algorithms

MDPI

Article

Unsupervised Transformer-Based Anomaly Detection in ECG Signals

Abrar Alamr * and Abdelmonim Artoli

Computer Science Department, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia; aartoli@ksu.edu.sa
* Correspondence: 438204241@student.ksu.edu.sa

Abstract: Anomaly detection is one of the basic issues in data processing that addresses different problems in healthcare sensory data. Technology has made it easier to collect large and highly variant time series data; however, complex predictive analysis models are required to ensure consistency and reliability. With the rise in the size and dimensionality of collected data, deep learning techniques, such as autoencoder (AE), recurrent neural networks (RNN), and long short-term memory (LSTM), have gained more attention and are recognized as state-of-the-art anomaly detection techniques. Recently, developments in transformer-based architecture have been proposed as an improved attention-based knowledge representation scheme. We present an unsupervised transformer-based method to evaluate and detect anomalies in electrocardiogram (ECG) signals. The model architecture comprises two parts: an embedding layer and a standard transformer encoder. We introduce, implement, test, and validate our model in two well-known datasets: ECG5000 and MIT-BIH Arrhythmia. Anomalies are detected based on loss function results between real and predicted ECG time series sequences. We found that the use of a transformer encoder as an alternative model for anomaly detection enables better performance in ECG time series data. The suggested model has a remarkable ability to detect anomalies in ECG signal and outperforms deep learning approaches found in the literature on both datasets. In the ECG5000 dataset, the model can detect anomalies with 99% accuracy, 99% F1-score, 99% AUC score, 98.1% recall, and 100% precision. In the MIT-BIH Arrhythmia dataset, the model achieved an accuracy of 89.5%, F1 score of 92.3%, AUC score of 93%, recall of 98.2%, and precision of 87.1%.

Keywords: unsupervised transformers; deep learning; anomaly detection; ECG signal

check for updates

Citation: Alamr, A.; Artoli, A. Unsupervised Transformer-Based Anomaly Detection in ECG Signals. *Algorithms* **2023**, *16*, 152. https://doi.org/10.3390/a16030152

Academic Editor: Frank Werner

Received: 12 February 2023
Revised: 28 February 2023
Accepted: 7 March 2023
Published: 9 March 2023

1. Introduction

The detection of anomalies is important to many contemporary applications and continues to be of paramount importance with the explosion of sensor use [1] Anomaly detection in electrocardiogram (ECG) time series data has recently received considerable attention due to its impact on controlling the quality of ECG time series processes and identifying abnormal data source behavior [2,3]. The process of anomaly detection in time series data involves the use of complicated algorithms and models to detect anomalous data within a selected period. An effective anomaly detector can recognize the contrasts between normal and anomalous time series data [4,5].

As the demand for real-time anomaly detection is increasing nowadays, the necessity for intelligent, robust, and computationally efficient models has been realized and is beginning to gain more attention in most live applications [6]. These models play a critical role in most time series applications due to the inevitability of error incidence. The properties of time series data are critical for selecting the appropriate approach to designing a suitable anomaly detector [7] Successful examples of anomaly detectors identify anomalies by measuring statistical deviations in time series data, such as the autoregressive integrated moving average (ARIMA) [8], cumulative sum statistics (CUSUM) [9], and exponentially weighted moving average (EWMA) [10]. However, traditional time series anomaly detection methods, on the other hand, suffer from a lack of the model's expected efficiency and accuracy [11].

Recently, several intelligent computing methods for anomaly detection have been developed. Of these, deep learning and neural networks are trending algorithms known to be accurate and efficient [12]. However, beginning in the late 2000s, interest grew in identifying anomalies in large volumes and highly variant online time series data [13]. Aside from commonly used techniques [14] such as autoencoders (AEs), recurrent neural networks (RNNs), convolutional neural network (CNN) [15], and long short-term memory (LSTM), contemporary deep neural architectures are typically utilized for anomaly detection and healthcare prediction [16–18]. Most of these methods have inherent characteristics that hinder their use when dealing with time series data, although they have evolved to overcome previously proposed techniques. For instance, by using multiplicative gates that impose constant error flow through the internal states of special units called memory cells, LSTM neural networks overcome the vanishing gradient problem faced by RNNs [19] In addition, LSTM networks obviate the need for a pre-specified time window and can model complex multivariate sequences accurately because of their capacity to learn long-term correlations in a series. However, the primary objective of these methods is to produce more realistic sequences instead of extracting meaningful features that facilitate downstream tasks.

At present, A transformer architecture that follows the procedure of prediction was first introduced as an efficient alternative to recurrent neural network RNNs in natural language processing (NLP) [20]. Therefore, in this work, we introduce a transformer architecture-based anomaly detection model to detect anomalies in human heartbeat time series signals, such as premature ventricular contraction (PVC), supraventricular premature (SP) or ectopic beat (EB), and other ECG anomalies. Here, we used the transformer encoder introduced in [20] to develop novel unsupervised transformer anomaly detection. The model learns the distribution pattern of normal data and detects anomalies by comparing the loss function between the predicted data and the original data. The proposed model outperforms current state-of-the-art deep learning modeling approaches for detecting ECG time series data.

In the rest of this article, we survey the existing literature on deep leaning anomaly detection in ECG time series in a related work section. The proposed model and datasets used are presented in the materials and methods section. The experimental setup and results analysis are shown in the results and discussion sections. Finally, the paper ends with a conclusion.

2. Related Work

We present some of the existing deep-learning models that are used for ECG time series anomaly detection. The authors in [21] have used LSTM unit neural network architecture to construct a predictive model for healthy ECG signals. An added benefit of using LSTM networks is that, as required by other techniques, the ECG signal can be fed directly into the network without any elaborate preprocessing. The findings are optimistic and show that LSTM models could be feasible for detecting ECG signal anomalies. In addition, the authors in [18] proposed an encoder-decoder framework for anomaly detection based on LSTM networks that learns to reconstruct normal time series behavior using reconstruction error to detect anomalies on three predictable time series datasets: power demand, space shuttle, and ECG. They showed that the model is robust and can detect anomalies from predictable, unpredictable, periodic, aperiodic, and quasi-periodic time series. Remarkably, the authors in [22] combined AE with LSTM using ECG data from a patient with a myocardial infarction to show that the system can accurately classify an irregular wave interval. In [23] stacked LSTM networks were used for anomaly detection in a time series. The efficacy of this approach was demonstrated on four datasets: ECG, space shuttle, power demand, and a multisensor engine dataset. In [24], the authors focused on the identification of time series data anomalies using a fusion model of LSTM and GAN, namely LSTM-GAN. The authors verified the algorithm's output using two sets of time series data. The experimental results demonstrated that compared to traditional algorithms, LSTM-GAN achieved superior performance in processing time series data. Schlegl et al. [25] developed the Deep Convolutional Generative Adversarial Nets (DC-GAN) based on Anomaly Detection Generative Adversarial Nets (AnoGAN) model for Anomaly detection. As an unsupervised learning model, AnoGAN uses normal data for learning. By comparing with Query data, it is possible to detect an Anomaly. In the method, a decision boundary is subjective. Therefore, through repeated experiments, it is necessary to apply a decision boundary according to conditions. LSTM can also use in unsupervised learning such as [26], the authors used MIT-BIH arrhythmia dataset to detect anomaly in ECG signal in unsupervised learning manner.

However, few models use transformer-based architecture for time series anomaly detection. In [27], the authors proposed transformer-based with generative adversarial networks (GAN) for anomaly detection of time series data. The transformer-based generators can extract contextual features of time series data to prompt performance. In the training and anomaly detection stages, the authors used two encoders and two decoder transformer blocks. They showed that the model has better performance in anomaly detection than state-of-the-art anomaly detection techniques using three datasets: Secure Water Treatment (SWaT), Water Distribution (WADI), and KDD Cup 1999. In [28] the authors used transformer architecture in time series data; the model consists of three encoder blocks and one decoder block. The input time series data were split into train sequences and label sequences, of which train sequences were fed into the encoder, while label sequences were fed to the decoder. They also replaced the original multihead self-attention method with a multibranch attention mechanism. The F-scores for the WADI and SWaT datasets were 0.84 and 0.91, respectively.

From the above discussion, we realize that transformer-based anomaly detection methods have great potential to increase both accuracy and performance if they are used in detecting ECG anomalies, due to transformer effectiveness and capacity of simultaneously obtaining long-distance context data. We will be using transformer approach for anomaly detection in ECG time series data as an unsupervised learning model which has not been used in this area before. However, the work by [29], and a few recently published ones have considered end-to end learning method which requires huge labeling as a characteristic of supervised learning. For time series data, labelled data may not be always available. Our work considers only unsupervised learning for better adaptation to the input signals. Furthermore, our model is different from the existing models since it consists of two standard encoder transformer layers without a decoder. Therefore, in this work, we will investigate the capability of transformer encoder in anomaly detection in ECG time series as an unsupervised learning model and compare our findings with state-of-the-art deep learning models in accuracy and F1-scores, as detailed subsequently.

3. Materials and Methods

3.1. ECG Time Series Data

A time series is a collection of data points that are organized chronologically. Most commonly, a time series is a sequence taken at successive similarly spaced points in time. In this work, we used the ECG5000 dataset [30] and the MIT-BIH Arrhythmia dataset [31] to validate the proposed model. Both datasets are in the ECG domain.

3.1.1. The ECG5000 Dataset

We used a five-thousand ECG (ECG5000) time series dataset [30] in our experiment. The original data were obtained from PhysioNet's BIDMC Congestive Heart Failure Database (CHFDB). In two processes, the data were pre-treated: each heartbeat was first extracted, and then each heartbeat was interpolated to be the same length, equaling 140 time steps [32]. According to Figure 1, there are five different heartbeat types: normal (N), R-on-T VPC (R-on-T), PVC, SP, or EB, and unclassified beat (UB).

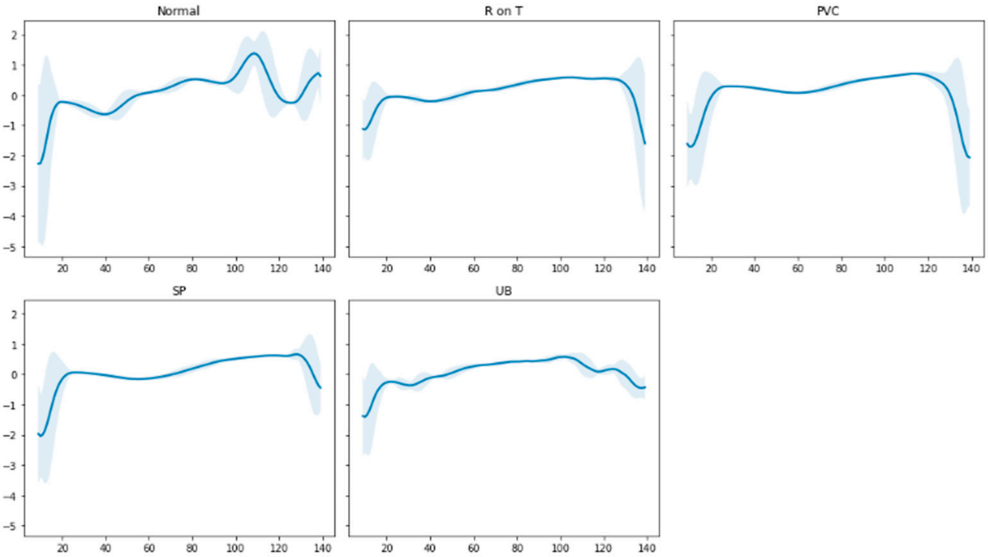


Figure 1. Time series nature of the signals of different types of heartbeats on ECG5000.

The ECG5000 dataset was created by selecting 5000 sequences from the extrapolated ECG, of which 2989 sequences were normal, and the remaining 2011 sequences were anomalous. For our proposed model, the data were divided into three sets—training, validation, and testing—in an 80:10:10 ratio. Since we applied unsupervised learning, we eliminated anomalous data from the training and validation data. In the training stage, the model learned the latent space of normal heartbeats. Table 1 lists the number of sequences in each dataset.

Table 1. Overview of each set number of sequences on the ECG5000 dataset.

Dataset	Normal	Anomalous	Total
Train data	2335	0	2335
Validation data	292	0	292
Test data	266	234	500

3.1.2. The MIT-BIH Arrhythmia Database

The MIT-BIH Arrhythmia database is a clinical database [31] that includes two-channel ECG recordings of 48 patients at the Beth Israel Hospital (BIH) Arrhythmia Laboratory. Each recording was half an hour long. Modified limb lead II signals were chosen for this study. Following the advice of the Association for the Advancement of Medical Instrumentation (AAMI) [33] five categories were used to categorize the chosen heartbeats as follows: normal beat (N), supraventricular ectopic (S), ventricular ectopic (V), fusion beat (F), and unknown beat (Q). According to the AAMI suggestions, we chose 44 of the 48 recordings for our experiment and removed four recordings (102, 104, 107, and 217) since they were of poor quality.

Preprocessing

Signal preprocessing is an essential step, especially when dealing with physiological data, such as the ECG, taking into account all potential sources of noise, such as motion artifacts and power line interference, that can impair the performance of any ensuing models. Our preprocessing stage contains two steps. First, we applied filter to eliminate noise and then we extracted the heartbeat. We have tried different filters such as, Butterworth bandpass filter, bandpass filter and one median filter, we found that two median filters perform better. The following preprocessing procedures were used in this work:

1. Median Filter: We used a median filter with a 200-ms sliding window. Then, using a 600-ms window, we applied a second median filter. The baseline of the raw signals was contained in the second filter's output. The second filter output was subtracted from the unprocessed ECG data to eliminate the baseline wander (see Figure 2). This step enhanced the baseline correction and eliminated some artifacts [34].
2. Heartbeat Extraction: This entails picking a neighborhood around each beat. This interval was estimated using R-peak annotation with ±50 ms before and after the beat.

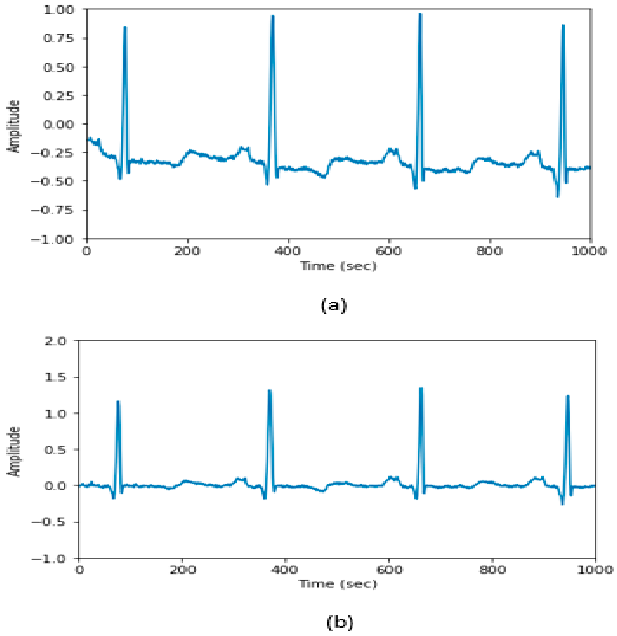


Figure 2. Raw signal data (**a**) before the median filter and (**b**) after applying the median filter.

After heartbeat extraction, the total number of normal sequences was 72,722, while the total number of anomalous sequences was 10,579. Since the number of normal sequences was large compared with anomalous sequences, we used a sample of 18,824 sequences of them. The data were divided into 80% training data and 20% testing data. The training data that was split into training data and validation data contained only N beats. Table 2 summarizes the number of sequences in each dataset.

Table 2. Overview of each set number of sequences on the MIT-BIH Arrhythmia dataset.

Dataset	Normal	Anomalous	Total
Training data	12,045	0	12,045
Validation data	3012	0	3012
Test data	3767	2115	5882

3.2. Proposed Unsupervised Transformer Architecture

In this paper, we introduce a transformer-based network for anomaly detection in ECG signals. The overall model architecture comprises two parts: an embedding layer and a standard transformer encoder (see Figure 3). All normal time series data of ECG signals, which take the form of a 2D tensor of sequence length × number of features, were first encoded into sequences of embeddings, which were then fed into a multilayer bidirectional transformer network to produce their corresponding representations. The final linear dense layers predicted the input ECG signal with the same input sequence length.

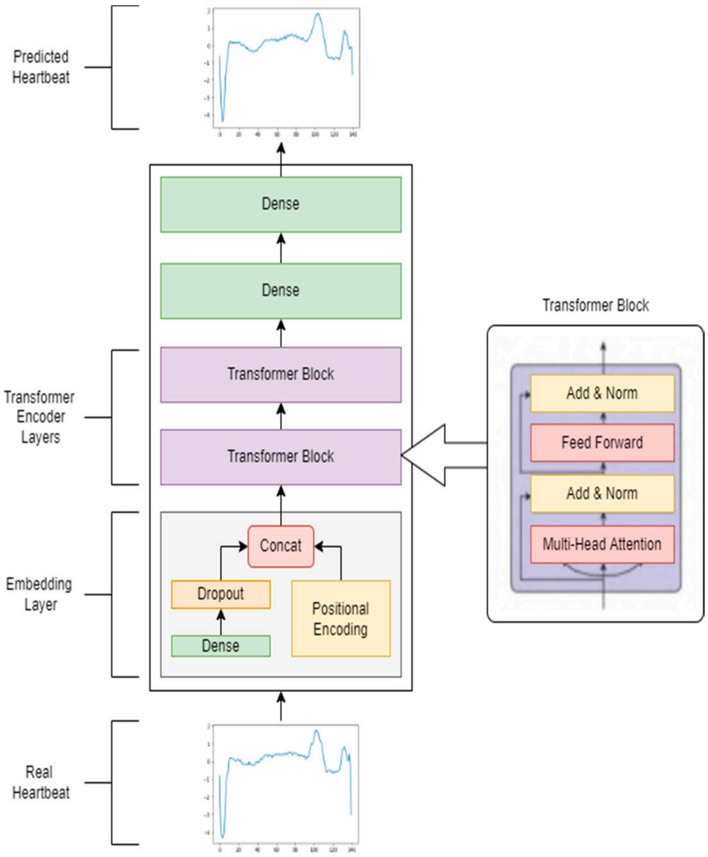


Figure 3. Structure of the unsupervised transformer architecture.

The embedding layer was introduced for higher dimensionality mapping of input tensors into a feature space. Because the input dimension was linked to the size of the transformer's hidden layer, we partitioned this layer into two parts: a linear dense layer, which projected the input into a higher-dimensional vector, and a dropout layer to avoid overfitting. Sinusoidal positional encoding was used to encode the order of the input sequences:

$$\mathrm{PE}_{(\mathrm{pos},2\mathrm{i})} = \sin\left(\mathrm{pos}/10000^{2\mathrm{i}/\mathrm{d}_{\mathrm{model}}}\right) \tag{1}$$

$$\mathrm{PE}_{(\mathrm{pos},2\mathrm{i}+1)} = \cos\left(\mathrm{pos}/10000^{2\mathrm{i}/\mathrm{d}_{\mathrm{model}}}\right) \tag{2}$$

where pos is the position, i is the dimension and $\mathrm{d}_{\mathrm{model}}$ = the embedding size of the model.

In searching for the best combination of model configurations, we tested both one and two transformer encoder blocks. The first transformer encoder block created equivalent hidden representations of embedding sequence input. Then, to iteratively produce higher-level representations, these representations were given as input to the second transformer encoder block. A single transformer block (shown in Figure 3) included two main sublayers: a position-wise fully connected feedforward network and a multihead self-attention layer (FFN). Both sublayers adopt a residual connection [35] and layer normalization [36]. As illustrated in Figure 4, each of the h parallel scaled dot-product attention layers that make up a multi-head attention layer is referred to as a head [20]. A query vector and a set of key-value pairs were transformed into an output vector using scaled dot-product attention.

$$\mathrm{Attention}(\mathrm{Q},\mathrm{K},\mathrm{V}) = \mathrm{softmax}\left(\frac{\mathrm{QK}^{\mathrm{T}}}{\sqrt{\mathrm{d}_{\mathrm{k}}}}\right)\mathrm{V} \tag{3}$$

where Q, K and V stand for the matrices stacked by several queries, keys, and value vectors as rows, respectively, and the dimension of the query/key vectors is d_{k}. Before calculating attention, multihead attention maps Q, K and V onto various lower-dimensional feature subspaces using various linear dense layers. Using an additional dense layer, the outputs from h heads are concatenated and projected onto a final hidden representation, as shown below:

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^O \tag{4}$$

where

$$\mathrm{head}_i = \mathrm{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{5}$$

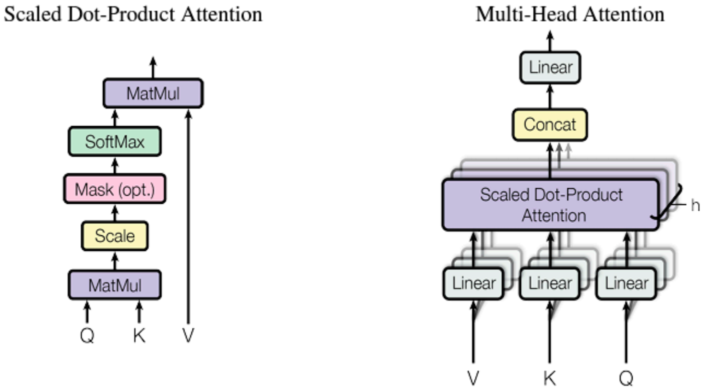


Figure 4. Illustration of the scaled dot-product attention (**left**) and multi-head attention consisting of several attention layers running in parallel (**right**).

The weight matrices of each head's inner dense layers are denoted by W_i^Q, W_i^K, and W_i^V, while that of the top dense layer is denoted by W^O. For this paper, $W_i^Q \in R^{d_{model} \times d_k}$, $W_i^K \in R^{d_{model} \times d_k}$, $W_i^V \in R^{d_{model} \times d_V}$, $W^O \in R^{d_{model} \times hd_V}$, h = number of heads, and we use $d_K = d_V = \frac{d_{model}}{h}$.

The internal relationship of an input sequence is learned by a transformer's self-attention mechanism. The query, key, and value vectors in the self-attention mechanism are all similar. In other words, attention is calculated for each location in a sequence between a given position and another position. Thus, the hidden representation of each observation captures the global sequence information and highlights the region surrounding the ith observation by weighting the sum of all the positions in a sequence.

A position-wise FFN was then applied to the hidden state of each position independently and identically. The FFN was made up of two linear transformations with a ReLU activation function between them:

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{6}$$

where x_i is the ith hidden state produced by the multi-head attention layer, W_1 and W_2 are weight matrices, and b_1 and b_2 are bias terms of the inner and output dense layers, respectively.

3.3. Anomaly Score and Threshold

Because we implemented an unsupervised schema, the model was trained on only an N heartbeat sequence. The model predicts the original values of the input sequence. The loss function for our model is the mean-absolute error (MAE) [37] between the input signal and the predicted signal as shown below in Equation (7). As it is common to use root mean square error (RMSE) such as in [18], we have first attempted to use it as a loss function. However, we found that the MAE revealed the best choice because it performs better.

$$\mathrm{loss} = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| \tag{7}$$

The MAE is the total absolute variation between each signal's actual value y_i and the model's prediction value x_i. Through training, the model learned the saliency and activation characteristics of normal signals, resulting in a lower loss prediction error. The model detects anomalies in the testing data by calculating whether the predicted loss error is greater than a fixed threshold. In this work, the threshold equals two standard deviations above the mean loss of the trained normal data as the following equation:

$$\text{Threshold} = \text{mean}\ (\text{train data loss}) + 2 \times \text{standard deviation}\ (\text{train data loss}) \tag{8}$$

After calculating the threshold using the training data, the procedure was converted into a binary classification problem. In the anomaly detection stage, the calculated threshold determines whether a data sequence is normal or abnormal by comparing its prediction error. Using testing data, a specific input sequence was categorically provided to the model, and the predicted error was then compared to the calculated threshold. The input sequence was categorized as anomalous if the predicted error exceeded the threshold; otherwise, it was categorized as normal. Figure 5 shows the graphical architectural procedure of our anomaly detection model for clarification.

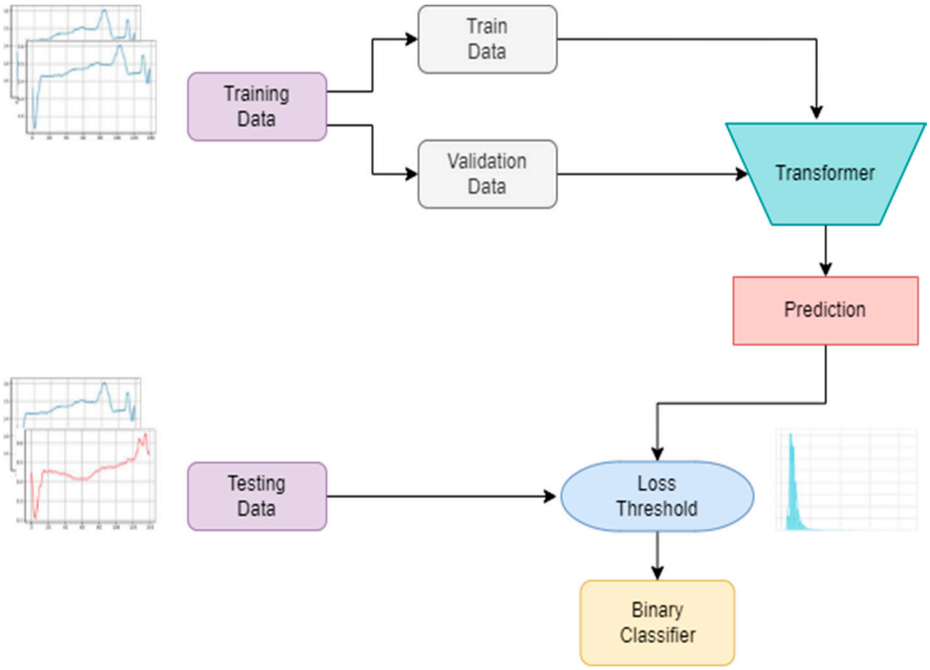


Figure 5. Architectural procedure for anomaly detection model.

4. Results and Discussion

4.1. Experimental Setup

To fine-tune our model, we performed many experiments with different numbers of transformer encoder blocks (consisting of a multihead attention layer followed by a feedforward layer), different hidden state sizes, and different numbers of attention heads. Dropout was implemented on only the first dense layer, with a dropout rate of 0:2. We use an Adam optimizer with a learning rate of 1e–4 and early stopping. The model was trained for a maximum of 70 epochs; however, all experiments converged before this point. We trained our model using only normal data in batch sizes of 16 sequences. All experiments were implemented in Python (v3.7.6) [38] using Tensorflow (v1.14.0) [39] and Keras (v2.3.1) [40] machine learning libraries. We ran all the experiments on a Windows 10 Pro 64-bit operating system with an AMD Ryzen 5 3600 6-Core processor 3.59 GHz, NVIDIA 185 GEFORCE RTX 2080 SUPER (GPU), and 32 GB installed memory.

4.2. Performance Metrics

A true positive (TP) represents accurately anticipated positive values in the binary classifier, while a true negative (TN) represents correctly predicted negative values. A false positive (FP) is a negative expected value that is positive, while a false negative (FN) is a positive projected value that is negative. The anomaly detection results of our model were evaluated using the area under the curve (AUC), accuracy, precision, recall, and F1-scores, as follows:

- AUC: The AUC is computed by building the receiver operating characteristic (ROC) curve based on the false positive (FP) and the true positive (TP).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{9}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{10}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{11}$$

$$\text{F1} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 * \text{TP}}{2 * \text{TP} + \text{TP} + \text{TN}} \tag{12}$$

4.3. ECG 5000 Dataset Results

The proposed model was trained on normal heartbeats and thus learned the distribution of normal signals. Anomalous heartbeats do not obey this distribution; thus, the model cannot accurately predict them when they were input into the model. To observe this intuitively, using testing dataset we illustrate the input time sequences and the predicted time sequences in Figure 6. This figure shows that the actual and predicted normal heartbeat signals are roughly the same (Figure 6b). However, when the input is an anomalous heartbeat signal, the model maps it to a continuous latent space of normal heartbeats, as shown in Figure 6a, which provides an opportunity for anomaly detection. Despite the fact that some abnormal samples usually resemble their normal predictions, local differences can assist in distinguishing them.

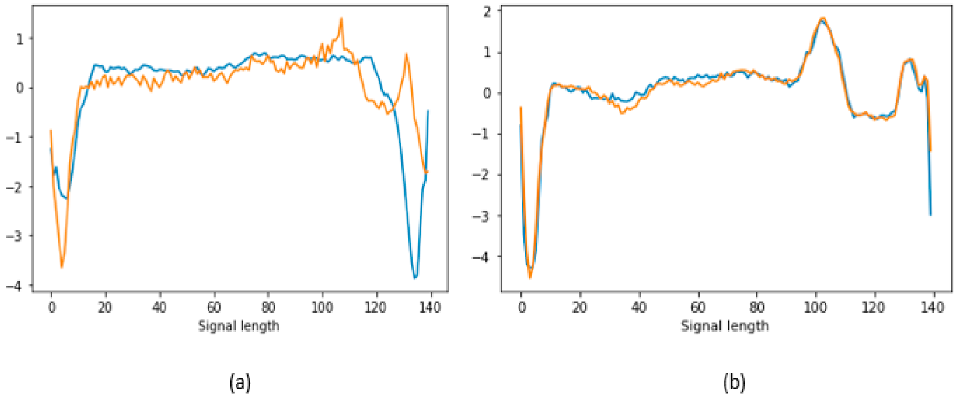


Figure 6. (**a**) Anomalous heartbeat (blue) with corresponding predicted heartbeat (orange); (**b**) normal heartbeat (blue) with corresponding predicted heartbeat (orange).

As shown in Table 3, we trained a series of unsupervised transformer networks to determine the best number of encoder blocks (1 or 2), embedding size (32, 64, 128, or 256), and number of attention heads (16 or 32). The best-performing model, which contains two encoder blocks, 128 embedding sizes, and 32 attention heads, achieved impressive results, with an F1 score of 99%, an accuracy of 99%, a recall of 98.1%, and a precision of 100%. The calculated threshold was 0.29. However, from Table 3, we can observe that a large embedding size obtained better performance from the model. Since it handles heartbeat signals at distinct time steps, our model is more suitable for sequential time series data. Figure 7, shows the ROC curve, which plots the true positive rate (TPR) against the false positive rate (FPR) at various classification thresholds to represent the model's ability to distinguish between positive and negative classes. Our model achieved an AUC score of 99% that indicating a perfect classifier model, since the higher AUC score means the better model ability in making predictions.

Table 3. Transformer anomaly detection results for the ECG5000 test dataset.

No. Encoder Blocks	No. Heads	Hidden Size	F1	Accuracy	Recall	Precision
1	16	32	96.7%	96.8%	97.7%	96.2%
1	16	64	97.6%	97.6%	96.9%	98.4%
1	16	128	98%	98%	96.9%	99.2%
1	16	256	98.2%	98.2%	96.6%	100%
1	32	32	98.4%	98.4%	96.9%	100%
1	32	64	97%	97%	96.6%	97.7%
1	32	128	98.4%	98.4%	97.7%	99.2%
1	32	256	98.8%	98.8%	97.7%	100%
2	16	32	98.2%	98.2%	97.3%	99.2%
2	16	64	98.4%	98.4%	97.3%	96.6%
2	16	128	98.6%	98.6%	97.3%	100%
2	16	256	98.4%	98.4%	97.3%	99.6%
2	32	32	98.2%	98.2%	96.9%	99.6%
2	32	64	98.6%	98.6%	97.3%	100%
2	32	128	99%	99%	98.1%	100%
2	32	256	98.2%	98.2%	96.9%	99.6%

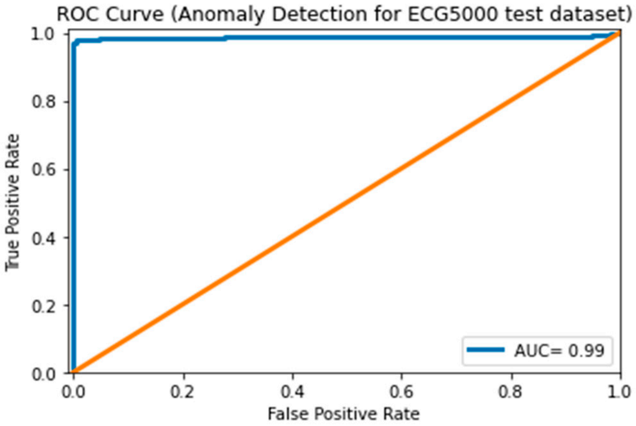


Figure 7. ROC curve of ECG 5000 testing set.

Since the ECG5000 dataset has already been used in previous studies, evaluating our model against other supervised and unsupervised methods is crucial to understanding its benefits and drawbacks, as well as its place in the anomaly detection field. The outcomes of a few supervised and unsupervised algorithms on the same dataset are shown in Table 4. In [41] and [42] the dataset splitting was 4500 heartbeats (80%) held for testing and 500 (20%) for training and validation tasks (20%). In [41] the authors used the Variational Recurrent Autoencoder (VRAE) to represent the data and then applied clustering and the Wasserstein distance to detect the anomaly, successfully detecting anomaly claiming outperformance over existing supervised and unsupervised methods on the ECG5000 dataset. Similarly, the authors in [42] used the same Variational Autoencoder (VAE) method coupled with a local similarity score on the two datasets we used in this work, namely ECG5000 and MIT-BIH Arrhythmia, achieving a comparable AUC with the literature. Nevertheless, our model outperforms these deep learning models in all evolution metrics, with an F1 score of 99%, an accuracy of 99%, a recall of 98.1%, and a precision of 100%. However, the comparison was considered unfair because we used different data-splitting methods. Therefore, we compared our results to those obtained recently by [43] and [44] because the data was split into 80% training 10% validation and 10% testing, exactly as we did here. The authors of [43] Concat Attention Autoencoder (CAT-AE), AE-without-Attention, and VAE claim state-of-the-art accuracy and precision in anomaly detection. On the other hand, the authors in [44] used an unsupervised LSTM AE. As shown in Table 4, our model outperforms these deep-learning models using the same data-splitting methods.

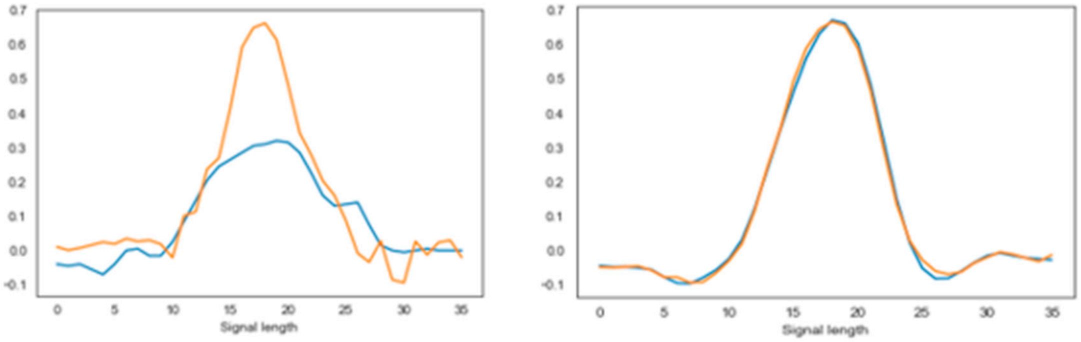
Table 4. Results of the ECG5000 test dataset.

Model	S/U	Accuracy	Recall	Precision	F1-Score
Hierarchical [41]	U	95.5%	94.6%	95.8%	94.6%
Spectral [41]	U	95.8%	95.1%	94.7%	94.7%
VRAE + Wasserstein [41]	U	95.1%	94.6%	94.6%	94.6%
VRAE + k-Means [41]	U	95.9%	95.3%	95.4%	95.2%
VAE [42]	U + S	96.8%	—	—	95.7%
VAE [43]	S	95.2%	92.5%	98.4%	95.4%
AE-Without-Attention [43]	S	97%	95.5%	98.8%	97.1%
CAT-AE [43]	S	97.2%	95.6%	99.2%	97.4%
LSTM AE [44]	U	97.93%	—	—	—
This work	U	99%	98.1%	100%	99%

For evidence, symbols (S,U) denote the supervised and unsupervised learning, respectively.

4.4. MIT-BIH Arrhythmia Dataset Results

Examples of abnormal and normal heartbeats, together with their corresponding predictions, are shown in Figure 8. Normal heartbeats appear to be rather accurately predicted by the model, as shown in Figure 8b. There is little doubt that the model has picked up on the basic morphological behavior of a normal cardiac cycle pattern. The model can adjust the output to the input as much as the normal latent space allows, while continuously aiming to minimize prediction loss. The model attempts to forecast the input using only the normal properties of the cardiac cycle so abnormal heartbeats are mapped onto a normal heartbeat latent space. As expected, this results in lower prediction quality and higher prediction loss for abnormal heartbeats, as shown in Figure 8a.



(**a**) (**b**)

Figure 8. (**a**) Anomalous heartbeat (blue) with corresponding predicted heartbeat (orange); (**b**) normal heartbeat (blue) with corresponding predicted heartbeat (orange).

As shown in Table 5, we trained multiple transformer-encoder networks to determine the best number of attention-feedforward blocks (1 or 2), embedding size (512 or 1024 neurons), and number of attention heads (16 or 32). As can be seen in Table 5 the transformer-encoder anomaly detection results for the MIT-BIH test dataset for different models were close together. However, the best model configurations contained two blocks, 64 embedding sizes, and 32 attention heads. Our best-performing transformer-encoder model achieved an accuracy of 89.5%, F1 score of 92.3%, recall of 98.2%, and precision of 87.1%. Figure 9 presents the ROC curve of the best model on the testing set with an AUC score of 93% which means that the model is performing relatively well in distinguishing between anomaly and normal signals. The calculated threshold value was 0.12.

Table 5. Transformer anomaly detection results for the MIT-BIH test dataset.

No. Encoder Blocks	No. Heads	Hidden Size	F1	Accuracy	Recall	Precision
1	16	32	91.6%	88.6%	97.5%	86.3%
1	16	64	91.9%	89%	98.3%	86.3%
1	16	128	91.8%	88.7%	98.1%	86.2%
1	16	256	91.6%	88.4%	98.4%	85.6%
1	32	32	92.1%	89.2%	98.3%	86.6%
1	32	64	92.1%	89.2%	98.3%	86.6%
1	32	128	92%	89.1%	98.2%	86.6%
1	32	256	91.73%	88.6%	98.4%	85.9%
2	16	32	92.1%	89.2%	98.2%	86.69%
2	16	64	91.8%	88.8%	98%	86.1%
2	16	128	91.8%	88.8%	98.5%	86%
2	16	256	91.71%	88.6%	98.4%	85.8%
2	32	32	92.2%	89.4%	98%	87.1%
2	32	64	92.31%	89.5%	98.2%	87.1%
2	32	128	91.8%	88.7%	98.4%	86%
2	32	256	91.8%	88.8%	98.6%	85.99%

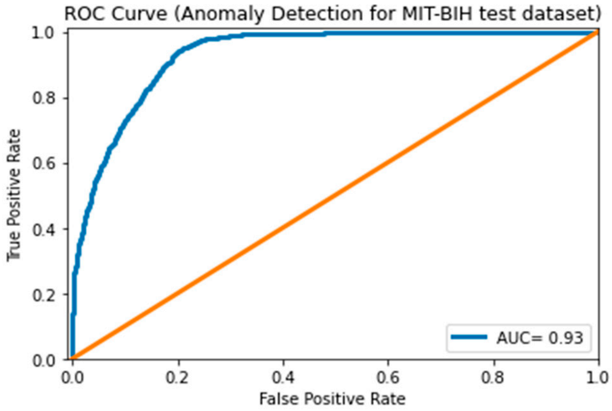


Figure 9. ROC curve of the MIT-BIH testing set.

Comparing our results with other existing deep learning anomaly detection models that used the MIT-BIH dataset is shown in Table 6. In [26], the authors used stacked LSTM and applied unsupervised training, since no anomaly classes are needed for training the model. The dataset splitting was 80% training and 20% testing. Therefore, we also compare our model with theirs because they followed the same training procedure and data splitting. Our model achieved an F1 of 92.3%, recall of 98.2% and precision of 87.1% while they reported an F1 of 81%, recall of 87% and precision of 82% which indicates that our model outperforms the unsupervised LSTM attempts. Furthermore, in [45] the authors used a novel hybrid architecture consisting of LSTM cells and multi-layer perceptrons (MLP), and the dataset splitting was 70% training and 30% testing. They achieved 87% F1 and 75% sensitivity in a supervised manner, which is also less than our result, while the higher accuracy may be attributed to the fact that the authors have used supervised learning. Finally, the authors in [42] used unsupervised variational AEs with AAMI dataset splitting, but they had the lowest F1 and accuracy with 76.55% and 87.77%, respectively. Our model, compared with state-of-the-art application anomaly detection methods using either supervised or unsupervised deep-learning models, outperforms them. However, this model's performance in anomaly detection still lacks a satisfactory F-score.

Table 6. Results of the MIT-BIH Arrhythmia test dataset.

Model	S/U	Dataset Splitting	F1	Accuracy	Recall (Sensitivity)	Precision
Stacked LSTM [26]	U	80% training, 20% testing	81%	-	87%	82%
(LSTM) with (MLP) [45]	S	70% training, 30% testing	87%	95%	75%	-
VAE [42]	U	AAMI Dataset splitting	76.55%	87.77%	-	-
This work	U	80% training, 20% testing	92.3%	89.5%	98.2%	87.1%

For evidence, symbols (S,U) denote the supervised and unsupervised learning, respectively.

5. Conclusions

We have introduced a robust and effective unsupervised transformer anomaly detection model in time series data. The suggested model was utilized to detect anomalies in human heartbeat time series signals, such as premature ventricular contraction (PVC), supraventricular premature (SP), and other ECG anomalies. A transformer encoder network and linear dense decoder networks comprise the model architecture. The ECG time series anomaly detection approach is based on a sequence prediction method that has two stages: model training, in which the model learns the normal data distribution, and anomaly detection, in which the ECG time series anomaly score is computed to identify abnormalities. We adapt the anomaly detection method into an unsupervised framework by using a pre-set threshold of mean plus two times the standard deviation of training loss.

We have shown how the transformer encoder can be leveraged for ECG time series anomaly detection. Our results for both the ECG5000 and MIT-BIH Arrhythmia datasets have shown that a transformer encoder is a viable substitute for this task. The transformer encoder was tested against several state-of- the art deep learning models. We have demonstrated that our model outperformed the compared models in accuracy, F1-score, recall, and precision. For the ECG5000 heartbeat dataset, we have achieved a 99% accuracy, a 99% F1 score, a 100% precision, a 98.1% recall, and a 99% AUC score, demonstrating an excellent ability to determine abnormalities in ECG heartbeat signals. Moreover, the model revealed respectable weighted F1 score of 92.3%, accuracy of 89.5%, AUC score of 93%, recall of 98.2%, and precision of 87.1% on the MIT-BIH Arrhythmia dataset. In addition, the proposed model is more robust than other deep learning models in recent ECG analyses using MIT-BIH Arrhythmia dataset. Though this work has focused on comparing the model with deep learning anomaly detection approaches, we aim in future work to expand our investigation on other anomaly detection techniques of time series data.

Author Contributions: Conceptualization, A.A. (Abrar Alamr) and A.A. (Abdelmonim Artoli); methodology, A.A. (Abrar Alamr); software, A.A. (Abrar Alamr); validation, A.A. (Abrar Alamr) and A.A. (Abdelmonim Artoli); formal analysis, A.A. (Abrar Alamr); investigation, A.A. (Abrar Alamr); writing—original draft preparation, A.A. (Abrar Alamr); writing—review and editing, A.A. (Abdelmonim Artoli); visualization, A.A. (Abrar Alamr); supervision, A.A. (Abdelmonim Artoli); project administration, A.A. (Abdelmonim Artoli) All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The ECG5000 Data is available at http://www.timeseriesclassification.com/description.php?Dataset=ECG5000 (accessed on 11 February 2023) and MIT-BIH Arrhythmia Data is available at https://physionet.org/content/mitdb/1.0.0/ (accessed on 11 February 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chatterjee, A.; Ahmed, B.S. IoT anomaly detection methods and applications: A survey. *Internet Things* **2022**, *19*, 100568. [CrossRef]
2. Li, H.; Boulanger, P. Structural Anomalies Detection from Electrocardiogram (ECG) with Spectrogram and Handcrafted Features. *Sensors* **2022**, *22*, 2467. [CrossRef]
3. Schmidl, S.; Wenig, P.; Papenbrock, T. Anomaly Detection in Time Series: A Comprehensive Evaluation. *Proc. VLDB Endow.* **2022**, *15*, 1779–1797. [CrossRef]
4. Mehrotra, K.G.; Mohan, C.K.; Huang, H. *Introduction. Anomaly Detection Principles and Algorithms*; TSC; Springer: Cham, Switzerland, 2017; pp. 3–19. [CrossRef]
5. Karim, F.; Majumdar, S.; Darabi, H.; Chen, S. LSTM Fully Convolutional Networks for Time Series Classification. *IEEE Access* **2017**, *6*, 1662–1669. [CrossRef]
6. Ariyaluran Habeeb, R.A. Clustering-based real-time anomaly detection—A breakthrough in big data technologies. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, 8. [CrossRef]
7. Thudumu, S.; Branch, P.; Jin, J.; Jack Singh, J. A comprehensive survey of anomaly detection techniques for high dimensional big data. *J. Big Data* **2020**, *7*, 42. [CrossRef]
8. Contreras, J.; Espinola, R.; Nogales, F.J.; Conejo, A.J. ARIMA models to predict next-day electricity prices. *IEEE Trans. Power Syst.* **2003**, *18*, 1014–1020. [CrossRef]
9. Gao, J.; Liang, F.; Fan, W.; Wang, C.; Sun, Y.; Hann, J. On community outliers and their efficient detection in information networks. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 25–28 July 2010; pp. 813–822.
10. Barnett, V.; Lewis, T. Outliers in Statistical Data. Wiley Series in Probability and Mathematical Statistics. In *Applied Probability and Statistics*, 2nd ed.; Wiley: Chichester, UK, 1984.
11. Audibert, J.; Michiardi, P.; Guyard, F.; Marti, S.; Zuluaga, M.A. USAD: UnSupervised Anomaly Detection on Multivariate Time Series. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Virtual, 6–10 July 2020; pp. 3395–3404. [CrossRef]
12. Kaur, H.; Singh, G.; Minhas, J. A review of machine learning based anomaly detection techniques. *arXiv* **2013**, arXiv:1307.7286. [CrossRef]
13. Laxhammar, R. Anomaly Detection. *Conform. Predict. Reliab. Mach. Learn. Theory Adapt. Appl.* **2014**, *14*, 71–97. [CrossRef]
14. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survy. *ACM Comput. Surv.* **2009**, *41*, 1–58. [CrossRef]
15. Ejay, N.; Oluwarotimi, W.S.; Mojisola, G.A.; Guanglin, L. Intelligence Combiner: A Combination of Deep Learning and Handcrafted Features for an Adolescent Psychosis Prediction using EEG Signals. In Proceedings of the 2022 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0&IoT), Trento, Italy, 7–9 June 2022. [CrossRef]
16. Chen, Z.; Yeo, C.K.; Lee, B.S.; Lau, C.T. Autoencoder-based network anomaly detection. In Proceedings of the Wireless Telecommunications Symposium, Phoenix, AZ, USA, 17–20 April 2018; pp. 1–5. [CrossRef]
17. Nanduri, A.; Sherry, L. Anomaly detection in aircraft data using Recurrent Neural Networks (RNN). In Proceedings of the ICNS 2016: Securing an Integrated CNS System to Meet Future Challenges, Herndon, VA, USA, 19–21 April 2016; pp. 1–8. [CrossRef]
18. Malhotra, P.; Ramakrishnan, A.; Anand, G.; Vig, L.; Agarwal, P.; Shroff, G. LSTM-based Encoder-Decoder for Multi-Sensor Anomaly Detection. *arXiv* **2016**, arXiv:1607.00148.
19. Lu, L.; Krause, B.; Murray, I.; Renals, S. Multiplicative LSTM for sequence modelling. *arXiv* **2017**, arXiv:1609.07959.
20. Vaswani, I.P.A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, L. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 5999–6009.
21. Chauhan, S.; Vig, L. Anomaly detection in ECG time signals via deep long short-term memory networks. In Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2015, Paris, France, 19–21 October 2015. [CrossRef]
22. Sugimoto, K.; Lee, S.; Okada, Y. Deep learning-based detection of periodic abnormal waves in ECG data. *Lect. Notes Eng. Comput. Sci.* **2018**, *2233*, 35–39.
23. Malhotra, P.; Vig, L.; Shroff, G.; Agarwal, P. Long Short Term Memory networks for anomaly detection in time series. In Proceedings of the 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2015, Bruges, Belgium, 22–23 April 2015; pp. 89–94.
24. Zhu, G.; Zhao, H.; Liu, H.; Sun, H. A Novel LSTM-GAN Algorithm for Time Series Anomaly Detection. In Proceedings of the 2019 Prognostics and System Health Management Conference, PHM-Qingdao 2019, Qingdao, China, 25–27 October 2019. [CrossRef]

25. Schlegl, T.; Seeböck, P.; Waldstein, S.M.; Schmidt-Erfurth, U.; Langs, G. *Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery*; Springer: Cham, Switzerland, 2017.
26. Thill, M.; Däubener, S.; Konen, W.; Bäck, T. Anomaly Detection in Electrocardiogram Readings with Stacked LSTM Networks. In Proceedings of the 19th Conference Information Technologies—Applications and Theory (ITAT 2019), Donovaly, Slovakia, 20–24 September 2019.
27. Xu, L. TGAN-AD: Transformer-Based GAN for Anomaly Detection of Time Series Data. *Appl. Sci.* **2022**, *12*, 8085. [CrossRef]
28. Chen, Z.; Chen, D.; Zhang, X.; Yuan, Z.; Cheng, X. Learning Graph Structures with Transformer for Multivariate Time Series Anomaly Detection in IoT. *IEEE Internet Things J.* **2021**, *9*, 9179–9189. [CrossRef]
29. Rui, H.; Jie, C.; Li, Z. A transformer-based deep neural network for arrhythmia detection using continuous ECG signals. *Comput. Biol. Med.* **2022**, *144*, 105325.
30. PhysioBank PhysioToolkit. PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation* **2000**, *101*, 215–220.
31. Moody, G.B.; Mark, R.G. The impact of the MIT-BIH arrhythmia database. *IEEE Eng. Med. Biol. Mag.* **2001**, *20*, 45–50. [CrossRef]
32. Chen, Y.; Hao, Y.; Rakthanmanon, T.; Zakaria, J.; Hu, B.; Keogh, E. A general framework for never-ending learning from time series streams. *Data Min. Knowl. Discov.* **2015**, *29*, 1622–1664. [CrossRef]
33. Luz, E.J.D.S.; Schwartz, W.R.; Cámara-Chávez, G.; Menotti, D. ECG-based heartbeat classification for arrhythmia detection: A survey. *Comput. Methods Programs Biomed.* **2016**, *127*, 144–164. [CrossRef]
34. Lee, M.; Song, T.-G.; Lee, J.-H. Heartbeat classification using local transform pattern feature and hybrid neural fuzzy-logic system based on self-organizing map. *Biomed. Signal Process. Control.* **2020**, *57*, 101690. [CrossRef]
35. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 630–645.
36. Lei Ba, J.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
37. Wang, Y.; Lu, W. Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *324*, 012049. [CrossRef]
38. Van Rossum, G.; Drake, F.L. *Python 3 Reference Manual:(Python Documentation Manual Part 2)*; CreateSpace: Scotts Valley, CA, USA, 2009.
39. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv* **2016**, arXiv:1603.04467.
40. Chollet, F. Keras. Available online: https://github.com/fchollet/keras (accessed on 12 January 2015).
41. Pereira, J.; Silveira, M. Learning Representations from Healthcare Time Series Data for Unsupervised Anomaly Detection. In Proceedings of the 2019 IEEE International Conference on Big Data and Smart Computing, BigComp 2019, Kyoto, Japan, 27 February–2 March 2019. [CrossRef]
42. Matias, P.; Folgado, D.; Gamboa, H.; Carreiro, A.V. Robust anomaly detection in time series through variational AutoEncoders and a local similarity score. In Proceedings of the BIOSIGNALS 2021—14th International Conference on Bio-Inspired Systems and Signal Processing; Part of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies, BIOSTEC 2021, Virtual, 11–13 February 2021; Volume 4, pp. 91–102. [CrossRef]
43. Oluwasanmi, A.; Aftab, M.U.; Baagyere, E.; Qin, Z.; Ahmad, M.; Mazzara, M. Attention Autoencoder for Generative Latent Representational Learning in Anomaly Detection. *Sensors* **2021**, *22*, 123. [CrossRef] [PubMed]
44. Khandual, A.; Dutta, K.; Lenka, R.; Nayak, S.R.; Bhoi, A.K. MED-NET: A novel approach to ECG anomaly detection using LSTM auto-encoders. *Int. J. Comput. Appl. Technol.* **2021**, *65*, 343. [CrossRef]
45. Sivapalan, G.; Nundy, K.K.; Dev, S.; Cardiff, B.; John, D. ANNet: A Lightweight Neural Network for ECG Anomaly Detection in IoT Edge Sensors. *IEEE Trans. Biomed. Circuits Syst.* **2022**, *16*, 24–35. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.