

## Article

# TimeTector: A Twin-Branch Approach for Unsupervised Anomaly Detection in Livestock Sensor Noisy Data (TT-TBAD)

Junaid Khan Kakar <sup>1,2</sup>, Shahid Hussain <sup>3</sup> , Sang Cheol Kim <sup>2</sup> and Hyongsuk Kim <sup>2,4,\*</sup> <sup>1</sup> Department of Electronics and Information Engineering, Jeonbuk National University, Jeonju 54896, Republic of Korea; junaidk@jbnu.ac.kr<sup>2</sup> Core Research Institute of Intelligent Robots, Jeonbuk National University, Jeonju 54896, Republic of Korea; skkim7777@gmail.com<sup>3</sup> Innovation Value Institute (IVI), School of Business, National University of Ireland Maynooth (NUIM), W23 F2H6 Maynooth, Ireland; shahid.hussain@mu.ie<sup>4</sup> Department of Electronics Engineering, Jeonbuk National University, Jeonju 54896, Republic of Korea

\* Correspondence: hskim@jbnu.ac.kr

**Abstract:** Unsupervised anomaly detection in multivariate time series sensor data is a complex task with diverse applications in different domains such as livestock farming and agriculture (LF&A), the Internet of Things (IoT), and human activity recognition (HAR). Advanced machine learning techniques are necessary to detect multi-sensor time series data anomalies. The primary focus of this research is to develop state-of-the-art machine learning methods for detecting anomalies in multi-sensor data. Time series sensors frequently produce multi-sensor data with anomalies, which makes it difficult to establish standard patterns that can capture spatial and temporal correlations. Our innovative approach enables the accurate identification of normal, abnormal, and noisy patterns, thus minimizing the risk of misinterpreting models when dealing with mixed noisy data during training. This can potentially result in the model deriving incorrect conclusions. To address these challenges, we propose a novel approach called “TimeTector-Twin-Branch Shared LSTM Autoencoder” which incorporates several Multi-Head Attention mechanisms. Additionally, our system now incorporates the Twin-Branch method which facilitates the simultaneous execution of multiple tasks, such as data reconstruction and prediction error, allowing for efficient multi-task learning. We also compare our proposed model to several benchmark anomaly detection models using our dataset, and the results show less error (MSE, MAE, and RMSE) in reconstruction and higher accuracy scores (precision, recall, and F1) against the baseline models, demonstrating that our approach outperforms these existing models.

**Keywords:** anomaly detection; unsupervised learning; time series sensor; LSTM autoencoder; multi-head attention mechanism; DAE; reconstruction and prediction branch; livestock



**Citation:** Kakar, J.K.; Hussain, S.; Kim, S.C.; Kim, H. TimeTector: A Twin-Branch Approach for Unsupervised Anomaly Detection in Livestock Sensor Noisy Data (TT-TBAD). *Sensors* **2024**, *24*, 2453. <https://doi.org/10.3390/s24082453>

Academic Editor: Richard J. Pavinelli

Received: 23 February 2024

Revised: 18 March 2024

Accepted: 20 March 2024

Published: 11 April 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Anomaly detection has been a central focus in machine learning research for many years. An anomaly in time series data is a deviation from the expected pattern or trend within a given time frame, showing up as outliers, fluctuations, or changes in the data, indicating unusual behavior [1]. Livestock anomalies refer to unexpected changes in trends or patterns, such as irregularities in population dynamics, production metrics, or environmental factors like temperature, humidity, and gases. It has widespread applications across various domains, including livestock farming [2], cyber-intrusion detection, healthcare, sensor networks, and image anomaly detection. The expansion of livestock farms and the number of animals necessitate advanced management techniques for optimizing productivity with limited manpower. Failure to address the irregularities in livestock data can lead to various consequences, including untreated illnesses, reduced productivity, and heightened mortality rates, posing significant risks to animal welfare and farm

profitability [3]. Detection and management of these anomalies are crucial for ensuring optimal farm management and animal well-being. Precision farming emerges as a solution, propelling the transition toward automated livestock smart farms [4]. Central to this is anomaly detection, which is crucial in sectors like finance and healthcare but pivotal in livestock management. Farmers can gain valuable insights by detecting outliers in data points that deviate significantly from typical patterns, which are frequently caused by discrepancies and errors in the sensors. Organizations are exploring methods to enhance pig productivity with minimal human involvement. Livestock health, well-being, and productivity can be significantly improved by paying close attention to them and promptly detecting anomalies. This can help in identifying potential health issues or environmental inconsistencies in livestock settings, allowing for swift interventions to maintain animal welfare and farm productivity. Studies have demonstrated that focusing on the livestock sector can result in positive outcomes, as evidenced by a wealth of research [4,5]. Given the increasing scale of livestock rearing, adopting intelligent agricultural solutions is essential to ensuring optimal farm and livestock housing conditions, especially in pig farming, where it is crucial. The seamless integration of different equipment within and around barns is essential for creating and maintaining optimal growth conditions for livestock. Gas sensors in pig barns ensure the safety and health of animals and workers. Conventional outlier detection techniques lack efficacy when dealing with large, multi-dimensional datasets. Autoencoders, representing a type of deep learning technique, have excelled in detecting outliers due to their ability to identify complex patterns and relationships in the data. Autoencoders have revolutionized anomaly detection and prediction [6]. Advanced analytical techniques have the capability to identify complex, nonlinear relationships within data. These techniques allow for the precise identification of anomalies in offering predictive insights essential for improving the efficiency and productivity of livestock management.

Having accurate data is essential for making informed decisions. Several factors can cause variations in data quality, including measurement errors, inaccurate data entries, sensor failures, or deliberate sabotage. These challenges can be complex to manage using sophisticated methodologies. Analyzing spatial interconnections within multi-sensor datasets is crucial for making informed decisions in the current era. The complexity lies in differentiating between noise, typical, and atypical data. Therefore, there is a need for a dependable mechanism that enables precise identification and interpretation of such data. Several approaches for analyzing spatial dependencies in multiple sensors have been studied [7,8]. To deal with such challenges, researchers have developed various methods, e.g., data-driven anomaly detection [9] and maximum mean discrepancy (MMD) [10]. Deep learning approaches, especially those involving autoencoders, have surfaced as potent solutions for outlier detection, although some may fall short in performance. Their performance is often tied to their adeptness in unraveling intricate nonlinear relationships within data [11,12]. Through learning intricate data patterns, these techniques facilitate enhanced anomaly identification, fostering a more accurate and insightful analysis.

Pig house sensors collect real-time management data, ensuring optimal growth and animal welfare. Equipment inside and outside the house maintains a suitable environment and boosts productivity. Gas sensors monitor hazardous gases like carbon dioxide CO<sub>2</sub> and ammonia NH<sub>3</sub> which can accumulate due to animal respiration and waste. Integrating gas sensors with ventilation systems helps control the indoor climate and maintain a healthy environment for animals and workers. High carbon dioxide CO<sub>2</sub> levels can pose a suffocation risk [13,14]. Using mechanisms that capture spatial-temporal correlations and dependencies among diverse data streams has proven highly effective. Traditional models like RNNs, VAEs, or GANs may need to become more capable of addressing this challenge with the same level of efficiency [15]. Using our proprietary dataset from the pig farm, we show that pig diseases are the most critical factor affecting productivity. Preventing infections is crucial for maintaining pig productivity. Proper maintenance of

the pig house and equipment can create a healthier environment and decrease the chances of infection [16,17].

Autoencoder-based models are effective in smart farming, especially in anomaly detection. A study found that they can detect anomalies in agricultural vehicles by identifying obstructive objects in the field. The study also introduced a semi-supervised autoencoder trained with a max-margin-inspired loss function that outperformed basic and denoising autoencoders in the generation of anomaly maps using relative-perceptual-L1 loss [18]. According to the research, autoencoders are identified as a versatile tool for anomaly detection. Autoencoders use an encoder and a decoder to minimize reconstruction errors for normal data. Anomalies result in higher reconstruction errors and are identifiable as outliers [19,20].

Several techniques have been proposed to overcome the limitations of outlier detection: LSTM, LSTM-AE, ConvLSTM, and GRU. These standard techniques have limitations in identifying the capacity to detect a wider range of subtle and complex anomalies, especially in intricate, dynamic industrial environments [21,22]. One such technique is represented by ensemble methods which combine the outputs of multiple models to improve accuracy and robustness. Like in any technology, there is always a chance of malfunction, probably leading to undetected hazardous conditions [23]. Advanced analytics and machine learning improve the speed and accuracy of real-time data processing, enabling precise anomaly detection and interpretation of complex data relationships in livestock farming. However, the growing use of IoT devices in agriculture raises security and privacy concerns that must be addressed. Current anomaly detection methods in agriculture, especially in livestock farming and industrial settings, must handle real-time, multi-dimensional data. Traditional autoencoder architectures struggle to adapt to the complexities of dynamic and noisy farm environments. Unpredictable variables often affect data in this field, resulting in a higher rate of false positives or negatives.

In our research, we present a new model called the “TimeTector (TT-TBAD)”, which offers a revolutionary approach to detecting anomalies in agricultural environments. Our innovative methodology is specifically designed to handle the complexities and nuances of multi-sensor and real-time series data that are common in farming settings, thereby providing a robust and advanced autoencoder ensemble approach that addresses the unique challenges of monitoring pig farm environments. It is capable of detecting outliers with remarkable accuracy and can also make predictive inferences, which is a feature that is often absent in traditional models. As a result, “TimeTector (TT-TBAD)” represents a significant advancement in the field, providing an effective and precise solution for detecting anomalies in critical agricultural applications. Our innovative TT-TBAD methodology is designed to detect anomalies in livestock sensor data using advanced techniques. It utilizes a Twin-Branch Shared LSTM Autoencoder framework, incorporating a Denoising Autoencoder (DAE) and several Multi-Head Attention mechanisms. This approach can be used to effectively handle the noisy data commonly found in agricultural settings. The model captures complex spatial-temporal correlations and hierarchical relationships while minimizing reconstruction and prediction errors. Real-time performance insights are provided via a dynamic feedback loop. Optimized for multi-sensor time series data, TT-TBAD enhances data quality by reducing noise and providing accurate forecasts, enabling proactive farm management and improved productivity and animal welfare. In summary, this study makes three key contributions:

- This model integrates advanced mechanisms, including Multi-Head Attention, Hierarchical Multi-Head Attention, Cross Multi-Head Attention, and Feature-Enhanced Multi-Head Attention. These features enable the analysis of hierarchical data, identification of complex relationships, and interpretation of auxiliary features. Moreover, the model’s dual-branch LSTM structure enhances spatial-temporal data analysis, resulting in improved accuracy for anomaly detection in livestock farming, a capability not adequately addressed by traditional models.

- The integration of a Denoising Autoencoder (DAE) with Shared LSTM encoder branches manages noisy agricultural sensor data during training, mitigating overfitting and enhancing adaptability. The development of a Twin-Branch structure for multi-task learning enables simultaneous data reconstruction and future prediction, distinguishing between normal operations, noisy data, and anomalies. This dual-task learning approach generates precise, actionable insights in real time, a capability not typically found in traditional models.
- The developed model uses a feedback loop based on reconstruction and prediction errors to continuously monitor and adjust its performance, through a dynamic rather than static threshold-based mechanism. This adaptive mechanism significantly improves the detection accuracy and the model's adaptability to evolving farm conditions by boosting the accuracy and adaptability in anomaly detection, aiding decision-making. Finally, evaluation results showed a lower construction on error (MSE, MAE, and RMSE) and high accuracy (precision, recall, and F1) when evaluated against the benchmark and baseline anomaly detection models.

The rest of the paper is organized as follows: Section 2 provides related work, exploring state-of-the-art techniques for anomaly detection in sensor data. Sections 3 and 4 describe our proposed methodology and detailed framework. The dataset explanation and analysis are described in Section 5. Performance evaluation, results, and analysis of the experiment follow in Section 6. Finally, Section 8 concludes the paper with directions for possible future work.

## 2. Related Work

Several research studies have been performed on existing state-of-the-art techniques for detecting anomalies in indoor and outdoor pig farm environment datasets and other similar fields [24]. This detection process has been studied for decades in natural language processing (NLP) and images and videos [25]. Various methods are available for anomaly detection, including statistical, supervised machine learning-based, and unsupervised deep learning-based approaches. These methods have been explored in detail through literature reviews in similar studies; as such, our focus is on unsupervised learning due to the lack of labels. Outlier detection in real-time series data has unique challenges that require efficient processing and handling of data drift or concept drift [26]. Machine learning and deep learning advancements in analyzing time series data from sensor networks show that spatial information from sensor locations can be leveraged to improve predictions [27].

The use of techniques such as sliding windows, feature selection, and online learning algorithms for handling real-time series data in outlier detection tasks has been explored in previous research, where challenges specific to real-time series data in outlier detection have been acknowledged [28]. These challenges include handling the data drift, where the statistical properties of the data change over time, and the concept drift, where there is the evolution of the underlying concepts being monitored. To address these challenges, techniques like sliding windows, which limit the data window for analysis, and feature selection, which focuses on the most relevant attributes, have been explored [29]. Additionally, online learning and multilayer neurocontrol of high-order uncertain nonlinear systems with active disturbance rejection algorithms, which adapt to changing data distributions and solve real-world problems, have been investigated to improve the efficiency of outlier detection in real-time scenarios [30,31].

Additionally, autoencoder-based anomaly detection has proven to be highly effective in IoT-based smart farming systems, exhibiting exceptional performance in accurately detecting anomalies in key variables such as soil moisture, air temperature, and air humidity. The integration of this technology has shown superior performance, making it a valuable tool for precision agriculture. This study proposes a new approach for anomaly detection using autoencoder-based reconstructed inputs. The proposed method outperforms existing techniques and facilitates the creation of supervised datasets for further research. The

hourly forecasting models for soil moisture and air temperature demonstrate low numbers of estimation errors, indicating the effectiveness of this method [32].

In order to improve the accuracy of intrusion detection on the Internet of Things (IoT), intrusion labels are integrated into the decoding layers of the model. However, this approach may not effectively handle the data and concept drift challenges frequently encountered in real-time anomaly detection. Additionally, multi-sensor technology IoT devices can be used in both cloud and edge computing infrastructure to provide remote control of watering and fertilization, real-time monitoring of farm conditions, and solutions for more sustainable practices. However, these improvements in efficiency and ease of use come with increased risks to security and privacy. Combining vulnerable IoT devices with the critical infrastructure of the agriculture domain broadens the attack surface for adversaries, making it easier for them to launch cyber-attacks [33]. An end-to-end deep learning framework for extracting fault-sensitive features from multi-sensor time series data showcases data-driven methods to handle spatiotemporal cases [34].

Anomaly detection involves a comprehensive survey of deep learning techniques for anomaly detection, covering various methods and their applications across different domains. However, the survey might not delve deep into real-time series data anomaly detection [35]. A survey on anomaly detection that focused on time series using deep learning and covered a range of applications, including healthcare and manufacturing, provided specific insights into livestock health monitoring or multi-task learning [36]. Knowledge distillation for compressing an unsupervised anomaly detection model and addressing the challenge of deploying models on resource-constrained devices are examples of the challenges associated with real-time data processing and multi-task learning in anomaly detection [37]. One learning-based study introduced a simple yet effective method for anomaly detection by learning small perturbations to perturb normal data and subsequently classifying the normal and perturbed data into two different classes using deep neural networks. The focus on perturbation learning might not align with the Multi-Head Attention and memory-driven multi-task learning approaches in our work [38].

### 3. Preliminaries

#### 3.1. LSTM Network

Hochreiter and Schmid Huber introduced LSTM in 1997. It is an advanced RNN that addresses the vanishing gradient problem and models long-term dependencies; Figure 1. LSTM is popular in commercial applications for its memory cells controlled by gate units [39], including input, forget, and output gates, to selectively store information over time as given in Equation (1). A detailed explanation is given in Section 4.

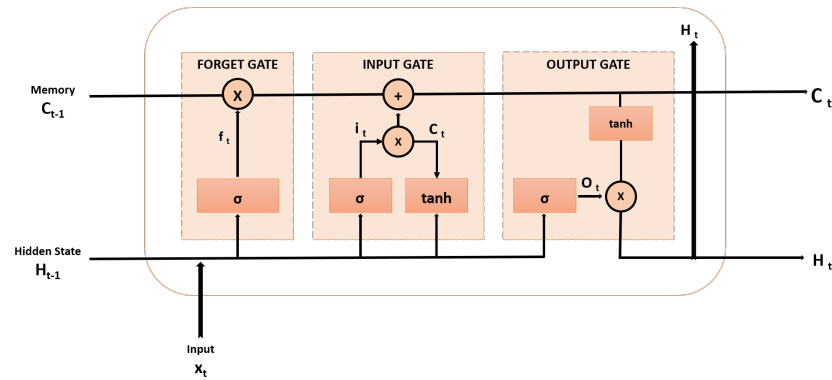
$$LSTM_{Structure} = \begin{cases} f_t &= \sigma(W_f \times [h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i \times [h_{t-1}, x_t] + b_i) \\ \tilde{c}_t &= \tanh(W_c \times [h_{t-1}, x_t] + b_c) \\ c_t &= f_t \times c_{t-1} + i_t \times \tilde{c}_t \\ o_t &= \sigma(W_o \times [h_{t-1}, x_t] + b_o) \\ h_t &= o_t \times \tanh(c_t) \end{cases} \quad (1)$$

#### 3.2. Encoder–Decoder

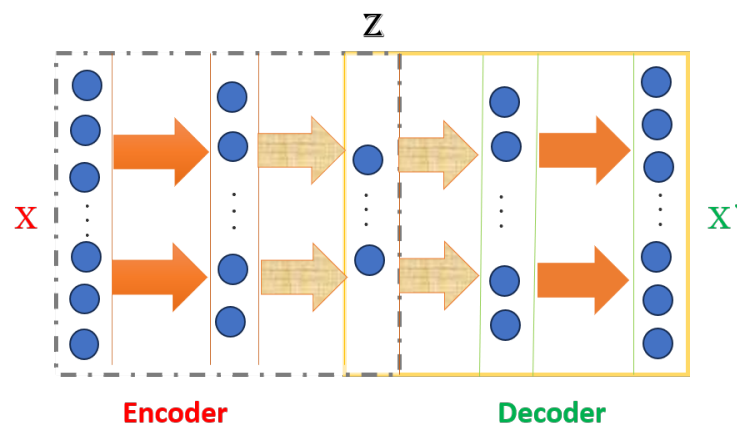
Autoencoders use an encoder to compress input data into a “latent space” and a decoder to reconstruct original data as shown in Figure 2. Minimizing the reconstruction error during training allows for effective feature extraction and data reconstruction, presented by Equation (2) [21,33,34].

$$Output = \|x - g(f(x))\|^2 \quad (2)$$





**Figure 1.** Long short-term memory (LSTM) cell with an input gate, an output gate, and a forget gate [39].



**Figure 2.** Image illustrates an encoder–decoder neural network model. The encoder compresses the input ‘X’ into a condensed representation ‘Z’, which the decoder then uses to reconstruct the output ‘X prime.’ Arrows indicate process flow and blue dots represent distinct layers [21].

### 3.3. Multi-Head Attention Mechanism

Multi-Head Attention is a powerful mechanism that enables the model to concentrate on different parts of the input data at the same time. It is like having several sets of “eyes” focusing on various features or aspects of the data, allowing for the model to capture diverse relationships simultaneously. Recently, the attention mechanism has helped solve many research problems [40]. This mechanism is a core part of the transformer architecture introduced by Vaswani et al., 2017 [41]. Multi-Head Attention is essential because, in simple self-attention, the model learns only one set of attention weights, which can be limiting. For example, a sentence may have a grammatical relationship with one word and a semantic relationship with another. With multiple heads, the model can simultaneously capture these various relationships, leading to better results in natural language processing tasks. In our case, the Multi-Head Attention mechanism is a key component, allowing for the extraction of complex features from sequence data. By utilizing multiple heads, the mechanism can focus on a range of spatial–temporal dependencies within the data, effectively capturing both long- and short-range dependencies. Using our approach, even the slightest patterns in data can be detected, significantly increasing system performance and accuracy.

## 4. The Proposed TT-TBAD Method

Time series anomaly detection is the process of analyzing sets of observations in a multivariable time series  $X = \{x_1, x_2, \dots, x_N\}$ , with  $x_N \in \mathbb{R}^O$  to identify any anomalies. The goal is to develop a scoring function that assigns a higher score to anomalous observa-

tions than normal ones ( $x_j$ ). The input format includes  $N$  observations, each of  $O$  variables represented by real numbers. The scoring function  $\Phi$  assigns a higher score to anomalous observations, which helps detect any unusual behavior  $x_i$ . We have  $\Phi(x_i) > \Phi(x_j)$ .

#### 4.1. Denoising Autoencoder (DAE)

Denoising autoencoders reconstruct the original input from a corrupted version to teach the hidden layer more resilient features. We integrate it into our proposed TT-TBAD to help the model learn the underlying structure of the data and ignore the noise in the training data as shown in Figure 3. The corrupted version is created by adding noise and passed through the encoder to acquire latent representation  $h$ , as presented by Equations (3) and (4). Latent representation  $h$  is then passed through the decoder to obtain the reconstructed version  $x'$  and loss  $L$  is computed by taking the difference between the original input  $x$  and the reconstructed version  $x'$ , as presented by Equations (5) and (6).

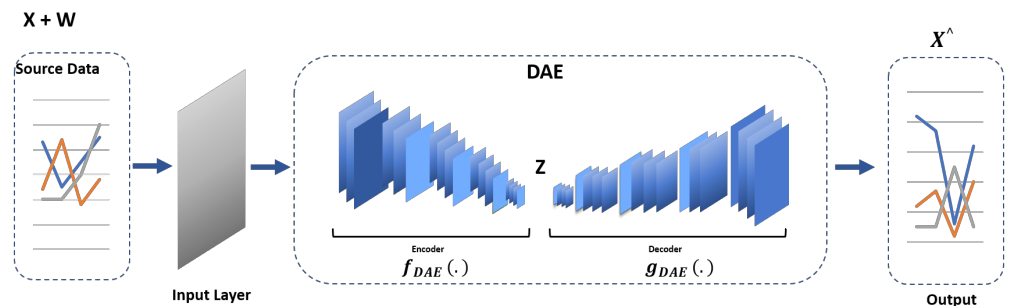
$$\tilde{x} = x + \text{noise} \quad (3)$$

$$h = f(\tilde{x}) \quad (4)$$

$$x' = g(f(\tilde{x})) \quad (5)$$

$$L(x, x') = \frac{1}{n} \sum_{i=1}^n (x_i - x'_i)^2 \quad (6)$$

where  $x$  is the original data,  $\tilde{x}$  is the data after the noise is added,  $x'$  is the reconstructed data, and  $n$  is the dimensionality of the input.



**Figure 3.** The denoising autoencoder architecture shows the addition of noise to the data passing through the encoder–decoder and subsequently through the shared LSTM.

#### 4.2. Shared LSTM Encoder with Attention

We designed a shared encoder with LSTM and attention mechanisms. This encoder can process input sequences and generate a context-aware representation that is passed to the model's numerous Multi-Head Attention mechanisms. For input  $X$ , the LSTM updates its cell state  $C$  and hidden state  $h$  at each timestep  $t$ . After processing the sequence, we have a series of hidden states  $ht$ . In our case, we are passing these hidden states on to the next phase as presented in Equation (7).

$$H = [h_1, h_2, \dots, h_T] \quad (7)$$

#### 4.3. Attention Mechanism Phase

Attention mechanisms empower models to dynamically allocate significance to different portions of the input sequence, where each sequence is then transformed into  $Q$ ,  $K$ , and  $V$  using a “Feature Linear Project”. With the help of these four diverse attention mechanisms, i.e., “Multi-Head Attention”, “Hierarchical Multi-Head Attention”, “Cross-Multi-Head Attention”, and “Feature-Enhanced Multi-Head Attention”, the job is to refine the LSTM–derived hidden states as shown in Equation (8), each offering a unique perspec-

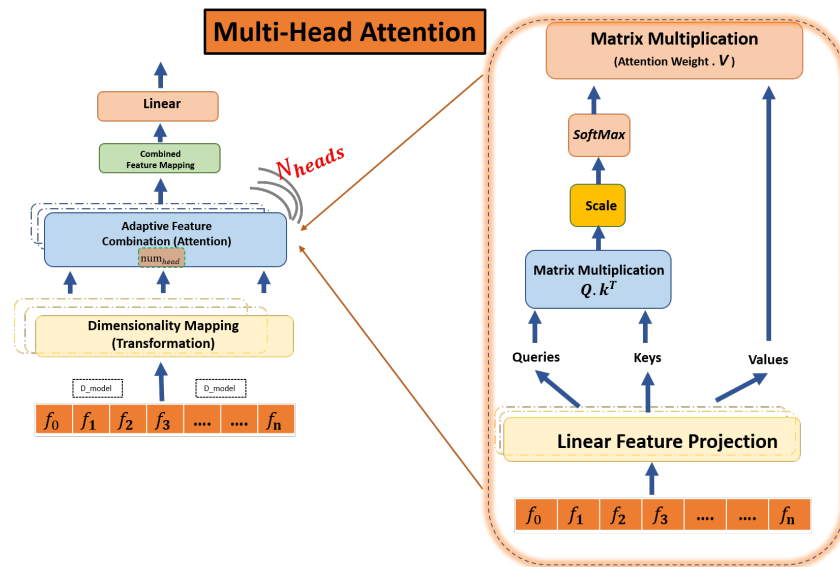
tive on data attention [42]. The detailed representation of model architecture is shown in the figure in Section 4.8.

$$Q = H \times W_Q, K = H \times W_K, V = H \times W_V, \quad (8)$$

where  $W_Q$ ,  $W_K$ , and  $W_V$  are learned weight matrices for queries, keys, and values, respectively. The learned weight matrices by each of these layers have dimensions  $d_{model} \times d_{model}$ , whereas  $d_{model}$  is the dimension of each attention. After the transformation,  $q$ ,  $k$ , and  $v$  are split into multiple attention heads, as given in Equation (9). In our case, with  $d_{model} = 64$  and  $num_{heads} = 8$ , each head processes a different subspace of the model dimensions, specifically handling  $d_{model} / num_{heads}$  dimensions, which is  $64 / 8 = 8$ .

#### 4.4. Multi-Head Attention

As previously stated, the transformer architecture employs a mechanism that processes a group of input sequences by projecting them into various subspaces and performing independent attention in each subspace. The Multi-Head Attention algorithm takes in the input sequences in the form of matrices  $Q$ (queries),  $K$ (keys), and  $V$ (values) and operates by conducting the following steps as shown in Figure 4 (left).



**Figure 4.** Structure of the Multi-Head Attention module: **(left)** the multi-head adaptive feature combination attention module with  $N$  parallel heads; **(right)** the basic scaled-dot product attention in each head separately.

Weight matrices multiply input sequences to generate new sequences for each attention head. The attention weights are obtained using a scaled-dot product and passed through *Softmax*. The final output is the concatenation of all attention head outputs, as presented by Equation (9).

$$\text{MultiHead}(Q, K, V) = \text{Attention}(QW_Q, KW_K, VW_V) \quad (9)$$

#### 4.5. Hierarchical Multi-Head Attention

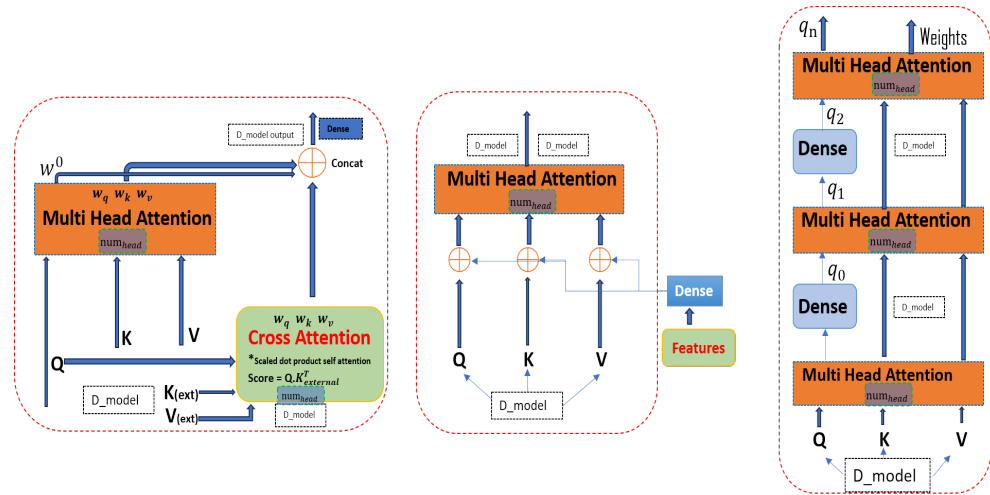
Different mechanisms are used in the literature to emphasize the special characteristics of the input data. Hierarchical Multi-Head Attention also exhibits the same characteristics. The model employs multiple layers of attention in a hierarchical structure. The output of one layer serves as the query for the next layer, allowing for the model to capture more complex relationships and dependencies between different parts of the sequence, as shown in Figure 5 (right) for hierarchical attention layer  $L$ , which consists of initial and subsequent layers.



For the first layer, the original  $Q$ ,  $K$ , and  $V$  matrices are used. In subsequent layers, the previous layer  $output_{l-1}$  is used as the query for the current layer, while  $K$  and  $V$  remain the same as shown in Equation (10).

$$Q_l = Output_{l-1}, K_l = K, V_l = V \quad (10)$$

where output  $L_1$  is the output of the hierarchical layer's  $L_1$ th.



**Figure 5.** Structure of the Cross-Multi-Head Attention module (**left**), Feature-Enhanced Multi-Head Attention module (**center**), and Hierarchical Multi-Head Attention module (**right**).

#### 4.6. Cross-Multi-Head Attention

The attention mechanism combines standard self-attention with cross-attention, which enhances the queries, keys, and values by projecting additional features. This provides context awareness to the attention mechanism, enabling the model to more effectively learn the dependencies between the different parts of the sequence given in Equation (11). The original and external keys and values are split into multiple heads, enabling parallel processing and capturing different types of relationships in the data. Moreover, the representation of cross-multi-head attention is visualized in Figure 5 (left). An external source from the LSTM layer provides  $V_{external}$  and  $K_{external}$  shown in the above equation. The final output may consist of a combination of both standard and cross-attention outputs in Equations (12) and (13).

$$\text{CrossAttention}(Q, K_{external}, V_{external}) \quad (11)$$

$$\text{Scores} = Q \cdot K_{external}^T \quad (12)$$

$$\text{CrossAttention}(Q, K_{external}, V_{external}) = \text{softmax}(Q \cdot K_{external}^T) \cdot V_{external} \quad (13)$$

The output is a matrix where each element is a weighted sum of the elements in the external  $V_{external}$ , with the weights determined by the compatibility of the queries with the keys.

#### 4.7. Feature-Enhanced Multi-Head Attention

This mechanism is designed to improve the functionality of queries, keys, and values by adding additional features. These additional features can be in the form of metadata or other relevant data that can be used to provide more context to the data being queried. Doing so makes it easier to retrieve the relevant data and perform more complex operations. Upon instantiation, it creates an additional feature-dense layer designed to project additional features  $F'$  into the same dimension as the queries, keys, and values. Feature matrix  $F$  is projected via a weight matrix  $W^F$  to obtain  $F'$ , which is then added to  $Q$ ,  $K$ , and

$V$  to compute the enhanced attention, as shown in Figure 5 (center). We consider feature matrix  $F$  in Equation (14) and the weight matrix for feature projection in Equation (15).

$$F' = FW^F \quad (14)$$

$$Q' = Q + F', \quad K' = K + F', \quad V' = V + F' \quad (15)$$

where  $W^F$  is a weight matrix for feature projection and the  $Q$ ,  $K$ , and  $V$  are the keys that are utilized to compute the attention.

By combining the output of different mechanisms, a comprehensive and nuanced representation of the input data is generated, allowing for more effective analysis and processing in further model layers.

#### 4.8. Attention Score Calculation (ASC)

To compute attention scores, each query is compared with all keys. The attention score for each subspace is independently executed using the adaptive feature combination attention approach between query  $q_t$  and key  $k_t$ , which is calculated for each head and shows the result in Figure 4 (right), determined through specific Equation (16).

This computation results in a matrix of attention scores, denoted as  $s_{t,j}$ , which has a size of  $T \times T$ , whereas  $T$  denotes a transpose operation. The value of  $s_{t,j}$  represents the score of attending from position  $t$  to position  $j$ .  $q_t$  and  $k_t$  are query and key vectors at positions  $t$  and  $j$  (17) [43]. A higher score indicates stronger relevance, causing the model to prioritize the corresponding value. Before using the softmax function, the attention score  $s_{t,j}$  is substituted by the root of each head ( $d_k$ ) to stabilize the gradients, and finally, after passing the result through a *Softmax* function, the weights are as shown in Equation (18) [43]:

$$\text{ASC } s_{t,j} = q_t^T k_j \quad (16)$$

$$\text{Softmax}\left(\frac{s_{t,j}}{\sqrt{d_k}}\right) \quad (17)$$

$$\text{Attention}(Q_i, K_i, V_i) = \text{Softmax}\left(\frac{Q_i W_i^Q}{\sqrt{D_k}}\right) V_i \quad (18)$$

where  $D_k = d_{\text{model}} / \text{num}_{\text{heads}}$  is the depth of each head. When computing attentive representation and concatenation for each timestep  $t$ , the weighted sum of all values is presented in Equation (19). In this way, we obtain a sequence of attentive representations  $R = [r_1, r_2, r_3, r_4, \dots, r_n]$ , as presented in Equation (20). The above process is repeated  $N$  times (for  $N_{\text{heads}}$ ) with different weight matrices  $W_Q^n$ ,  $W_K^n$ ,  $W_V^n$  each time and produces attentive representation sequence  $R_n$  differently as a final output, as given in Equation (21). Here, we acquire the output of all attention mechanisms after concatenation for the concatenated output, as presented in Equation (22) which can be seen in the detailed model architecture; Figure 6.

$$r_t = \sum_{j=1}^T a_{t,j} v_j \quad (19)$$

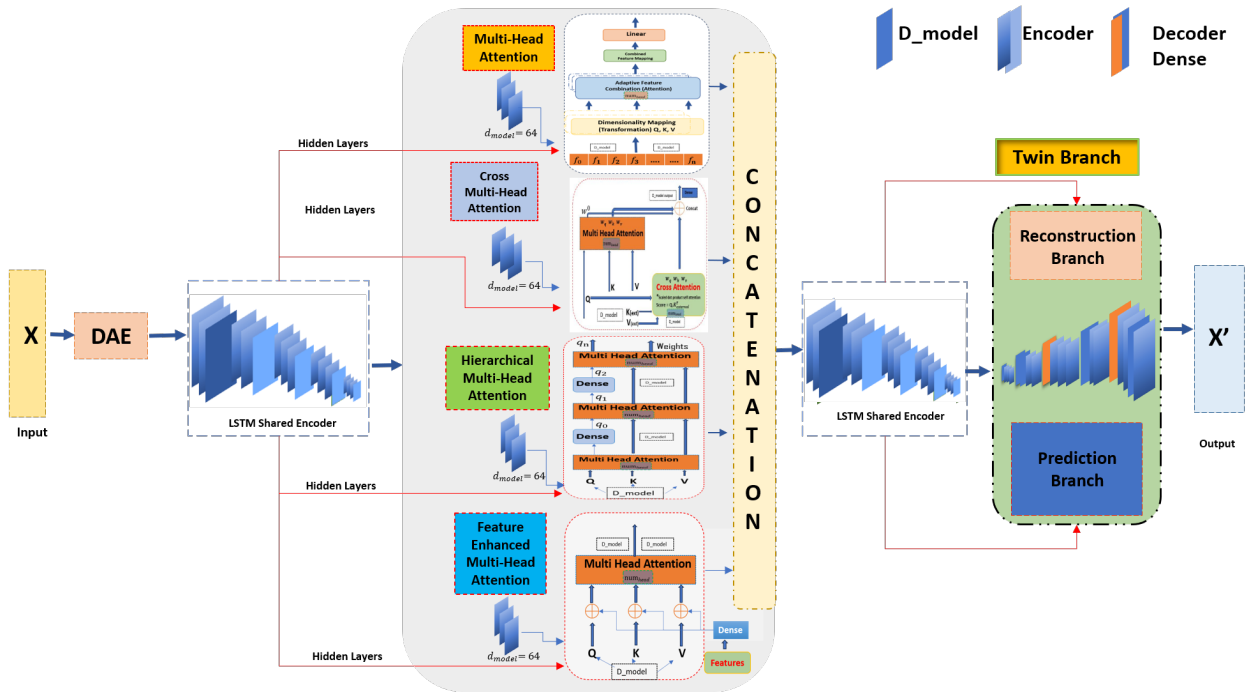
$$\text{Concat} = [r_1, r_2, r_3, r_4, \dots, r_n] \quad (20)$$

$$O_t = [R_t^1, R_t^2, R_t^3, \dots, R_t^N] \times W_O \quad (21)$$

$$\text{Concat Output} = \text{Concat}(\text{MHA}, \text{H-MHA}, \text{C-MHA}, \text{FE-MHA}) \quad (22)$$

where in Equation (21)  $W_O$  is another learned weight matrix, and Equation (22) denotes the combination of outputs from multiple attention mechanisms into a single tensor. In this case, the second shared encoder evaluates the merged output, capturing intricate temporal

patterns, assimilating information from various attention outputs, and transmitting it to the twin branch.



**Figure 6.** Architecture of the proposed TT-TBAD highlighting two shared LSTM encoders and an attention branch portion leading to the Twin-Branch section.

#### 4.9. Reconstruction and Prediction Based on Encoded Representation with Attention

The input sequence  $X$  is employed in condensed representation  $E$  by encoding function  $f_{enc}$ , which likely employs attention mechanisms to capture significant features of  $X$ .

Two mappings denote the twin branch (reconstruction branch, prediction branch); as shown in the architecture of the proposed model Figure 6, the first one reconstructs  $X$

from  $E$  and is represented as  $X = f_{rec}(E)$ . The second mapping predicts future  $X$  values from  $E$ , denoted as  $P = f_{pred}(E)$ .

##### 4.9.1. Reconstruction Branch

To measure the error in the reconstruction process, we use reconstruction error  $L_{rec}$ , which calculates the discrepancy between  $X$  and its reconstructed version  $\hat{X}$ , across all timesteps using the Euclidean  $L_2$  norm. This essentially computes the geometric distance between the corresponding elements of  $X$  and  $\hat{X}$  at each timestep  $t$  and adds up these distances as shown in Equation (23). The LSTM encoder takes a noisy input sequence,  $X_{noisy}$  with  $t$  timesteps and  $F$  features and produces a hidden representation,  $H_E$ , in Equation (24). This encoded sequence  $H_E$  is then repeated to match the original sequence length  $L$  as follows in (25). The repeated encoded sequence  $H_{rp}$  is applied on the layer (decoder) to generate a new hidden representation  $H_D$  as given in Equation (26). Finally, hidden representation  $H_D$  is processed with a TimeDistributed Dense layer to reconstruct sequence  $X_{Rec}$  as follows in Equation (27).

$$L_{Rec} = \sum_{i=1}^t ||X_i - \hat{X}_i||^2 \quad (23)$$

$$H_E = \text{LSTM}(X_{noisy}) \quad (24)$$

$$H_{rp} = \text{Repeat}(H_E, L) \quad (25)$$

$$H_D = \text{LSTM}(H_{rp}) \quad (26)$$

$$X_{\text{Rec}} = \text{TimeDistributed}(\text{Dense})(H_D) \quad (27)$$

#### 4.9.2. Prediction Branch

Similarly, to measure the error in the prediction process, we use prediction error  $L_{\text{pred}}$  which measures the discrepancy between predicted future values  $P$  and actual future values  $X_{\text{future}}$  over a defined future timeframe as shown in (28). The geometric distance is again used to quantify the discrepancy at each future timestep  $T_f$ , and these distances are summed up. The representation by encoder  $H_E$  undergoes processing via Dense layers to predict future values. The intermediate representation is  $H_{\text{predDense}}$ , and the final predicted values are  $Y_{\text{pred}}$  as given in Equations (29) and (30).

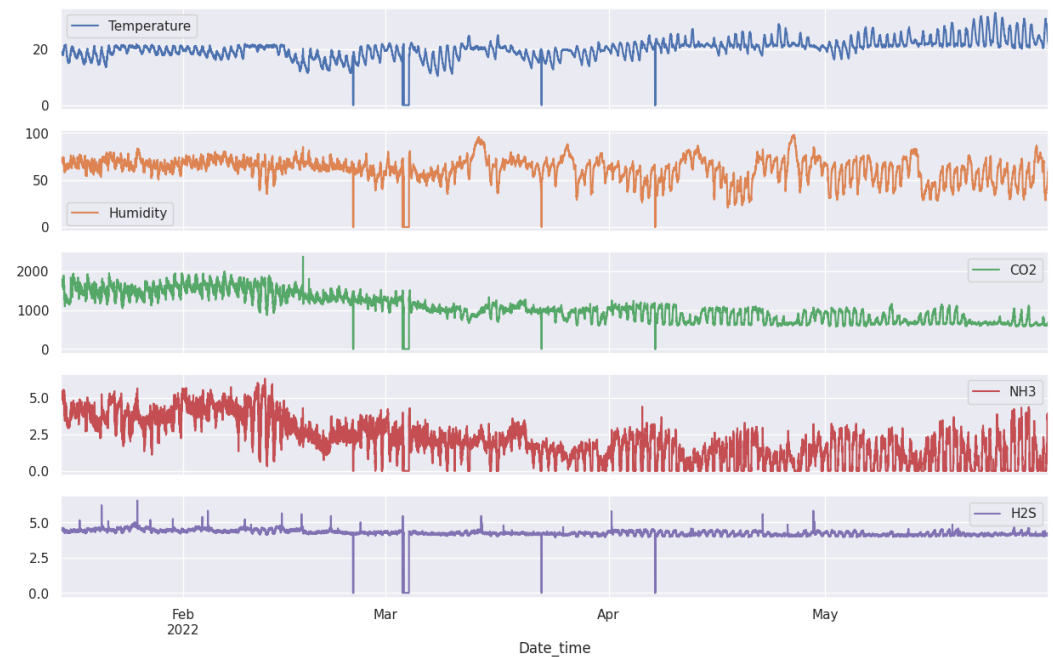
$$L_{\text{pred}} = \sum_{i=1}^{\text{future\_timesteps}} ||X_{\text{future},i} - P_i||^2 \quad (28)$$

$$H_{\text{predDense}} = \text{Dense}_{\text{relu}}(H_E) \quad (29)$$

$$Y_{\text{pred}} = \text{Dense}(H_{\text{predDense}}) \quad (30)$$

### 5. Dataset Analysis and Explanation

The datasets as shown in Figures 7 and 8 are composed of multiple time series produced by different sensors. These sensors help us maintain a comfortable environment for the pigs. We installed several sensors inside our pig barn to monitor temperature ( $^{\circ}\text{C}$ ), humidity (%), carbon dioxide  $\text{CO}_2$  (ppm), ammonia  $\text{NH}_3$  (ppm), and hydrogen sulfide  $\text{H}_2\text{S}$  (ppm).

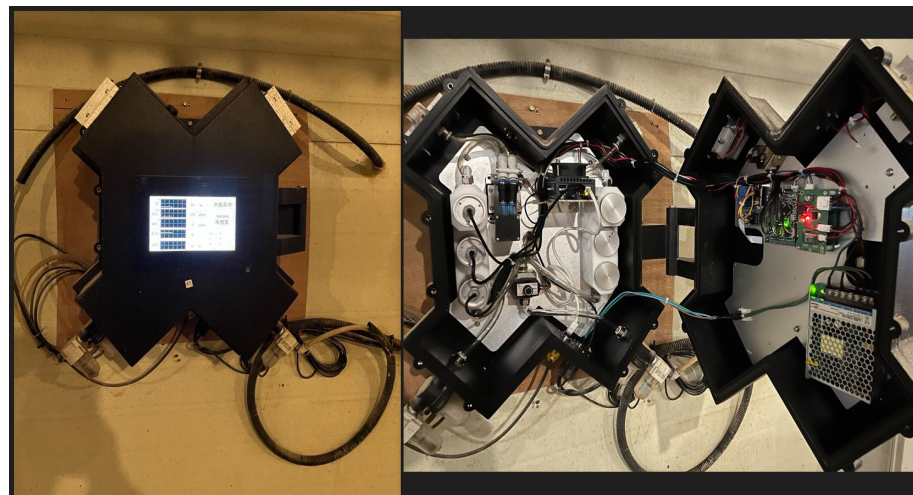


**Figure 7.** Fattening data collected indoors from a pig farm environment includes measurements of temperature, humidity,  $\text{CO}_2$ ,  $\text{NH}_3$ , and  $\text{H}_2\text{S}$ .



**Figure 8.** Outdoor data retrieved from the weather forecast station at the pig farm environment includes temperature, humidity, solar energy, and accumulated solar energy per day.

We installed weather forecasting devices outside to obtain accurate weather readings. These devices include temperature ( $^{\circ}\text{C}$ ), humidity (%), carbon dioxide  $\text{CO}_2$  (ppm), ammonia  $\text{NH}_3$  (ppm), and hydrogen sulfide  $\text{H}_2\text{S}$  (ppm). By monitoring both the inside and outside environment, we can make informed decisions to ensure the well-being of our pigs, as shown in Figure 9. Our setup consists of interconnected sensors that generate copious amounts of data, up to 39,456 data points per sensor. Additional details are found in Table 1.



**Figure 9.** Experimental setup involves the use of sensor detectors placed in a box equipped with a Jetson Nano board for monitoring indoor and outdoor environmental conditions in a pig farm.

All these sensors converge at a local server, Jetson Nano, where the data are immediately stored as they are collected. Acting as an intermediate repository and processing node, this Jetson Nano is then connected to a central server, which serves as a comprehensive data hub for all the collected data. This includes sensor data, imagery, and video feeds from cameras, all securely stored in real-time to ensure seamless consolidation, correlation, and analysis. As mentioned in the introduction in Section 1, the dataset is sourced from our pig

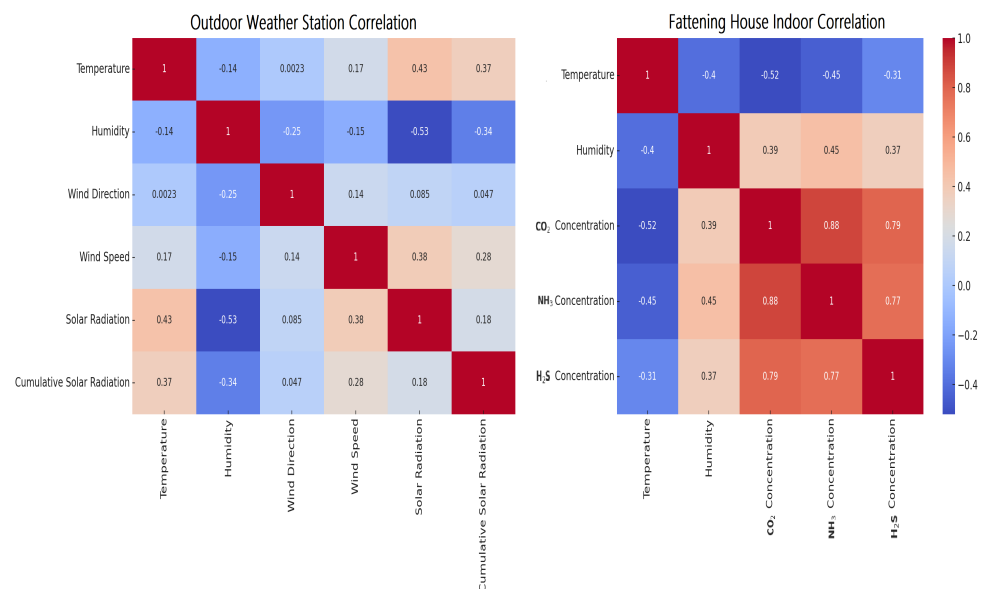
barn farm project. The sensors produce data with a 5-minute delay, creating a multivariate time series of nine features. These features include information about the indoor conditions of the barn, such as fattening and outdoor weather forecasts. For more detailed information, please refer to Table 1.

**Table 1.** A characterization of the various features in the Indoor and Outdoor fattening barn datasets.

| Indoor Fattening Barn          | Values |
|--------------------------------|--------|
| No of sensors                  | 5      |
| No of dataset samples          | 39,456 |
| No of minutes per instance     | 5      |
| No of the training set samples | 80%    |
| No of test set samples         | 10%    |
| No of validation set samples   | 10%    |
| Outdoor Weather Forecast       | Values |
| No of sensors                  | 4      |
| No of dataset samples          | 39,456 |
| No of minutes per instance     | 5      |
| No of the training set samples | 80%    |
| No of test set samples         | 10%    |
| No of validation set samples   | 10%    |

### 5.1. Data Analysis and Correlation

In livestock farming, environmental parameters play a vital role in animal health, growth, and productivity. Even minor changes in temperature or CO<sub>2</sub> levels can drastically affect the well-being of these animals as shown in Figure 10. Therefore, it is crucial to study moderate correlations that may seem insignificant at first glance.



**Figure 10.** Confusion matrices for outdoor (left) and indoor (right) pig farm datasets.

### 5.2. Outdoor Weather Station Correlation

Our research shows that temperature and humidity are negatively correlated, indicating that when temperature increases, humidity tends to decrease, and vice versa. On the other hand, solar radiation is positively correlated with cumulative solar radiation, which is expected as the latter is a cumulative measure. The other features do not exhibit strong correlations as shown in Figure 10 (left).



### 5.3. Fattening House Indoor Correlation

Figure 10 (right) shows that temperature and humidity are negatively correlated, which is the same as at the outdoor weather station, with similar data for outdoors. CO<sub>2</sub> concentration has a slight negative correlation with humidity. The other features in the fattening house data do not have strong correlations.

When analyzing data, it is crucial to consider the correlation between variables. The strength of the correlation can be classified into three categories.

#### 5.3.1. Weak Correlation

Even if the correlation is weak, it should still be considered if the variable has practical significance, such as CO<sub>2</sub> levels. General interpretations for the strength of the correlation coefficient ( $rr$ ) can be

$$(|rr| < 0.5) \quad (31)$$

#### 5.3.2. Moderate Correlation

Moderate correlations should be closely examined, as their combined effects or interactions with other variables may be important as shown in Equation (32).

$$0.5 \leq |rr| < 0.7 \quad (32)$$

#### 5.3.3. Strong Correlation

Strong correlations are obviously significant and should be included in the analysis. However, if using linear models, caution should be observed regarding multicollinearity as given in Equation (33).

$$(|rr| \geq 0.7) \quad (33)$$

## 6. Experimental Results and Discussion

### 6.1. Implementation Details

We developed and tested anomaly detection models for multivariate time series using a Shared LSTM Twin-Branch technique and a Multi-Head Attention mechanism. The models were implemented in Python 3.7 using popular frameworks such as Pandas, NumPy, Scikit-Learn, TensorFlow 2, and the Keras API. The experiments were performed on a PC equipped with an Intel (R) Core™ i7-10700 CPU @ 2.90 GHz and 32 GB of RAM. The model architecture parameters are explained in Table 2.

**Table 2.** Network architecture for Twin-Branch Shared LSTM with Multi-Head Attention. Bold emphasizes the sections in the table (Input, Attention layers, and Twin branch layers)

| Parameter                                                                         | Values           |
|-----------------------------------------------------------------------------------|------------------|
| Input Shape                                                                       | (24, 4), (24, 5) |
| DAE Gaussian Noise                                                                | 0.1              |
| No. of Shared LSTM Encoder Layers                                                 | 2                |
| Kernel regularization to each layer with strength                                 | L2(0.001)        |
| <b>Attention layers</b>                                                           |                  |
| • MultiHeadAttention layer<br>( $d_{model}, num_{heads}, num_{layers}$ )          | (64, 8, 0)       |
| • HierarchicalMultiHeadAttention<br>( $d_{model}, num_{heads}, num_{layers}$ )    | (64, 8, 4)       |
| • CrossMultiHeadAttention<br>( $d_{model}, num_{heads}, num_{layers}$ )           | (64, 8, 0)       |
| • FeatureEnhancedMultiHeadAttention<br>( $d_{model}, num_{heads}, num_{layers}$ ) | (64, 8, 4)       |
| <b>Twin-Branch layers (reconstruction and prediction)</b>                         |                  |
| Reconstruction Branch (Loss of weight)                                            | (64, 1.0)        |
| Prediction Branch (Loss of weight)                                                | (64, 0.5)        |
| Optimizer, Loss                                                                   | Adam, MSE        |
| Batch Size                                                                        | 24, 32           |
| Epoch, Callbacks, Feedback Loop                                                   | 20, 4, Yes       |

## 6.2. Performance Metrics

The process of evaluating an unlabeled anomaly detection model involves using various performance metrics, including mean squared error (MSE), mean absolute error (MAE), root mean squared error (RMSE), and the  $R^2$  score as shown in Equations (34)–(37). These metrics are instrumental in the accurate assessment of the model's performance. While MSE and MAE capture the reconstruction errors vital for anomaly detection, RMSE provides additional information on the magnitude of errors. In addition, the  $R^2$  score, although unusual in anomaly detection, helps in comparing the ability of our model to capture data variance with that of baseline models as given in the equations below. This comprehensive set of metrics ensures that our model's anomaly detection performance is robust, even in the absence of ground truth labels [11].

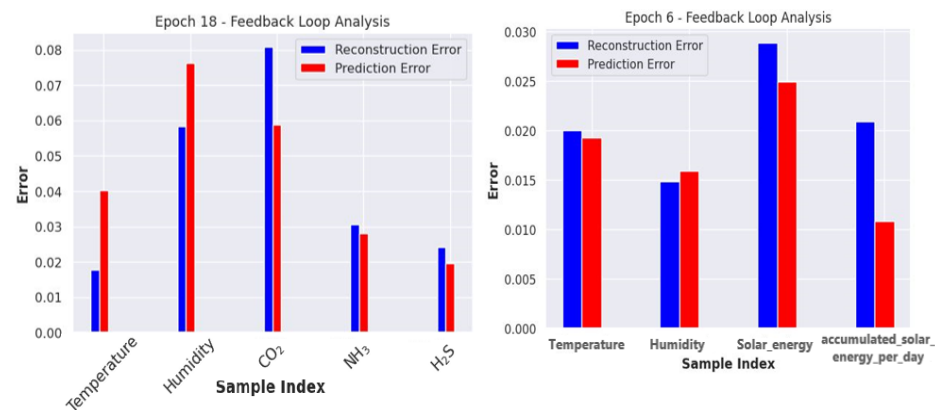
$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (34)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (35)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (36)$$

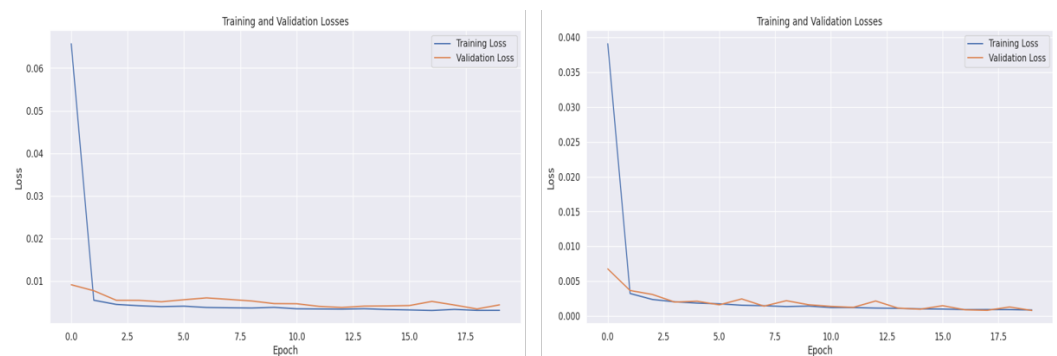
$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (37)$$

The model parameters and their respective values are detailed in Table 2. We split the raw data into training and validation sets to train our model, with an 80% and 10% split, respectively. Since we do not have separate data for testing, we set aside 10% of the data for testing the model, as given in Table 1. To normalize the data, we utilized the MinMaxScaler from the scikit-learn library in Python, which scales each feature to a specified range, usually between zero and one. Normalization guarantees that each feature contributes equally to the final prediction, which is particularly useful when dealing with features of varying magnitudes. It also aids in model convergence during training and ensures accurate model performance evaluation through reported metrics such as MSE, MAE, RMSE, and  $R^2$  Score across all features. The batch sizes of the dataset used for training the model varied from 24 to 32. To enhance the ability of the model to learn data representation and make the training more robust, we included Gaussian noise using DAE in the training set. The progress of model training was analyzed using a sophisticated monitoring technique known as the feedback loop paradigm [44], which allowed for a meticulous examination of the process. The reconstruction and prediction errors were comprehensively evaluated throughout each epoch and systematically recorded. These errors were visualized as a bar graph, revealing feature-specific and iteration-specific nuances, as demonstrated in Figure 11. This in-depth analysis facilitated a nuanced understanding of the model's learning dynamics, providing insights into the complex challenges encountered and overcome during training. Eventually, the model was successfully trained on the validation set after employing specific techniques.



**Figure 11.** Feedback loop at each iteration for the indoor (left) fattening dataset and outdoor (right) weather forecast dataset.

During the training of our model, we evaluated its performance on the validation dataset for each epoch. The graph in Figure 12 shows the model performance on the indoor fattening barn dataset and the outdoor weather forecast dataset. The graph highlights that the model learned quickly at the beginning of the training process, but the loss curves stabilized in the later epochs to avoid overfitting. We employed the *kernel regularization* technique to prevent overfitting, which is elaborated in Table 2. Additionally, we applied the early stopping technique to save computational resources.



**Figure 12.** A representation of the training and validation loss for the indoor (left) fattening barn dataset and outdoor (right) weather forecast dataset.

### 6.3. Anomaly Detection and Prediction for the Indoor Fattening Barn Environment Dataset

Our model is trained using a shared LSTM encoder with several Multi-Head Attention mechanisms and reconstructed with Twin-Branch techniques, which utilize reconstruction errors to identify abnormalities and prediction errors that enable accurate predictions of the following values in the sequence during runtime. The method assigns loss weights of 0.5 and 1.0, respectively, for the two branches involved in the process.

We comprehensively evaluated our proposed method by comparing it with several baseline models. To ensure a fair comparison between different models, we made sure to maintain identical experimental conditions. This involved using the same data splits for training and testing, applying the same preprocessing routines, and utilizing the exact architectures and parameters as specified in the original papers. This ensured that any observed differences in performance were solely due to the inherent properties of the models themselves, rather than any discrepancies in the methodology used to evaluate them. This validates the credibility of our approach and facilitates a fair comparison, underscoring the reliability and thoroughness of our study. TT-TBAD was conducted and compared against various state-of-the-art baseline models across multiple features. Table 3 displays the MSE, MAE, RMSE, and  $R^2$  scores for each feature. Our proposed

“TT-TBAD” methodology outperforms the other analyzed baseline models. Regarding predicting humidity, our model achieves an over  $R^2$  score = 0.93 accuracy in almost all feature cases except for  $H_2S$ .

**Table 3.** An evaluation of the proposed TT-TBAD model against benchmark models for various features of the indoor fattening barn environment.

| Feature                     | Method                 | MSE            | MAE           | RMSE         | $R^2$ Score |
|-----------------------------|------------------------|----------------|---------------|--------------|-------------|
| Temperature ( $^{\circ}C$ ) | LSTM-AE [21]           | 0.0023         | 0.018         | 0.015        | 0.84        |
|                             | GRU [22]               | 0.00048        | 0.015         | 0.026        | 0.78        |
|                             | LSTM [22]              | 0.0062         | 0.015         | 0.021        | 0.79        |
|                             | ConvLSTM-AE [22]       | 0.0045         | 0.0537        | 0.05         | 0.81        |
|                             | Attention-Bi-LSTM [39] | 0.0014         | 0.032         | 0.038        | 0.82        |
|                             | <b>TT-TBAD (ours)</b>  | <b>0.00011</b> | <b>0.012</b>  | <b>0.016</b> | <b>0.93</b> |
| Humidity (%)                | LSTM-AE [21]           | 0.0010         | 0.032         | 0.019        | 0.89        |
|                             | GRU [22]               | 0.0007         | 0.021         | 0.026        | 0.91        |
|                             | LSTM [22]              | 0.005          | 0.077         | 0.026        | 0.83        |
|                             | ConvLSTM-AE [22]       | 0.0032         | 0.046         | 0.048        | 0.86        |
|                             | Attention-Bi-LSTM [39] | 0.0019         | 0.029         | 0.033        | 0.93        |
|                             | <b>TT-TBAD (ours)</b>  | <b>0.0002</b>  | <b>0.014</b>  | <b>0.018</b> | <b>0.98</b> |
| $CO_2$ (ppm)                | LSTM-AE [21]           | 0.0011         | 0.053         | 0.039        | 0.71        |
|                             | GRU [22]               | 0.001          | 0.027         | 0.033        | 0.67        |
|                             | LSTM [22]              | 0.0006         | 0.01          | 0.033        | 0.79        |
|                             | ConvLSTM-AE [22]       | 0.0013         | 0.061         | 0.048        | 0.69        |
|                             | Attention-Bi-LSTM [39] | 0.0007         | 0.021         | 0.027        | 0.81        |
|                             | <b>TT-TBAD (ours)</b>  | <b>0.0001</b>  | <b>0.0098</b> | <b>0.013</b> | <b>0.94</b> |
| $NH_3$ (ppm)                | LSTM-AE [21]           | 0.0062         | 0.053         | 0.078        | 0.88        |
|                             | GRU [22]               | 0.001          | 0.032         | 0.042        | 0.83        |
|                             | LSTM [22]              | 0.001          | 0.029         | 0.042        | 0.85        |
|                             | ConvLSTM-AE [22]       | 0.0071         | 0.061         | 0.085        | 0.82        |
|                             | Attention-Bi-LSTM [39] | 0.0019         | 0.029         | 0.021        | 0.91        |
|                             | <b>TT-TBAD (ours)</b>  | <b>0.0011</b>  | <b>0.024</b>  | <b>0.013</b> | <b>0.94</b> |
| $H_2S$ (ppm)                | LSTM-AE [21]           | 0.071          | 0.035         | 0.10         | 0.11        |
|                             | GRU [22]               | 0.081          | 0.012         | 0.011        | 0.16        |
|                             | LSTM [22]              | 0.041          | 0.013         | 0.15         | −0.018      |
|                             | ConvLSTM-AE [22]       | 0.027          | 0.012         | 0.012        | 0.12        |
|                             | Attention-Bi-LSTM [39] | 0.00018        | 0.011         | 0.013        | 0.31        |
|                             | <b>TT-TBAD (ours)</b>  | <b>0.00013</b> | <b>0.0072</b> | <b>0.009</b> | <b>0.71</b> |

This could indicate potential data discrepancies. Unlike other models such as Attention and Bi-LSTM [39], which struggle with the complexities of their attention mechanisms, our model utilizes the power of the Multi-Head Attention technique and achieves higher precision. Through this distinction, the strength and ingenuity of our approach are highlighted.

After conducting a thorough collective comparison with other baseline models, as shown in Tables 4 and 5, our advanced TT-TBAD technique achieved a remarkable performance  $R^2$  score of 0.993 and 0.994, indicating its superior predictive capabilities. On the other hand, the “Attention Bi-LSTM” model falls behind. In addition, traditional approaches like LSTM [22] are not as effective, as their high error rates illustrate the challenges they face in understanding the complex dynamics of the barn environment.

We used the dynamic threshold approach based on rolling mean addition to  $k$  standard deviation to determine the threshold for the reconstruction error. This is a flexible method for anomaly detection in time series data. It can adjust to changing data patterns and is suitable for many applications, especially in dynamic environments, as shown in Equation (38). This approach is more effective than the traditional technique of taking the average of the five highest loss values from the validation set [45]. Our model can identify anomalies in the data by using the predetermined dynamic threshold. If the reconstruction

loss exceeds this threshold, the model will recognize those values as abnormal; Ref. [39], as shown in Figure 13.

$$\text{Dynamic Threshold}_t = \text{RollingMean}_t + k \times \text{RollingStdDev}_t \quad (38)$$

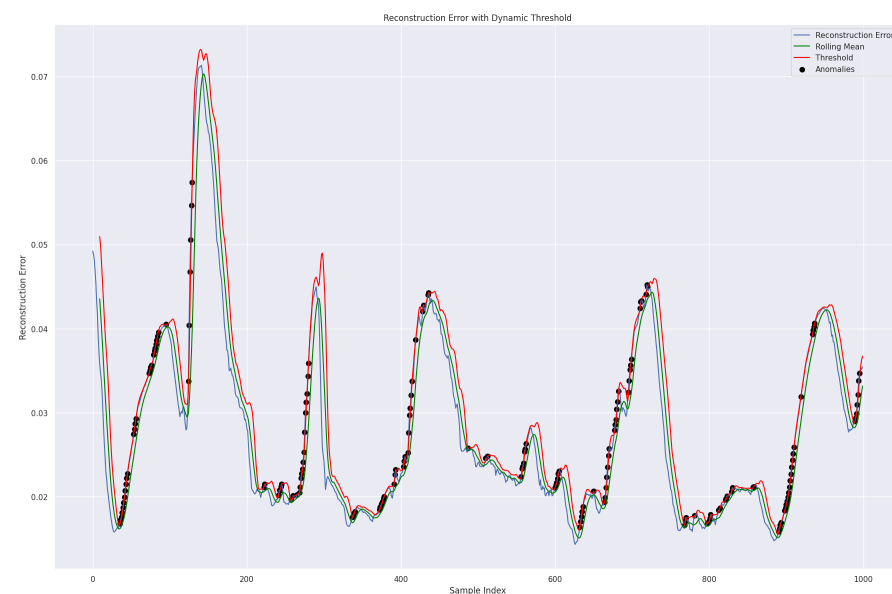
Meanwhile, the RollingMean represents the reconstruction error, and the RollingStdDev is the standard deviation of observations over a specified window and time.  $k$ , a predefined constant, determines the number of standard deviations from the rolling mean required to qualify a point as an anomaly.

**Table 4.** A comparative analysis of various anomaly detection models for the indoor fattening barn environment.

| Methods                | MSE             | MAE           | RMSE            | $R^2$ Score  |
|------------------------|-----------------|---------------|-----------------|--------------|
| LSTM-AE [21]           | 0.000112        | 0.0687        | 0.00106         | 0.89         |
| GRU [22]               | 0.00071         | 0.0220        | 0.0086          | 0.88         |
| LSTM [22]              | 0.00140         | 0.0196        | 0.0533          | 0.83         |
| ConvLSTM-AE [22]       | 0.0036          | 0.0446        | 0.00367         | 0.90         |
| Attention-Bi-LSTM [39] | 0.00104         | 0.0147        | 0.0340          | 0.97         |
| <b>TT-TBAD (ours)</b>  | <b>0.000404</b> | <b>0.0137</b> | <b>0.000404</b> | <b>0.993</b> |

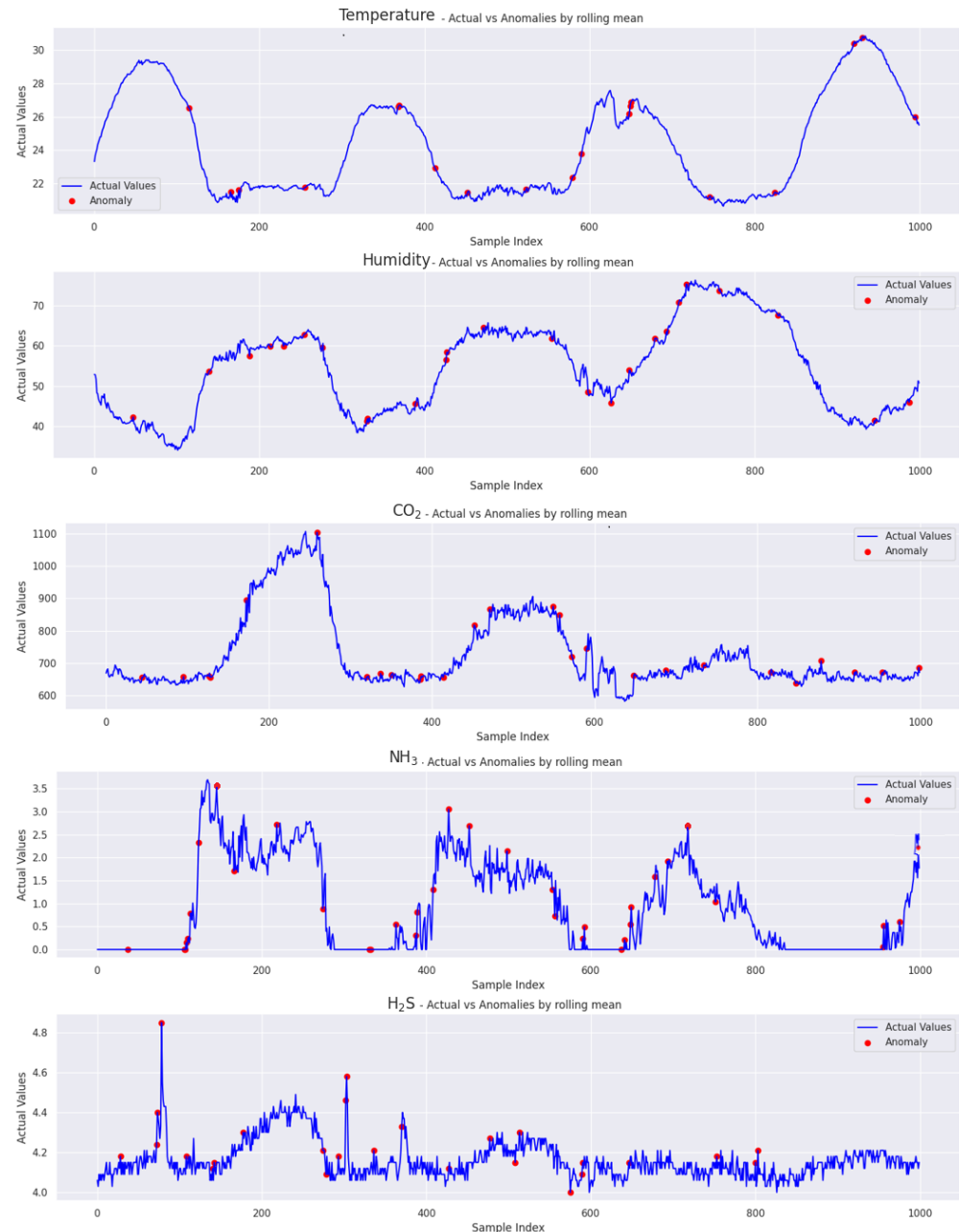
**Table 5.** A comparative analysis of various anomaly detection models for the outdoor fattening barn environment.

| Methods                | MSE             | MAE           | RMSE           | $R^2$ Score  |
|------------------------|-----------------|---------------|----------------|--------------|
| LSTM-AE [21]           | 0.00311         | 0.0477        | 0.0506         | 0.95         |
| GRU [22]               | 0.0020          | 0.0342        | 0.00202        | 0.90         |
| LSTM [22]              | 0.00471         | 0.0509        | 0.0638         | 0.86         |
| ConvLSTM-AE [22]       | 0.00367         | 0.0400        | 0.0491         | 0.96         |
| Attention-Bi-LSTM [39] | 0.0010          | 0.0197        | 0.0342         | 0.97         |
| <b>TT-TBAD (ours)</b>  | <b>0.000618</b> | <b>0.0120</b> | <b>0.00741</b> | <b>0.994</b> |



**Figure 13.** A representation of the proposed TT-TBAD model for anomaly detection with a dynamic threshold based on the rolling mean and reconstruction error over a specified window and time for the indoor fattening barn environment.

Smoothing the errors by averaging during the rolling mean calculation over a specified data window helps to reduce the reconstruction error. As a result, the green line representing the rolling mean appears smooth. Validation data were utilized for testing purposes, and we present the results in Figure 14.



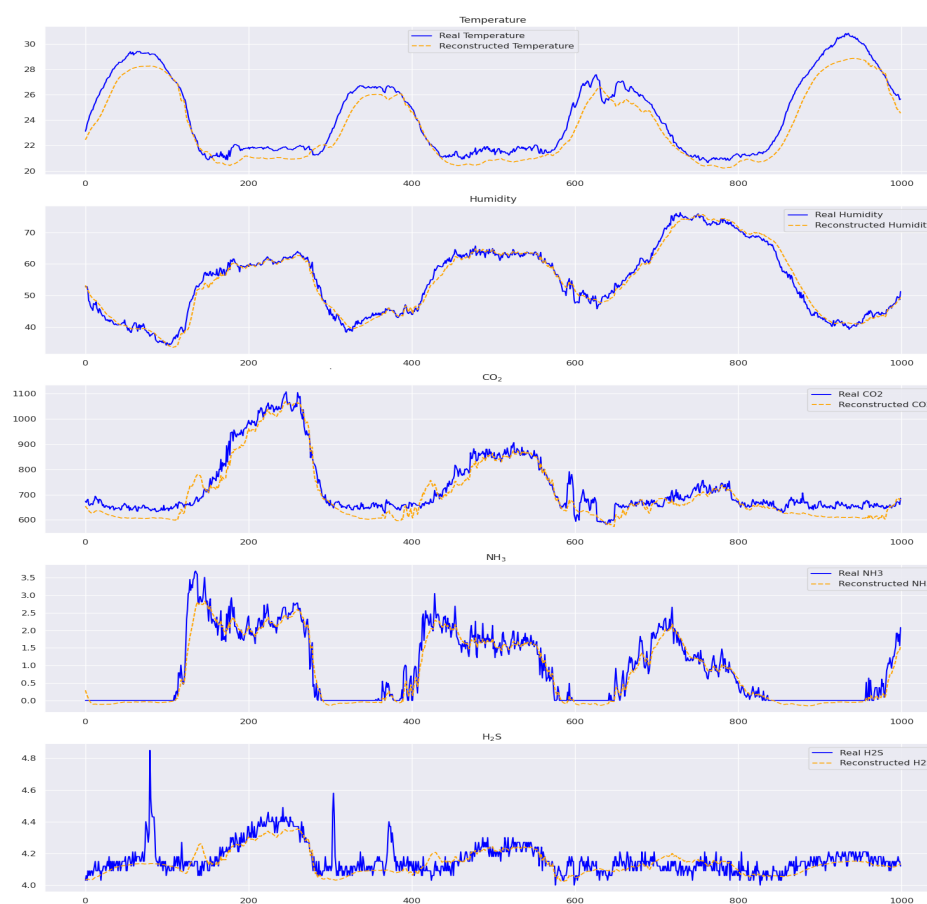
**Figure 14.** A representation of the proposed TT-TBAD model for anomaly detection with a dynamic threshold based on the rolling mean and reconstruction error over a specified window and time for each feature in the indoor fattening barn environment.

The figure includes five line charts that display the comparison between the anomalies and actual values for each feature separately, based on a small portion of the validation set. Each chart plots data points over a common x-axis, with several samples for each feature, including temperature, humidity, CO<sub>2</sub>, NH<sub>3</sub>, and H<sub>2</sub>S. After analyzing 39,456 samples, we identified anomalies in varying percentages across different metrics. The temperature metric showed a 3.29% anomaly rate, humidity at 3.39%, CO<sub>2</sub> at 3.58%, and NH<sub>3</sub> stood out with a higher rate of 4.19%.



Interestingly, H<sub>2</sub>S displayed the lowest anomaly rate at 2.62%. This is due to the model's higher reconstruction and prediction error, with an  $R^2$  score = 0.71. In comparison, other features exhibited metrics indicating better performance. This discrepancy emphasizes how model accuracy and prediction errors can impact anomaly detection.

The model predictions for temperature and humidity in the prediction phase are quite accurate, indicating its reliability. However, there are some noticeable discrepancies in temperature around indices 200, 600, and 800 and humidity near 800, which may be due to external factors or model limitations. CO<sub>2</sub> predictions match well with actual data, but some deviations at indices 200 and 600 could be attributed to environmental changes. On the other hand, predicting NH<sub>3</sub> and H<sub>2</sub>S levels is more challenging, with more pronounced fluctuations, particularly around indices 200–800 for NH<sub>3</sub> and 200–400 for H<sub>2</sub>S, highlighting potential challenges or sporadic environmental events. Moreover, predicting the lines using H<sub>2</sub>S presents challenges, making the model's future job more challenging. The deviations in these metrics underscore the areas for model improvement, as depicted in Figure 15.

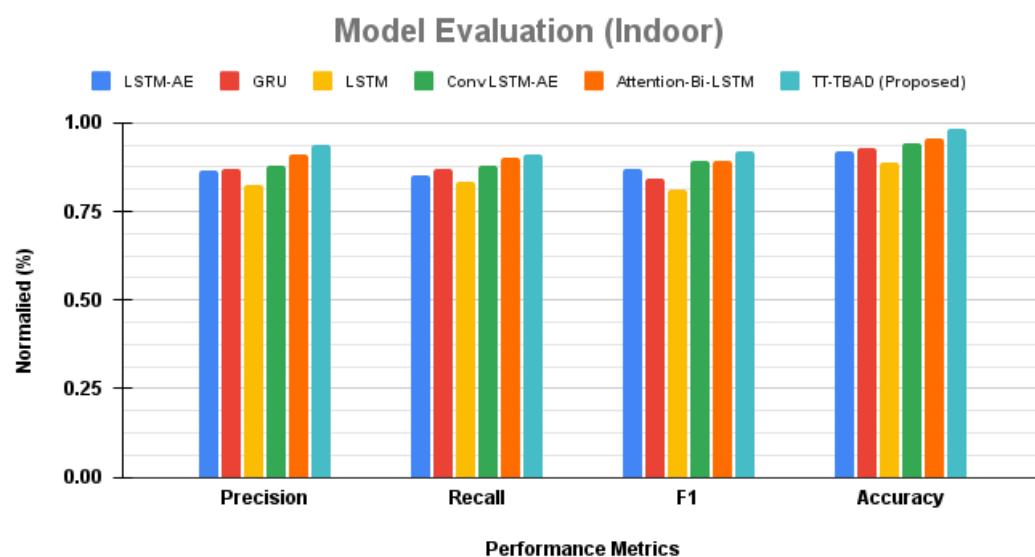


**Figure 15.** A comparison of the proposed TT-TBAD model for actual and predicted data points in the indoor fattening barn environment.

#### 6.4. Comparative Analysis of Anomaly Detection in Indoor Environment

The proposed TT-TBAD for anomaly detection is evaluated against state-of-the-art models, including LSTM-AE, GRU, LSTM, ConvLSTM-AE, and Attention-Bi-LSTM, with an assessment based on accuracy [46], precision [47], recall [48], and the F1 score [49]. The comparative study involves the utilization of a data annotation mechanism to label the partial test dataset, ensuring suitability for the detection evaluation. The results are presented in Figure 16, revealing that the LSTM-AE model demonstrated a robust balance between precision (0.867) and recall (0.853), resulting in a solid F1 score of 0.868. The GRU slightly outperformed, achieving precision and recall values of 0.87 and 0.871, respectively,

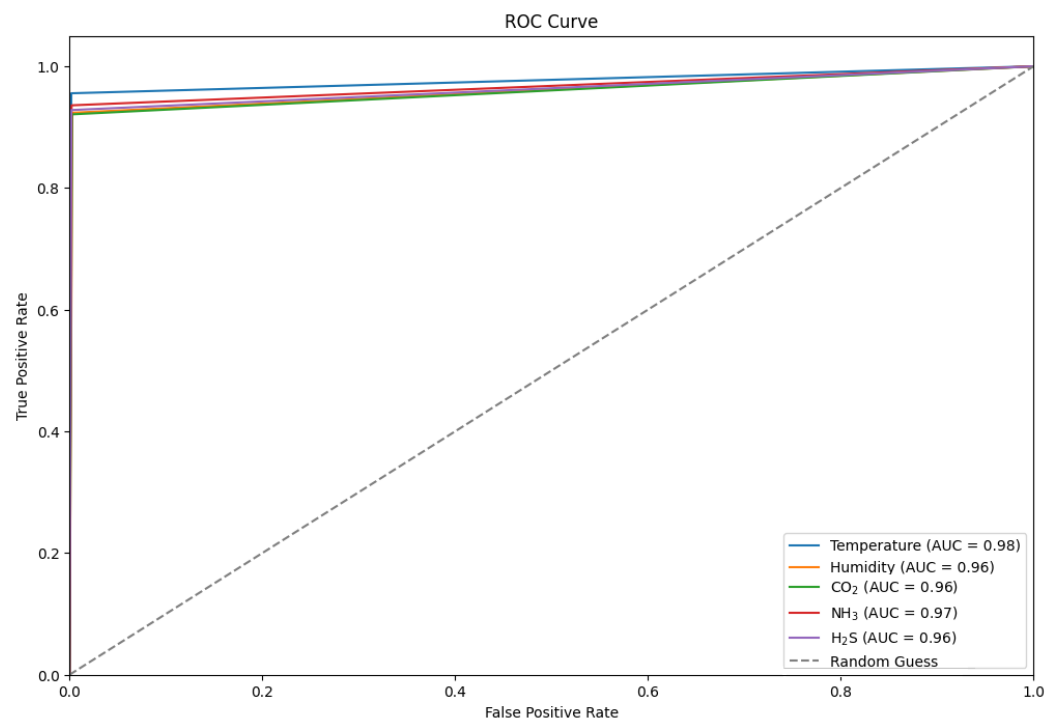
although there was a modest dip in the F1 score to 0.844. The standard LSTM lagged behind, with precision and recall scores of 0.824 and 0.835, hinting at a potential decrease in anomaly classification accuracy, as reflected by an F1 score of 0.809. In contrast, the ConvLSTM-AE model excelled, boasting precision and recall scores of 0.881 and 0.879, along with an impressive F1 score of 0.893, indicating enhanced anomaly detection capability. Additionally, the Attention-Bi-LSTM model achieved the highest precision at 0.91 and a recall of 0.902, resulting in an F1 score of 0.891, showcasing strong predictive confidence. However, the spotlight is on our TT-TBAD model, surpassing all others with outstanding precision (0.94), recall (0.913), and F1 score (0.92), coupled with an unmatched accuracy of 0.983. This establishes a new standard in anomaly detection, highlighting our model's exceptional precision and reliability.



**Figure 16.** A comparative study of the proposed TT-TBAD for anomaly detection in indoor environments against various state-of-the-art anomaly detection models.

The Receiver Operating Characteristic (ROC) curve highlights the effectiveness of environmental classifiers in detecting gases within an enclosed setting. Among the classifiers, temperature demonstrated the highest performance with an AUC score of 0.98.  $\text{NH}_3$ ,  $\text{CO}_2$ , and  $\text{H}_2\text{S}$  classifiers also performed exceedingly well with AUC scores ranging from 0.96 to 0.97. The humidity classifier had an AUC score of 0.96, indicating its potential effectiveness in detecting gases.

These AUC values presented in Figure 17, markedly higher than the random guess baseline (AUC = 0.5), reflect the classifiers' high accuracy and confirm their potential for reliable indoor air quality monitoring, offering substantial benefits for safety and hazard prevention in controlled environments.



**Figure 17.** A ROC curve of the proposed TT-TBAD for five classifiers based on environmental parameters of the barn dataset in an indoor environment.

#### 6.5. Anomaly Detection and Prediction for Outdoor Weather Forecast for the Barn Environment Dataset

Utilizing the same effective methodology employed in the previous dataset, we applied our method with great attention to detail to ensure uniformity and comparability in our analysis. Upon thoroughly examining Table 6, it is evident that the TT-TBAD method consistently outperforms its counterparts.

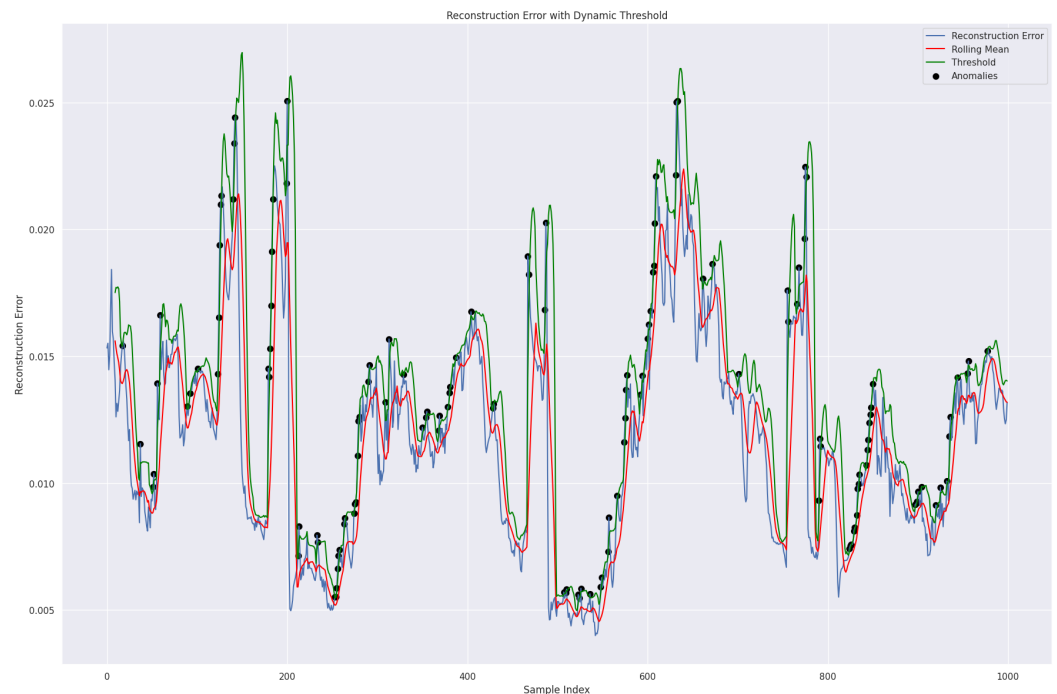
The performance of various models was evaluated based on different features such as temperature, humidity, solar energy, and accumulated solar energy per day. The TT-TBAD method displayed superior precision for temperature and humidity, with the lowest MSE, MAE, and RMSE values and  $R^2$  scores = 0.99 and 0.99, respectively. In solar energy, the TT-TBAD method outperforms other models with its impressive  $R^2$  score = 0.94 for Attention-Bi-LSTM and 0.98 for accumulated solar energy/day. Although Attention-Bi-LSTM has slightly better error metrics, TT-TBAD stands out for its adaptability and resilience, making it a reliable and versatile model for predictive analysis. Its consistency in performance across various datasets underlines its potential as an excellent choice for predictive analysis of solar energy.

The study used various modeling techniques for outdoor weather forecasting in a fattening barn environment. The results showed that our TT-TBAD method performed the best with the lowest mean squared error (MSE) of 0.000618 and an excellent  $R^2$  Score = 0.99. This demonstrates the exceptional predictive accuracy of our method in this particular setting. Other methods, such as LSTM-AE and ConvLSTM-AE, also performed well, with  $R^2$  Scores = 0.95 and 0.96, respectively. However, Attention-Bi-LSTM achieved the highest  $R^2$  score = 0.97. GRU slightly surpassed LSTM-AE with an  $R^2$  score = 0.90.

Figure 18 illustrates the same dynamic threshold approach is used for outdoor weather forecast datasets to identify anomalies in the data compared to the reconstruction error. The black data points indicate the anomalies, meaning the model finds it difficult to reconstruct them accurately. The blue line represents the difference between the original and reconstructed data.

**Table 6.** An evaluation of the proposed TT-TBAD model against benchmark models for various features of the **outdoor** fattening barn environment.

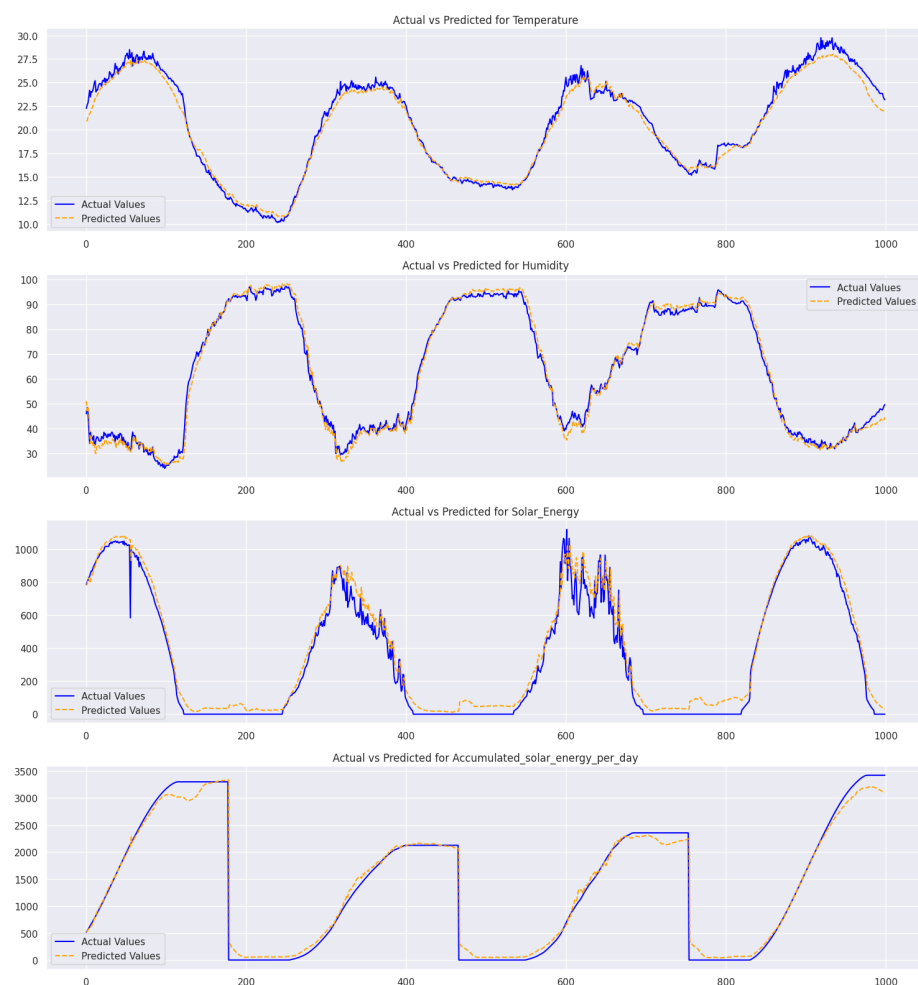
| Feature                                                | Method                 | MSE            | MAE            | RMSE           | $R^2$ Score |
|--------------------------------------------------------|------------------------|----------------|----------------|----------------|-------------|
| Temperature ( $^{\circ}\text{C}$ )                     | LSTM-AE [21]           | 0.0029         | 0.0415         | 0.0542         | 0.88        |
|                                                        | GRU [22]               | 0.00197        | 0.0372         | 0.0444         | 0.91        |
|                                                        | LSTM [22]              | 0.00254        | 0.04427        | 0.05045        | 0.90        |
|                                                        | ConvLSTM-AE [22]       | 0.00306        | 0.0439         | 0.05531        | 0.88        |
|                                                        | Attention-Bi-LSTM [39] | 0.00075        | 0.0216         | 0.0275         | 0.96        |
|                                                        | <b>TT-TBAD (ours)</b>  | <b>0.00026</b> | <b>0.00418</b> | <b>0.00462</b> | <b>0.99</b> |
| Humidity (%)                                           | LSTM-AE [21]           | 0.0049         | 0.0515         | 0.0705         | 0.93        |
|                                                        | GRU [22]               | 0.00092        | 0.0224         | 0.0303         | 0.95        |
|                                                        | LSTM [22]              | 0.00174        | 0.0342         | 0.0417         | 0.96        |
|                                                        | ConvLSTM-AE [22]       | 0.00371        | 0.04481        | 0.06096        | 0.94        |
|                                                        | Attention-Bi-LSTM [39] | 0.0013         | 0.0290         | 0.0301         | 0.97        |
|                                                        | <b>TT-TBAD (ours)</b>  | <b>0.00045</b> | <b>0.0151</b>  | <b>0.0211</b>  | <b>0.99</b> |
| Solar Energy ( $\text{w}/\text{m}^2$ )                 | LSTM-AE [21]           | 0.0050         | 0.0490         | 0.0712         | 0.72        |
|                                                        | GRU [22]               | 0.00142        | 0.0241         | 0.0377         | 0.86        |
|                                                        | LSTM [22]              | 0.01404        | 0.0843         | 0.0102         | 0.81        |
|                                                        | ConvLSTM-AE [22]       | 0.00231        | 0.03435        | 0.04808        | 0.87        |
|                                                        | Attention-Bi-LSTM [39] | 0.0014         | 0.0244         | 0.0383         | 0.91        |
|                                                        | <b>TT-TBAD (ours)</b>  | <b>0.0011</b>  | <b>0.0201</b>  | <b>0.0271</b>  | <b>0.94</b> |
| Accumulated Solar Radiation ( $\text{J}/\text{cm}^2$ ) | LSTM-AE [21]           | 0.0074         | 0.0489         | 0.0965         | 80.0        |
|                                                        | GRU [22]               | 0.00177        | 0.02591        | 0.0421         | 0.93        |
|                                                        | LSTM [22]              | 0.0040         | 0.0409         | 0.0638         | 0.89        |
|                                                        | ConvLSTM-AE [22]       | 0.00561        | 0.03727        | 0.07496        | 0.85        |
|                                                        | Attention-Bi-LSTM [39] | 0.00172        | 0.0138         | 0.0415         | 0.95        |
|                                                        | <b>TT-TBAD (ours)</b>  | <b>0.00075</b> | <b>0.0084</b>  | <b>0.0271</b>  | <b>0.98</b> |

**Figure 18.** A representation of the proposed TT-TBAD model for anomaly detection with a dynamic threshold based on rolling mean and reconstruction error over a specified window and time for the outdoor fattening barn environment.

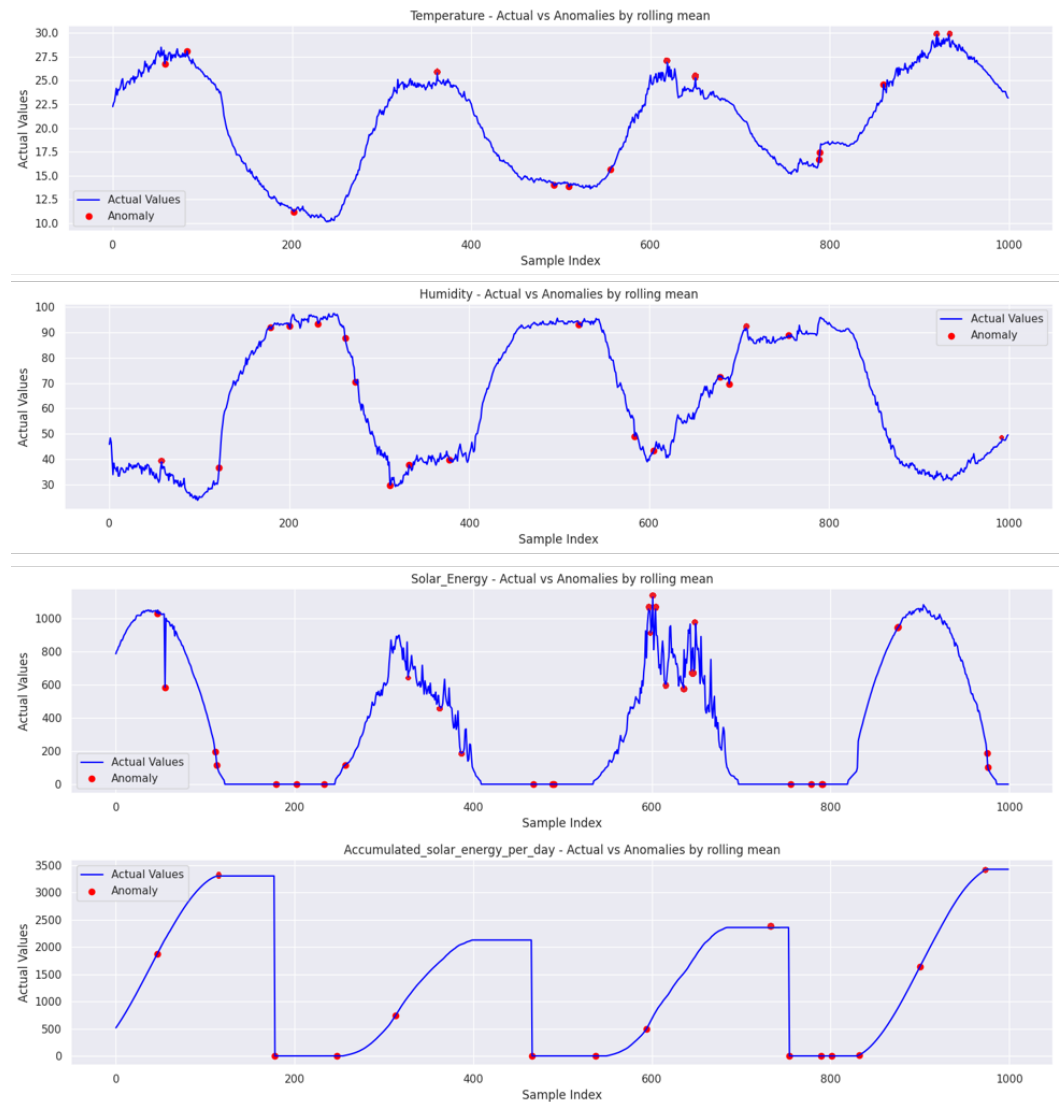
The solar energy graph has pronounced fluctuations with several anomalies, and the accumulated solar energy chart has a step-like pattern, with anomalies signifying unex-

pected drops or surges. These graphs emphasize the need to monitor environmental and energy parameters for consistency. The graphs in Figure 19 compare actual and predicted values. While the model predictions for temperature and humidity are highly accurate, noticeable discrepancies are evident in the solar energy graph. These inconsistencies underscore the imperative need to develop and refine prediction mechanisms, particularly in the context of solar energy parameters. It is crucial to address these discrepancies to enhance the accuracy of the model predictions.

Upon analyzing the data using the rolling mean approach, it becomes evident that metrics related to solar energy, such as solar energy and accumulated solar energy per day, exhibit a slightly higher percentage of anomalies, approximately 4.43% and 4.54%, respectively, as shown in Figure 20. In contrast, anomalies in temperature and humidity are only slightly lower, at around 3.36% and 3.44%, respectively. This discrepancy suggests that the solar metrics may be more susceptible to abnormalities or external factors influencing their readings when compared to the more consistent readings of temperature and humidity.



**Figure 19.** Comparison of actual and predicted values of a small section of the validation set in outdoor weather forecasts in a barn environment for each feature using a multivariate Twin-branch Shared LSTM with Multi-Head Attention.



**Figure 20.** A comparison of the proposed TT-TBAD model for actual and detected anomaly data points in the outdoor fattening barn environment.

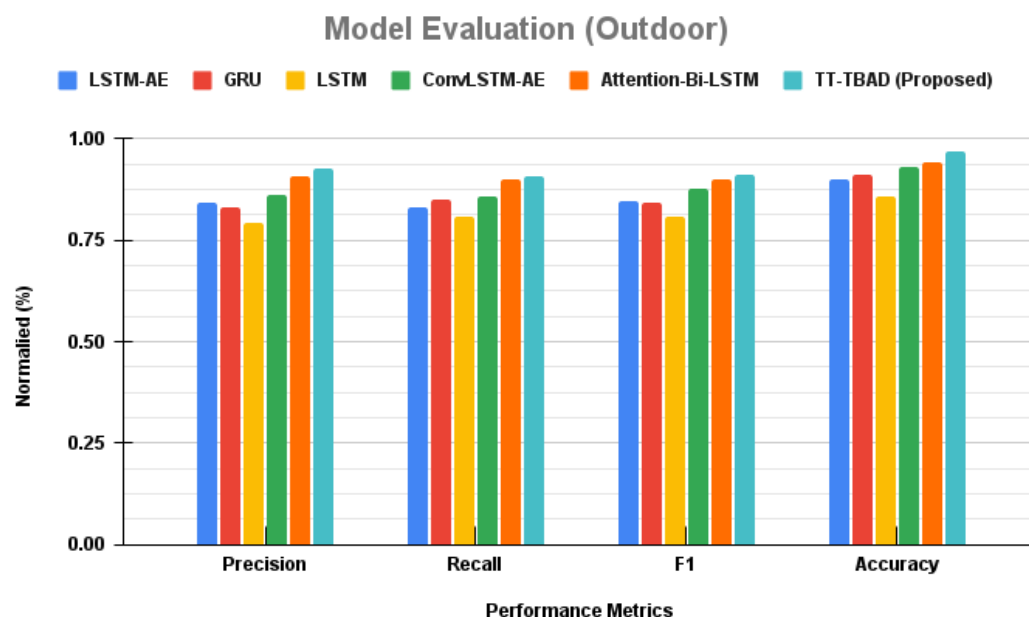
#### 6.6. Comparative Analysis of Anomaly Detection Models in Outdoor Environment

A comparative study of anomaly detection is performed for various models, and the results are presented in Figure 21. The figure shows that the LSTM-AE model achieves a solid precision rate of 0.842 paired with a recall of 0.83, harmonizing into an F1 score of 0.848. Nipping at its heels, the GRU model registers a precision of 0.83 with an incrementally higher recall of 0.851, achieving an F1 score of 0.841. Transitioning to the LSTM model, we observe a dip in precision to 0.792 and recall to 0.809, reflected in a modest F1 score of 0.81. In contrast, the ConvLSTM-AE model emerges stronger, with a precision of 0.862 and recall of 0.859, coalescing into a solid F1 score of 0.876. A leap forward is seen with the Attention-Bi-LSTM model, which achieves a notable precision of 0.909 and a recall of 0.899, commanding an F1 score of 0.901. The crescendo of our analysis is reached with the introduction of our TT-TBAD model. It sets a new benchmark with a precision of 0.927 and a recall of 0.909, melding into an F1 score of 0.91. This performance is further underscored by a remarkable accuracy of 0.971, positioning our model at the apex of anomaly detection capabilities.

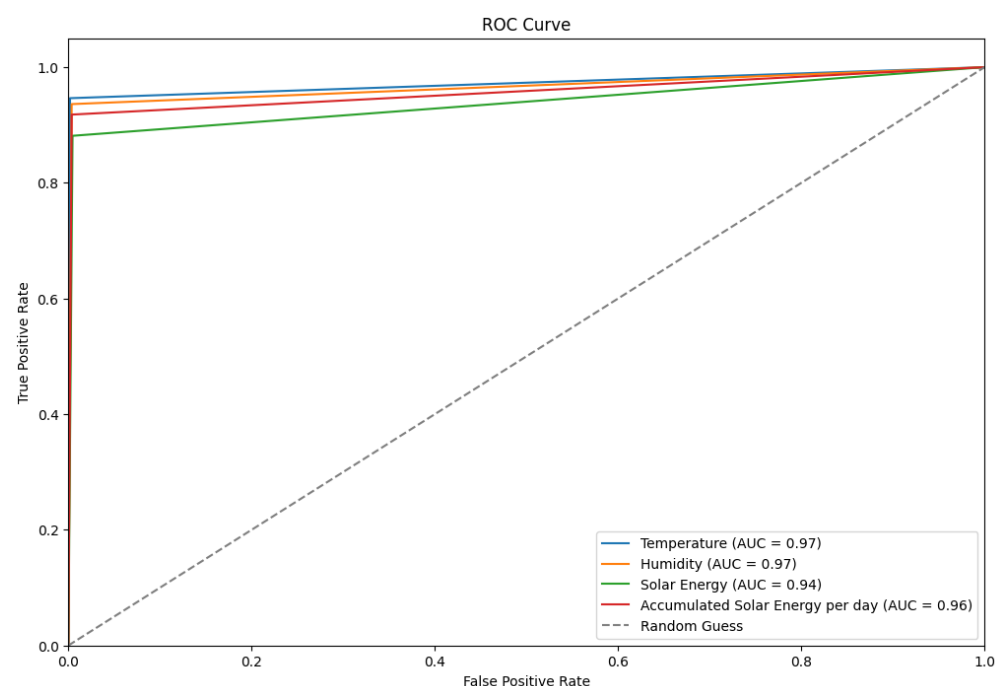
Moreover, the ROC curve presented in Figure 22 demonstrates the discriminative power of four classifiers based on environmental parameters. The figure shows AUC scores of 0.97 for both temperature and humidity, followed closely by accumulated solar energy



per day at 0.96 and solar energy at 0.94. The classifiers exhibit robust predictive capabilities, with their performance significantly surpassing the random classification benchmark (AUC = 0.5). This underscores their potential utility in sophisticated environmental sensing and prediction applications.



**Figure 21.** A comparative study of the proposed TT-TBAD for anomaly detection in outdoor environments against various state-of-the-art anomaly detection models.



**Figure 22.** An ROC curve of the proposed TT-TBAD for four classifiers based on environmental parameters of the barn dataset in an outdoor environment.

## 7. Discussion

The TimeTector-Twin-Branch Shared LSTM encoder (TT-TBAD) is a revolutionary development in the field of agricultural data analytics. It is particularly useful for detecting anomalies in time series data, which can be noisy and difficult to analyze. This model is de-

signed to understand agricultural datasets and redefine the standards of anomaly detection by identifying genuine irregularities that are crucial for operational farm decision making.

At its core, the proposed method uses a shared LSTM encoder, which is known for its ability to comprehend temporal data, along with advanced attention mechanisms. These mechanisms filter through data to eliminate irrelevant noise and highlight critical anomalies. This is especially important in agriculture, where distinguishing between benign fluctuations and significant outliers is essential for crop yield and livestock health. TT-TBAD also uses a Denoising Autoencoder (DAE) to counter the challenge of overfitting. This ensures that the model remains robust and generalizable, even when faced with new data. The Twin-Branch design of the model allows for it to simultaneously reconstruct data and predict future environmental and biological shifts, providing valuable predictive foresight for farmers which traditional models failed to provide. What sets TT-TBAD apart is its ability to balance architectural complexity with computational efficiency. Despite its advanced design, TT-TBAD has commendable training durations between 380 and 400 s, meaning that it respects the time-sensitive demands of agricultural operations. This efficiency is achieved without sacrificing the depth and breadth of its data analysis, maintaining high fidelity in anomaly detection and prediction. TT-TBAD introduces a dynamic thresholding technique based on a rolling mean, which adapts to evolving data patterns. This fine-tunes the model's sensitivity to fluctuations and enhances its accuracy, making it an indispensable asset in precision farming technologies. In sum, TT-TBAD is an excellent example of meticulous data analysis, advanced predictive modeling, and practical computational application. Its unmatched accuracy in detecting and predicting anomalies, underpinned by a dynamic and adaptive thresholding strategy, makes it a valuable tool for precision farming technologies that work in harmony with nature and machine learning innovation.

#### *Limitations and Assumptions*

The model is tested and found to be efficient in identifying any unusual or abnormal data points within the current datasets. However, its scalability and complexity make it challenging to process larger datasets, which requires longer training periods and more computational power. Moreover, parameter tuning has become more complex, and refined optimization techniques are necessary. The model's anomaly detection capability depends on the quality of the input data, and it is crucial to judiciously integrate noise to preserve its effectiveness. The model's design is based on fundamental assumptions about input data and training parameters, such as noise levels, sequence sizing, and batch processing, which aim to improve its performance in detecting anomalies within agricultural contexts.

## **8. Conclusions**

In our research paper, we introduce the TimeTector-Twin-Branch Shared LSTM encoder (TT-TBAD) model which employs a Twin-Branch Shared LSTM encoder and multiple Multi-Head Attention mechanisms to detect anomalies and make real-time predictions from multivariate time series sensor data. With the advent of deep learning algorithms, the challenge of capturing spatial-temporal correlations in multi-sensor time series data is efficiently addressed. These algorithms excel at generalizing normal, noisy, and abnormal data patterns, making them an ideal solution for handling complex data structures. Our model is a fusion of several robust frameworks. It incorporates the Denoising Autoencoder (DAE) to characterize multi-sensor time series signals, thus minimizing the risk of overfitting caused by noise and anomalies in the training data. Additionally, we utilize various Multi-Head Attention mechanisms such as Multi-Head Attention (MHA), Cross-Multi-Head Attention (C-MHA), Hierarchical Multi-Head Attention (Hi-MHA), and Feature-Enhanced Multi-Head Attention (FH-MHA). These mechanisms enable the model to capture diverse and hierarchical relationships, recognize cross-sequence patterns, and leverage external features, resulting in a comprehensive understanding of complex datasets. Our experimental evaluation is conducted on proprietary livestock barn environmental datasets using

performance metrics for both anomaly detection and future prediction, where our model demonstrates outstanding performance compared to benchmark traditional models with detection accuracies and  $R_2$  scores of 97.1%, 98.3% and 0.994%, 0.993%.

In our future work, we aim to develop a smart agriculture livestock environment that allows real-time monitoring through the barn sensors. The ultimate aim is to automate livestock farming to the greatest possible extent.

**Author Contributions:** Conceptualization, J.K.K. and S.C.K.; methodology, J.K.K. and S.H.; supervision, S.C.K.; original writing, J.K.K.; revision, J.K.K. and S.H.; visualization, J.K.K.; dataset collection, J.K.K. and S.H.; funding acquisition, S.C.K. and H.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry (IPET) and the Korea Smart Farm R&D Foundation through the Smart Farm Innovation Technology Development Program, funded by the Ministry of Agriculture, Food and Rural Affairs (MAFRA), Ministry of Science and ICT (MSIT), and Rural Development Administration (RDA) (1545027910), and by the National Research Foundation (NRF) funded by the Ministry of Education (NRF-2019R1A6A1A09031717).

**Data Availability Statement:** Due to the proprietary nature of the dataset which serves as a foundational element of our project, we are unable to make it available to the public. We recognize the importance of data transparency and reproducibility in research; however, the confidentiality agreements and the sensitive nature of the data necessitate these restrictions. We are committed to ensuring the integrity of our research within these constraints and are available to answer any queries related to our methodologies and findings, to the extent possible without compromising the confidentiality of the dataset.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Wu, R.; Keogh, E.J. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 2421–2429.
2. Park, H.; Park, D.; Kim, S. Anomaly Detection of Operating Equipment in Livestock Farms Using Deep Learning Techniques. *Electronics* **2021**, *10*, 1958. <https://doi.org/10.3390/electronics10161958>.
3. Tebueva, F.B.; Kopytov, V.Y.; Petrenko, V.I.; Shulgin, A.O.; Demirtchev, N.G. The Identification Of Data Anomalies From Information Sensors Based On The Estimation Of The Correlation Dimension Of The Time Series Attractor In Situational Management Systems. *J. Theor. Appl. Inf. Technol.* **2018**, *96*, 2197–2207.
4. Guarino, M.; Norton, T.; Berckmans, D.; Vranken, E.; Berckmans, D. A blueprint for developing and applying precision livestock farming tools: A key output of the EU-PLF project. *Anim. Front.* **2017**, *7*, 12–17. <https://doi.org/10.2527/af.2017.0103>.
5. Vranken, E.; Berckmans, D. Precision livestock farming for pigs. *Anim. Front.* **2017**, *7*, 32–37. <https://doi.org/10.2527/af.2017.0106>.
6. Alarcón, L.V.; Allepuz, A.; Mateu, E. Biosecurity in pig farms: A review. *Porc. Health Manag.* **2021**, *7*, 1–15.
7. Zhang, C.; Song, D.; Chen, Y.; Feng, X.; Lumezanu, C.; Cheng, W.; Ni, J.; Zong, B.; Chen, H.; Chawla, N.V. A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 29–31 January 2019; pp. 1409–1416. <https://doi.org/10.1609/aaai.v33i01.33011409>.
8. Cai, S.; Chen, J.; Yin, B.; Sun, R.; Zhang, C.; Chen, H.; Chen, J.; Lin, M. An efficient outlier detection approach for streaming sensor data based on neighbor difference and clustering. *Secur. Commun. Netw.* **2022**, *2022*, 3062541.
9. Wang, D.; Al-Rubaie, A.; Stincic, S.; Davies, J.; Aljasmi, A. Data-Driven Anomaly Detection Based on Multi-Sensor Data Fusion. In Proceedings of the 2021 International Conference on Smart Applications, Communications and Networking (SmartNets), Glasgow, UK, 22–24 September 2021; pp. 1–8. <https://doi.org/10.1109/SmartNets50376.2021.9555430>.
10. Tian, Z.; Zhuo, M.; Liu, L.; Chen, J.; Zhou, S. Anomaly detection using spatial and temporal information in multivariate time series. *Sci. Rep.* **2023**, *13*, 4400.
11. Zhang, Y.; Chen, Y.; Wang, J.; Pan, Z. Unsupervised Deep Anomaly Detection for Multi-Sensor Time-Series Signals. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 2118–2132. <https://doi.org/10.1109/TKDE.2021.3102110>.
12. Aliakbarisani, R.; Ghasemi, A.; Felix Wu, S. A data-driven metric learning-based scheme for unsupervised network anomaly detection. *Comput. Electr. Eng.* **2019**, *73*, 71–83. <https://doi.org/10.1016/j.compeleceng.2018.11.003>.
13. Zhuang, S.; Van Overbeke, P.; Vangeyte, J.; Sonck, B.; Demeyer, P. Evaluation of a Cost-Effective Ammonia Monitoring System for Continuous Real-Time Concentration Measurements in a Fattening Pig Barn. *Sensors* **2019**, *19*, 3669. <https://doi.org/10.3390/s19173669>.
14. Bhujel, A.; Arulmozhi, E.; Moon, B.E.; Kim, H.T. Deep-Learning-Based Automatic Monitoring of Pigs' Physico-Temporal Activities at Different Greenhouse Gas Concentrations. *Animals* **2021**, *11*, 89. <https://doi.org/10.3390/ani11113089>.

15. Darban, Z.Z.; Webb, G.I.; Pan, S.; Aggarwal, C.C.; Salehi, M. Deep learning for time series anomaly detection: A survey. *arXiv* **2022**, arXiv:2211.05244.
16. Norton, T.; Pessoa, J.; McAloon, C.; Garcia Manzanilla, E.; Boyle, L. Environmental risk factors influence the frequency of coughing episodes in finisher pigs: A case study on a farm free of respiratory disease. *Precis. Livest. Farming* **2022**, *22*, 522–529. Available online: [https://kuleuven.limo.libis.be/discovery/fulldisplay?docid=lirias3803805&context=SearchWebhook&vid=32KUL\\_KUL:Lirias&lang=en&search\\_scope=lirias\\_profile&adaptor=SearchWebhook&tab=LIRIAS&query=any,contains,LIRIAS3803805&offset=0%3F](https://kuleuven.limo.libis.be/discovery/fulldisplay?docid=lirias3803805&context=SearchWebhook&vid=32KUL_KUL:Lirias&lang=en&search_scope=lirias_profile&adaptor=SearchWebhook&tab=LIRIAS&query=any,contains,LIRIAS3803805&offset=0%3F) (accessed on 22 February 2024).
17. Benjamin, M.; Yik, S. Precision Livestock Farming in Swine Welfare: A Review for Swine Practitioners. *Animals* **2019**, *9*, 133. <https://doi.org/10.3390/ani9040133>.
18. Mujkic, E.; Philipsen, M.P.; Moeslund, T.B.; Christiansen, M.P.; Ravn, O. Anomaly detection for agricultural vehicles using autoencoders. *Sensors* **2022**, *22*, 3608.
19. Sharif, M.; Gupta, K.; Mohammed, M.A.; Jiwani, N. Anomaly detection in time series using deep learning. *Int. J. Eng. Appl. Sci. Technol.* **2022**, *7*, 296–305.
20. Ahmad, S.; Styp-Rekowski, K.; Nedelkoski, S.; Kao, O. Autoencoder-based condition monitoring and anomaly detection method for rotating machines. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Busan, Republic of Korea, 19–22 February 2020; pp. 4093–4102.
21. Adkisson, M.; Kimmell, J.C.; Gupta, M.; Abdelsalam, M. Autoencoder-based Anomaly Detection in Smart Farming Ecosystem. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), Virtually, 15–18 December 2021; pp. 3390–3399. <https://doi.org/10.1109/BigData52589.2021.9671613>.
22. Radaideh, M.I.; Pappas, C.; Walden, J.; Lu, D.; Vidyaratne, L.; Britton, T.; Rajput, K.; Schram, M.; Cousineau, S. Time series anomaly detection in power electronics signals with recurrent and ConvLSTM autoencoders. *Digit. Signal Process.* **2022**, *130*, 103704. <https://doi.org/https://doi.org/10.1016/j.dsp.2022.103704>.
23. Jeong, K.J.; Shin, Y.M. Time-series anomaly detection with implicit neural representation. *arXiv* **2022**, arXiv:2201.11950.
24. Wong, L.; Liu, D.; Berti-Equille, L.; Alnegheimish, S.; Veeramachaneni, K. AER: Auto-Encoder with Regression for Time Series Anomaly Detection. In Proceedings of the 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 17–20 December 2022; pp. 1152–1161. <https://doi.org/10.1109/BigData55660.2022.10020857>.
25. Krichen, M. Anomalies Detection Through Smartphone Sensors: A Review. *IEEE Sens. J.* **2021**, *21*, 7207–7217. <https://doi.org/10.1109/JSEN.2021.3051931>.
26. Homayouni, H.; Ghosh, S.; Ray, I.; Gondalia, S.; Duggan, J.; Kahn, M.G. An Autocorrelation-based LSTM-Autoencoder for Anomaly Detection on Time-Series Data. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Busan, Republic of Korea, 19–22 February 2020; pp. 5068–5077. <https://doi.org/10.1109/BigData50022.2020.9378192>.
27. Bloemheuvel, S.; van den Hoogen, J.; Jozinović, D.; Michelini, A.; Atzmueller, M. Graph neural networks for multivariate time series regression with application to seismic data. *Int. J. Data Sci. Anal.* **2022**, *16*, 317–332.
28. Yu, Y.; Zhu, Y.; Li, S.; Wan, D. Time series outlier detection based on sliding window prediction. *Math. Probl. Eng.* **2014**, *2014*, 879736.
29. Yu, K.; Shi, W.; Santoro, N.; Ma, X. Real-time Outlier Detection Over Streaming Data. In Proceedings of the 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Leicester, UK, 19–23 August 2019; pp. 125–132. <https://doi.org/10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00063>.
30. Guo, Y.; Liao, W.; Wang, Q.; Yu, L.; Ji, T.; Li, P. Multidimensional Time Series Anomaly Detection: A GRU-based Gaussian Mixture Variational Autoencoder Approach. In *Machine Learning Research, Proceedings of the 10th Asian Conference on Machine Learning, Beijing, China, 14–16 November 2018*; Volume 95, pp. 97–112.
31. Yang, G.; Yao, J. Multilayer neurocontrol of high-order uncertain nonlinear systems with active disturbance rejection. *Int. J. Robust Nonlinear Control.* **2024**, *34*, 2972–2987.
32. Benameur, R.; Dahane, A.; Kechar, B.; Benyamina, A.E.H. An Innovative Smart and Sustainable Low-Cost Irrigation System for Anomaly Detection Using Deep Learning. *Sensors* **2024**, *24*, 1162.
33. Maleki, S.; Maleki, S.; Jennings, N.R. Unsupervised anomaly detection with LSTM autoencoders using statistical data-filtering. *Appl. Soft Comput.* **2021**, *108*, 107443. <https://doi.org/https://doi.org/10.1016/j.asoc.2021.107443>.
34. Qiao, H.; Wang, T.; Wang, P.; Qiao, S.; Zhang, L. A time-distributed spatiotemporal feature learning method for machine health monitoring with multi-sensor time series. *Sensors* **2018**, *18*, 2932.
35. Pang, G.; Shen, C.; Cao, L.; Hengel, A.V.D. Deep Learning for Anomaly Detection: A Review. *ACM Comput. Surv.* **2021**, *54*, 38. <https://doi.org/10.1145/3439950>.
36. Zhang, J.E.; Wu, D.; Boulet, B. Time Series Anomaly Detection for Smart Grids: A Survey. In Proceedings of the 2021 IEEE Electrical Power and Energy Conference (EPEC), Wuhan, China, 28–30 May 2021; pp. 125–130. <https://doi.org/10.1109/EPEC52095.2021.9621752>.
37. Cai, J.; Fan, J. Perturbation Learning Based Anomaly Detection. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 14317–14330.
38. Ho, J.; Jain, A.; Abbeel, P. Denoising Diffusion Probabilistic Models. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6840–6851.

39. Mishra, S.; Kshirsagar, V.; Dwivedula, R.; Hota, C. Attention-Based Bi-LSTM for Anomaly Detection on Time-Series Data. In *Artificial Neural Networks and Machine Learning*, Proceedings of the ICANN 2021, Bratislava, Slovakia, September 14–17 2021; pp. 129–140.
40. Kaiser, L.; Gomez, A.N.; Shazeer, N.; Vaswani, A.; Parmar, N.; Jones, L.; Uszkoreit, J. One Model To Learn Them All. *arXiv* **2017**, arXiv:1706.05137.
41. Baan, J.; ter Hoeve, M.; van der Wees, M.; Schuth, A.; de Rijke, M. Understanding Multi-Head Attention in Abstractive Summarization. *arXiv* **2019**, arXiv:1911.03898.
42. Abbasimehr, H.; Paki, R. Improving time series forecasting using LSTM and attention models. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *13*, 673–691.
43. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In *Proceedings of the NIPS 2017: The 31st Annual Conference on Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, 4–9 December 2017.
44. Gunasegaram, D.; Barnard, A.; Matthews, M.; Jared, B.; Andreaco, A.; Bartsch, K.; Murphy, A. Machine learning-assisted in-situ adaptive strategies for the control of defects and anomalies in metal additive manufacturing. *Addit. Manuf.* **2024**, *81*, 104013.
45. Li, T.; Comer, M.; Delp, E.; Desai, S.R.; Mathieson, J.L.; Foster, R.H.; Chan, M.W. A Stacked Predictor and Dynamic Thresholding Algorithm for Anomaly Detection in Spacecraft. In *Proceedings of the MILCOM 2019—2019 IEEE Military Communications Conference (MILCOM)*, Norfolk, VA, USA, 12–14 November 2019; pp. 165–170. <https://doi.org/10.1109/MILCOM47813.2019.9021055>.
46. Alam, W.; Tahir, M.; Hussain, S.; Gul, S.; Hayat, M.; Irshad, R.R.; Pallonetto, F. Unveiling the Potential Pattern Representation of RNA 5-Methyluridine Modification Sites through a Novel Feature Fusion Model Leveraging Convolutional Neural Network and Tetranucleotide Composition. *IEEE Access* **2024**, *12*, 10023–10035.
47. Irshad, R.R.; Hussain, S.; Hussain, I.; Alattab, A.A.; Yousif, A.; Alsaiani, O.A.S.; Ibrahim, E.I.I. A novel artificial spider monkey based random forest hybrid framework for monitoring and predictive diagnoses of patients healthcare. *IEEE Access* **2023**, *11*, 77880–77894.
48. Abimannan, S.; El-Alfy, E.S.M.; Chang, Y.S.; Hussain, S.; Shukla, S.; Satheesh, D. Ensemble multifeatured deep learning models and applications: A survey. *IEEE Access* **2023**, *11*, 107194–107217.
49. Abimannan, S.; El-Alfy, E.S.M.; Hussain, S.; Chang, Y.S.; Shukla, S.; Satheesh, D.; Breslin, J.G. Towards Federated Learning and Multi-Access Edge Computing for Air Quality Monitoring: Literature Review and Assessment. *Sustainability* **2023**, *15*, 13951.

**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.