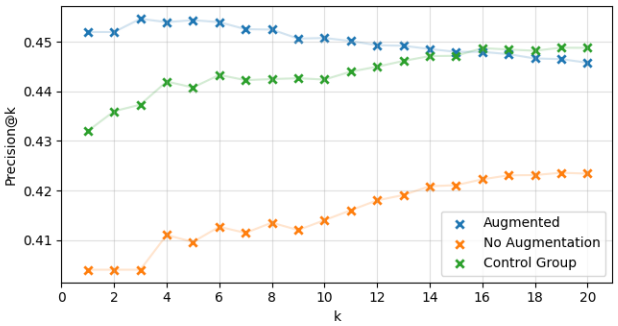

FIGURE 11. Average precision@k metrics over the 25 UCR subsets for reconstruction error-based anomaly detection (VABADt) and the impact of augmentation.

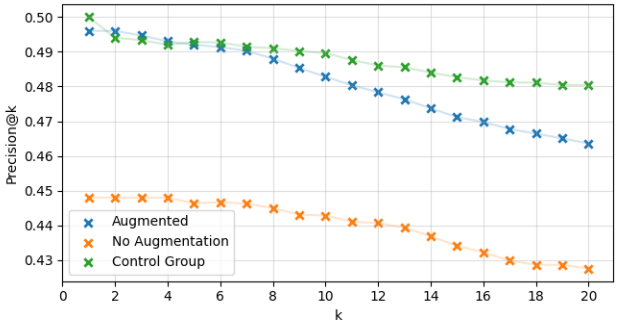

FIGURE 12. Average precision@k metrics over the 25 UCR subsets for density-based anomaly detection (VABADd) and the impact of augmentation.

The control group in this scenario increases the correct identification of anomalous windows. However, it appears that while overall true anomalous windows are assigned a large reconstruction error, there are instances that raise a bit more suspicion and are therefore ranked higher on this metric. This suggests that there could exist normal outliers causing large reconstruction errors for the control group and the one without augmentation. **Augmentation in this case helps reducing the proportion of normal data occurring at the top (i.e increasing the rate of true positives)**.

Upon examination of the average adjusted recall@k metric for the reconstruction-error-based approach (VABADt) in Figure 13, it is evident that the control group exhibits a greater rate of increase compared to the augmented group. This outcome suggests that the control group struggled to effectively differentiate between anomalous and normal events, whereas the augmented group displayed a less steep ascent in terms of adjusted recall@k, indicating that this method generally enhanced the ability to distinguish between anomalous and normal events as true anomalous instances raise more suspicion and thus rank higher.

Considering Table 5, **for the reconstruction-based approach (VABADt), augmentation performs in 20 out 25 cases at least as well as the control group**, in 9 out of 25 cases better and on 5 data sets the performance is worse compared to the control group. Augmentation performs worst

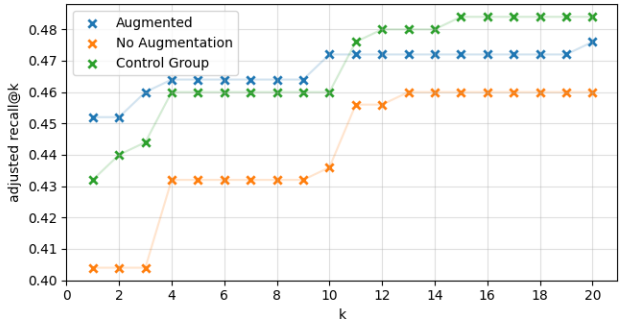

FIGURE 13. Average adjusted recall@k metrics over the 25 UCR subsets for reconstruction error-based anomaly detection (VABADt) and the impact of augmentation.

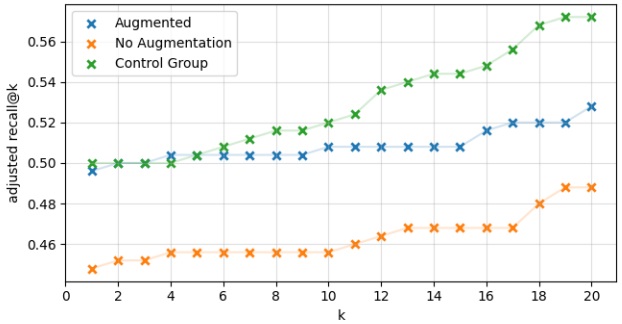

FIGURE 14. Average adjusted recall@k metrics over the 25 UCR subsets for density-based anomaly detection (VABADd) and the impact of augmentation.

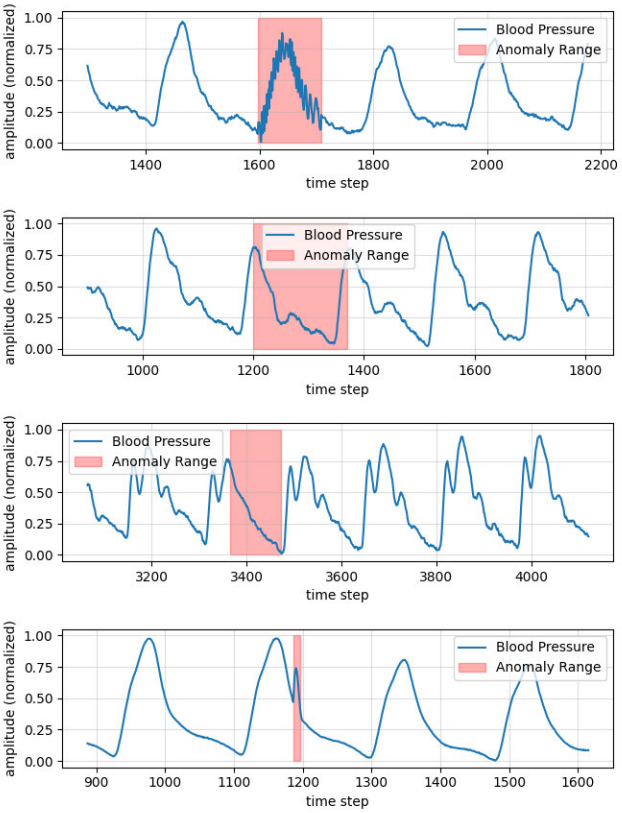

FIGURE 15. Blood pressure data sets number 28, 33, 35, 138 and their anomalies.

for set nr. 197 with a decrease in performance of 50% over the control group.

For time series in the blood pressure domain, the performance increase is quite significant, except for data set nr. 28. In Figure 15 we see the four blood pressure data sets and their anomalies. Interestingly, data set nr. 28 contains the most prominent anomaly where the authors added a synthetic anomaly consisting of a series of sine waves making a piece of smooth data become fluctuating and rough whereas the anomalies for data set 33, 35, and 138 are more subtle. The anomaly in data set 33 was generated through the implementation of a downsampling technique on a limited sample of normative arterial blood pressure data. Specifically, a reduction in temporal resolution was achieved by combining every two consecutive data points within the range via averaging. For number 35, a moving average was applied to the anomaly range, eliminating about 4 extreme points within that cycle. In data set nr. 138, a small second peak was introduced as an anomaly. It appears that for blood pressure time series, the size of an anomaly does not contribute significantly to its detection, but its type does. Augmentation in this case helps detecting presumably more subtle anomalies as the performance for data sets 33, 35 and 138 could be increased significantly.

For the density-based approach (VABADd) this suggests that by relying on an estimate of normality, this phenomenon does not occur and windows with low normality scores really tend to be true anomalous windows, as for all density-based setups the precision@k metric is monotonically decreasing. However, while the control group and the augmented scenario tends to perform equally well for small k's, augmentation declines more rapidly after $k = 7$. This obersation suggests that for the majority of the subsets of the UCR anomaly detection archive used, anomalous windows are still assigned the lowest normality scores. The more rapid decline in performance could imply that the estimated density, based on the augmented data, is more receptive to a larger variation of data and is thus less sensitive towards anomalies (i.e certain anomalies are not labeled as such anymore, resulting in a higher false positive rate).

The adjusted recall@k metric for the density-based approach (VABADd), as depicted in Figure 14, reveals that both the augmented group and the control group initially exhibit similar performance, but diverge after $k = 6$, with the control group displaying a greater rate of increase. This suggests that while the proposed augmentation method in combination with a density-based approach is able to rank true anomalous instances similarly to the control group for smaller k. However, the augmentation method fails in assigning low normality scores to a significant number of instances that the control group still deems suspicious. This finding suggests that the augmentation method results in the

TABLE 5. Precision@1 comparison of the reconstruction error-based and density-based approach using a variational auto-encoder on the various UCR subsets and the impact of data augmentation (sorted by time series type and increasing difference of augmented group over the control group). "No." = UCR data set number, "No aug" = no augmentation, "c grp" = control group, "aug" = augmentation applied and "Δ" = improvement of augmentation (aug) over the control group (c grp).

	average precision@1 VABADt (existing)					average precision@1 - VABADd (ours)				
No.	no aug	c grp	aug	Δ	No.	no aug	c grp	aug	Δ
197	0.4	**0.8**	0.3	-0.5	197	0.4	**0.8**	0.3	-0.5
221	0.3	**0.7**	0.5	-0.2	221	0.3	**0.8**	0.5	-0.3
123	**1.0**	**1.0**	**1.0**	0.0	123	0.4	**0.3**	0.1	-0.2
70	**1.0**	**1.0**	**1.0**	0.0	70	**1.0**	**1.0**	**1.0**	0.0
119	0.5	**0.6**	**0.6**	0.0	119	**1.0**	**1.0**	**1.0**	0.0
121	**0.0**	**0.0**	**0.0**	0.0	121	**1.0**	**0.8**	**0.8**	0.0
193	**0.5**	**0.5**	**0.5**	0.0	193	0.3	**0.5**	**0.5**	0.0
229	**0.0**	**0.0**	**0.0**	0.0	229	**0.0**	**0.0**	**0.0**	0.0
236	**0.0**	**0.0**	**0.0**	0.0	236	**0.0**	**0.0**	**0.0**	0.0
173	0.1	0.1	**0.5**	0.4	173	0.0	0.2	**0.3**	0.1
102	0.4	0.5	**0.6**	0.1	102	0.4	0.5	**0.6**	0.1
28	0.1	**0.3**	0.0	-0.3	28	0.2	**0.4**	0.1	-0.3
35	0.2	0.0	**0.3**	0.3	35	0.3	0.6	**0.7**	0.1
33	0.3	0.3	**0.7**	0.4	33	0.5	0.6	**0.9**	0.3
138	0.5	0.5	**0.9**	0.4	138	0.4	0.5	**0.8**	0.3
83	0.1	0.1	**0.3**	0.2	83	0.0	**0.3**	0.1	-0.2
6	**0.6**	**0.6**	0.3	-0.3	6	0.6	**0.6**	0.3	-0.3
114	**0.6**	0.4	0.5	0.1	114	**0.6**	0.4	0.5	0.1
22	**1.0**	**1.0**	**1.0**	0.0	22	0.8	0.6	**0.8**	0.2
131	**0.8**	0.6	**0.8**	0.2	131	0.7	0.7	**1.0**	0.3
54	**0.0**	**0.0**	**0.0**	0.0	54	**1.0**	**1.0**	**1.0**	0.0
249	**1.0**	**1.0**	**1.0**	0.0	249	**0.0**	**0.0**	**0.0**	0.0
53	**0.0**	**0.0**	**0.0**	0.0	53	**0.7**	0.3	0.6	0.3
59	0.3	**0.5**	0.1	-0.4	59	0.0	**0.3**	0.1	-0.2
62	**0.4**	0.3	**0.4**	0.1	62	**0.6**	0.3	0.4	0.1
	0.404	0.432	**0.452**	+0.02		0.448	**0.5**	0.496	-0.004

model classifying instances that were previously considered anomalous as more normal, while the control group continues to identify them as highly anomalous.

Let us revisit Table 5. For the density-based approach (VABADd) and augmentation, data set nr. 197 shows the worst performance decrease in the same realm as the reconstruction-based approach (VABADt). In both cases, this data set shows the worst performance. Similarly to the reconstruction-based approach, augmentation in combination with density estimation works best for blood pressure time series where it improves the performance of the reconstruction error-based approach for the same data sets (33, 35, 138) and also fails to improve the detection of the anomaly in data set 28 depicted in Figure 15. Additionally, in 7 out of 25 cases the performance decreases with augmentation and density-based scoring, in 10 out of 25 cases the performance increases and in 8 cases performances are on par.

IX. DISCUSSION AND CONCLUSION

The traditional assumption in reconstruction-based paradigms for anomaly detection (such as the use of autoencoders) is that reconstruction errors for anomalous data are larger than for normal data. However, in our paper we show that this is not necessarily the case.

To address this deficiency, we propose the incorporation of kernel density estimation based on the histogram of reconstruction errors. **The utilization of density estimation enhances model performance over reconstruction error-based approaches, yielding an improvement of 8.4% and 9.6% for traditional and variational autoencoder architectures, respectively.**

Reconstruction error-based methodologies, as exemplified in studies such as [17], [18], [19], [20], [21], [22], and [23], may therefore potentially derive benefits from the proposed approach. These traditional methods typically adhere to the standard threshold and reconstruction error-based paradigm.

Notably, our attention is drawn to [21], where our study identified enhancements in gait-related anomaly detection. While the study primarily focused on human gait, there exists a prospect for similar improvements in the context of cattle gait.

Furthermore, given that the time series employed in this research exhibit periodicity, exploring the adaptability of our approach to predominantly non-periodic time series, such as telemetry data in cyber security as investigated in [20], presents an intriguing avenue. While we have demonstrated an overall improvement in anomaly detection, the applicability of the proposed approach necessitates thorough evaluation on a case-by-case basis.

In evaluating the efficacy of our proposed methodology, we sought to benchmark its performance against state-of-the-art implementation and relevant references within the field. To this end, we draw upon the comprehensive study conducted by [25], wherein the authors systematically compared six distinct approaches (comprising three classical machine learning and three deep learning methodologies). Specifically, the evaluated methods included a classical autoencoder, TranAD (Transformer Networks for Anomaly Detection) [48], and GANF (Graph-Augmented Normalizing Flows for Anomaly Detection) [49]. Notably, the top-performing approach in their study achieved a score of 0.47, a benchmark we successfully surpassed by approximately 3%.

It is essential to highlight that our comparison is nuanced due to variations in the experimental setup. While [25] employed the entire benchmark dataset, we focused our analysis on a more restricted subset of 25 instances. We acknowledge the limitations associated with the subset utilization and the ensuing caution warranted when drawing direct comparisons with their results.

An alternative paradigm introduced by [28], who showcased a similar performance metric, achieved a score of 0.47 with their AER (Autoencoder with Regression) architecture – a composite model combining a vanilla autoencoder with an LSTM regressor. Noteworthy is the fact that this approach outperformed five alternative methodologies in their study. In our comparison, our methodology demonstrated an improvement of approximately 3%, yet we exercise caution in directly contrasting these findings due to the dissimilarities in benchmark subsets employed.

Despite our subset-specific evaluation, the observed performance gains suggest broader applicability of our methodology. The improved anomaly detection within our analyzed subset hints at its potential effectiveness across the entire benchmark, with due consideration for experimental variations. We see potential synergy with AER [28], given its combined regression and reconstruction approaches, which aligns with our demonstrated enhancements. This suggests an avenue for collaboration, highlighting the adaptability of our methodology with existing models in the anomaly detection landscape.

Additionally, we address the issue of data sparsity described in Section IV-A by implementing a generative approach for data augmentation. **For reconstruction-based approaches, data augmentation improves performance by 2%**. However, it should be noted that the efficacy of this approach appears to depend on the specific domain and type of anomaly. Despite these limitations, the results of this study suggest that further investigation into generative data augmentation methods are a promising area of future study.

Additionally, our research findings open a research avenue that centers on actively shaping the output distribution of reconstruction errors. This proactive approach could hold substantial potential for enhancing the inferential capabilities of the postprocessing phase. Unlike the approach applied in this work, which retroactively applies kernel density estimation to fit the data distribution, this approach advocates for the deliberate transformation of reconstruction error distributions during training towards more conventional probability density distributions.

By focusing on this proactive strategy, researchers can unlock greater potential for inference, enabling a broader scope of analytical insights and applications. This approach extends the boundaries of what can be gleaned from the data, thus warranting in-depth exploration and investigation as a valuable and research direction.

Furthermore, our study underscores the notion that the trajectory of research need not exclusively emphasize the construction of increasingly intricate and potent systems. Rather, it elucidates the substantial potential for refinement and augmentation of existing methodologies through dedicated attention to the postprocessing phase.

REFERENCES

[1] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," *ACM Comput. Surv.*, vol. 54, no. 3, pp. 1–33, Apr. 2021, doi: 10.1145/3444690.

[2] A. A. Cook, G. Misirli, and Z. Fan, "Anomaly detection for IoT time-series data: A survey," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6481–6494, Jul. 2020.

[3] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proc. MLSDA 2nd Workshop Mach. Learn. Sensory Data Anal.* New York, NY, USA: Association for Computing Machinery, Dec. 2014, pp. 4–11, doi: 10.1145/2689746.2689747.

[4] C. Yin, S. Zhang, J. Wang, and N. N. Xiong, "Anomaly detection based on convolutional recurrent autoencoder for IoT time series," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 1, pp. 112–122, Jan. 2022.

[5] A.-C. Sima, K. Stockinger, K. Affolter, M. Braschler, P. Monte, and L. Kaiser, "A hybrid approach for alarm verification using stream processing, machine learning and text analytics," in *Proc. 23rd Int. Conf. Extending Database Technol.*, Mar. 2018, pp. 26–29.

[6] C. Lehmann, L. Goren Huber, T. Horisberger, G. Scheiba, A. C. Sima, and K. Stockinger, "Big data architecture for intelligent maintenance: A focus on query processing and machine learning algorithms," *J. Big Data*, vol. 7, no. 1, pp. 1–26, Dec. 2020.

[7] S. Holzer and K. Stockinger, "Detecting errors in databases with bidirectional recurrent neural networks," in *Proc. 25th Int. Conf. Extending Database Technol.*, Apr. 2022, pp. 364–367.

[8] K. Choi, J. Yi, C. Park, and S. Yoon, "Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines," *IEEE Access*, vol. 9, pp. 120043–120065, 2021.

[9] K.-H. Lai, D. Zha, J. Xu, Y. Zhao, G. Wang, and X. Hu, "Revisiting time series outlier detection: Definitions and benchmarks," in *Proc. 34th Conf. Neural Inf. Process. Syst. Datasets Benchmarks Track*, 2021, pp. 1–13. [Online]. Available: https://openreview.net/forum?id=r8IvOsnHchr

[10] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, Mar. 2021, doi: 10.1145/3439950.

[11] S. Schmidl, P. Wenig, and T. Papenbrock, "Anomaly detection in time series: A comprehensive evaluation," *Proc. VLDB Endowment*, vol. 15, no. 9, pp. 1779–1797, Jul. 2022, doi: 10.14778/3538598.3538602.

[12] J. Kim, A. Sim, J. Kim, and K. Wu, "Botnet detection using recurrent variational autoencoder," in *Proc. IEEE Global Commun. Conf.*, 2020, pp. 1–6.

[13] B. K. Iwana and S. Uchida, "An empirical survey of data augmentation for time series classification with neural networks," *PLoS ONE*, vol. 16, no. 7, Jul. 2021, Art. no. e0254841, doi: 10.1371/journal.pone.0254841.

[14] A. Mikolajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *Proc. Int. Interdiscipl. PhD Workshop (IIPhDW)*, May 2018, pp. 117–122.

[15] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[16] R. Wu and E. J. Keogh, "Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 3, pp. 2421–2429, Mar. 2023.

[17] R.-J. Hsieh, J. Chou, and C.-H. Ho, "Unsupervised online anomaly detection on multivariate sensing time series data for smart manufacturing," in *Proc. IEEE 12th Conf. Service-Oriented Comput. Appl. (SOCA)*, nNov. 2019, pp. 90–97.

[18] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1544–1551, Jul. 2018.

[19] A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante, and K. Veeramachaneni, "TadGAN: Time series anomaly detection using generative adversarial networks," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2020, pp. 33–43.

[20] S. Zavrak and M. Iskefiyeli, "Flow-based intrusion detection on software-defined networks: A multivariate time series anomaly detection approach," *Neural Comput. Appl.*, vol. 35, no. 16, pp. 12175–12193, Jun. 2023, doi: 10.1007/s00521-023-08376-5.

[21] K. Zhang, S. Han, J. Wu, G. Cheng, Y. Wang, S. Wu, and J. Liu, "Early lameness detection in dairy cattle based on wearable gait analysis using semi-supervised LSTM-autoencoder," *Comput. Electron. Agricult.*, vol. 213, Oct. 2023, Art. no. 108252. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169923006403

[22] K. A. Alaghbari, M. H. Md. Saad, A. Hussain, and M. R. Alam, "Activities recognition, anomaly detection and next activity prediction based on neural networks in smart homes," *IEEE Access*, vol. 10, pp. 28219–28232, 2022.

[23] B. Kim, M. A. Alawami, E. Kim, S. Oh, J. Park, and H. Kim, "A comparative study of time series anomaly detection models for industrial control systems," *Sensors*, vol. 23, no. 3, p. 1310, Jan. 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/3/1310

[24] V. L. Cao, M. Nicolau, and J. McDermott, "A hybrid autoencoder and density estimation model for anomaly detection," in *Parallel Problem Solving from Nature—PPSN XIV*. Berlin, Germany: Springer, 2016, pp. 717–726.

[25] F. Rewicki, J. Denzler, and J. Niebling, "Is it worth it? Comparing six deep and classical methods for unsupervised anomaly detection in time series," *Appl. Sci.*, vol. 13, no. 3, p. 1778, Jan. 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/3/1778

[26] B. Barz, E. Rodner, Y. G. Garcia, and J. Denzler, "Detecting regions of maximal divergence for spatio-temporal anomaly detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 5, pp. 1088–1101, May 2019.

[27] T. Nakamura, M. Imamura, R. Mercer, and E. Keogh, "MERLIN: Parameter-free discovery of arbitrary length anomalies in massive time series archives," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2020, pp. 1190–1195.

[28] L. Wong, D. Liu, L. Berti-Equille, S. Alnegheimish, and K. Veeramachaneni, "AER: Auto-encoder with regression for time series anomaly detection," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Los Alamitos, CA, USA, Dec. 2022, pp. 1152–1161, doi: 10.1109/BigData55660.2022.10020857.

[29] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proc. KDD Workshop*, 1994, pp. 359–370.

[30] O. I. Provotar, Y. M. Linder, and M. M. Veres, "Unsupervised anomaly detection in time series using LSTM-based autoencoders," in *Proc. IEEE Int. Conf. Adv. Trends Inf. Theory (ATIT)*, Dec. 2019, pp. 513–517.

[31] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, J. Chen, Z. Wang, and H. Qiao, "Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications," in *Proc. World Wide Web Conf.*, 2018, pp. 187–196, doi: 10.1145/3178876.3185996.

[32] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[33] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, no. 3, pp. 1065–1076, Sep. 1962, doi: 10.1214/aoms/1177704472.

[34] B. Silverman, *Density Estimation for Statistics and Data Analysis*. London, U.K.: Chapman, 1986.

[35] D. W. Scott, "On optimal and data-based histograms," *Biometrika*, vol. 66, no. 3, p. 605, Dec. 1979.

[36] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds. Red Hook, NY, USA: Curran Associates, 2020, pp. 18661–18673. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf

[37] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Anal. Chem.*, vol. 36, no. 8, pp. 1627–1639, Jul. 1964, doi: 10.1021/ac60214a047.

[38] S. A. N. Laptev and Y. Billawala. (2015). *S5—A Labeled Anomaly Detection Dataset, Version 1.0 (16m)*. [Online]. Available: https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70

[39] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. New York, NY, USA: Association for Computing Machinery, Jul. 2019, pp. 2828–2837, doi: 10.1145/3292500.3330672.

[40] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 387–395, doi: 10.1145/3219819.3219845.

[41] E. Keogh, "MiLeTS'21: 7th KDD workshop on mining and learning from time series," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 63–64. [Online]. Available: https://kdd-milets.github.io/milets2021/slides/Irrational%20Exuberance_Eammon_Keogh.pdf

[42] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," 2019, *arXiv:1912.01703*.

[43] R. Cahuantzi, X. Chen, and S. Güttel, "A comparison of LSTM and GRU networks for learning symbolic sequences," in *Intelligent Computing*, K. Arai, Ed. Cham, Switzerland: Springer, 2023, pp. 771–785.

[44] B. Athiwaratkun and J. W. Stokes, "Malware classification with LSTM and GRU language models and a character-level CNN," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 2482–2486.

[45] S. Yang, X. Yu, and Y. Zhou, "LSTM and GRU neural network performance comparison study: Taking yelp review dataset as an example," in *Proc. Int. Workshop Electron. Commun. Artif. Intell. (IWECAI)*, Jun. 2020, pp. 98–101.

[46] T. Odland, "Tommyod/kdepy: Kernel density estimation in Python," Version v0.9.10, Zenodo, Eur. Org. Nuclear Res., IT Dept., Digit. Repositories Sect., Genève, Switzerland, Dec. 2018, doi: 10.5281/zenodo.2392268.

[47] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," 2017, *arXiv:1712.04621*.

[48] S. Tuli, G. Casale, and N. R. Jennings, "TranAD: Deep transformer networks for anomaly detection in multivariate time series data," *Proc. VLDB Endowment*, vol. 15, no. 6, pp. 1201–1214, Feb. 2022, doi: 10.14778/3514061.3514067.

[49] E. Dai and J. Chen, "Graph-augmented normalizing flows for anomaly detection of multiple time series," in *Proc. Int. Conf. Learn. Represent.*, 2022, pp. 1–16. [Online]. Available: https://openreview.net/forum?id=45L_dgP48Vd

ROBIN FREHNER received the degree in computer science from Zurich University of Applied Sciences, Switzerland, where he is currently pursuing the master's degree in data science. He was a Visiting Student with UC Berkeley and a Research Assistant with the Fraunhofer Institute. His primary research interest includes applying machine learning to anomaly detection in complex systems.

KESHENG WU (Senior Member, IEEE) is currently a Senior Computer Scientist with the Lawrence Berkeley National Laboratory. He works extensively on data management, data analysis, and scientific computing. He is the developer of a number of widely used algorithms, including FastBit bitmap indexes for querying large scientific datasets, Thick-Restart Lanczos (TRLan) algorithm for solving eigenvalue problems, and IDEALEM for statistical data reduction and feature extraction.

ALEXANDER SIM (Senior Member, IEEE) is currently a Senior Computing Engineer with the Lawrence Berkeley National Laboratory. He has authored or coauthored more than 350 technical publications and released a few software packages under open source license. His current research and development interests include data modeling, data analysis methods, learning models, distributed data infrastructure, dynamic resource management, and high-performance data systems.

JINOH KIM (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Minnesota, Twin Cities. He is currently an Associate Professor of computer science with Texas A&M University–Commerce. He is also an Affiliate Faculty Scientist with the Lawrence Berkeley National Laboratory (LBNL) and a member of the Silicon Valley Cybersecurity Institute (SVCSI). His main research interests include networked systems and distributed computing focusing on performance, reliability, scalability, visibility, and security, using machine intelligence and algorithmic methodologies.

KURT STOCKINGER received the Ph.D. degree in computer science from CERN, University of Vienna. He is currently a Professor of computer science and the Director of Studies in Data Science with Zurich University of Applied Sciences (ZHAW) and the Co-Head of the ZHAW Datalab. He is also an External Lecturer with the University of Zurich. Previously, he was with Credit Suisse (Schweiz), Zurich, Switzerland, the Lawrence Berkeley National Laboratory, Berkeley, California, California Institute of Technology, California, and CERN, Geneva, Switzerland. His research focuses on data science, with an emphasis on big data, natural language query processing, query optimization, and quantum computing. Essentially, his research interests include the intersection of databases, natural language processing, and machine learning.