

Article

Not peer-reviewed version

---

# Integrating Temporal Kolmogorov-Arnold Networks with Autoencoders for Enhanced Pipeline Leak Detection: A Novel Unsupervised Learning Detector

---

Hengyu Wu , [Zhu Jiang](#) <sup>\*</sup> , [Xiang Zhang](#) <sup>\*</sup> , Jian Cheng

Posted Date: 11 December 2024

doi: 10.20944/preprints202412.0977.v1

Keywords: Pipeline leak detection; Temporal Kolmogorov-Arnold Network (TKAN); Autoencoder (AE); Unsupervised learning; Time-series anomaly detection



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

## Article

# Integrating Temporal Kolmogorov-Arnold Networks with Autoencoders for Enhanced Pipeline Leak Detection: A Novel Unsupervised Learning Detector

Hengyu Wu <sup>1</sup>, Zhu Jiang <sup>1,2,\*</sup>, Xiang Zhang <sup>1,2,\*</sup> and Jian Cheng <sup>1</sup>

<sup>1</sup> College of Energy and Power Engineering, Xihua University, Chengdu 610039, China; 212023080700018@stu.xhu.edu.cn (H.W.); cj@stu.xhu.edu.cn (J.C)

<sup>2</sup> Key Laboratory of Fluid and Power Machinery (Xihua University), Ministry of Education, Chengdu, 610039, China

\* Correspondence: 0120110033@mail.xhu.edu.cn (Z.J.); zhangxiang@mail.xhu.edu.cn (X.Z.)

**Abstract:** Artificial Intelligence (AI) technologies have been widely applied to the automated detection of pipeline leaks. However, traditional AI methods still face significant challenges in effectively detecting the complete leak process. Furthermore, the deployment cost of such models has increased substantially due to the use of GPU-trained neural networks in recent years. In this study, we propose a novel leak detector, which includes a new model and a sequence labeling method that integrates priori knowledge with traditional reconstruction error theory. The proposed model combines the Kolmogorov-Arnold Network (KAN) with an Autoencoder (AE): First, a Temporal Kolmogorov-Arnold Network (TKAN) is employed to capture the complex temporal dependencies in the data, and then an Autoencoder is used to compress and reconstruct the input data, ultimately calculating the reconstruction loss. To improve leak detection, we developed a novel unsupervised anomaly sequence labeling method based on traditional reconstruction error theory, which incorporates an in-depth analysis of the reconstruction error curve along with prior knowledge. This method significantly enhances the interpretability and accuracy of the detection process. Field experiments were conducted on real urban water supply pipelines, and a benchmark dataset was established to evaluate the proposed model and method against commonly used models and method. Experimental results demonstrate that the proposed model and method achieved a high Segment-wise Precision of 93.1%. Overall, this study presents a transparent and robust solution for automated pipeline leak detection, facilitating the large-scale, cost-effective development of digital twin systems for urban pipeline leak emergency management.

**Keywords:** Pipeline leak detection; Temporal Kolmogorov-Arnold Network (TKAN); Autoencoder (AE); Unsupervised learning; Time-series anomaly detection

## 1. Introduction

China has abundant water resources as a whole, but the per capita water resources are only a quarter of the global average, and their distribution is extremely uneven [1]. Due to long operating years, installation issues, and environmental factors, water supply pipeline leaks frequently occur, posing significant challenges to the efficiency and safety of urban water supply systems. The average leakage rate stands at 13.4%, substantially higher than international standards, as detailed in the "China Urban and Rural Construction Statistical Yearbook" [2,3]. This not only results in water wastage but also raises the risk of secondary pollution. Therefore, enhancing the maintenance and management of water supply systems to reduce leakage is crucial for ensuring the security and sustainable use of water resources.

With the continuous deepening of research work on this issue, scholars have achieved a large number of results. The methods based on fluid dynamics models and finite element simulation are developed by using advanced mathematical models to reveal complex mechanics principles, such as model-based residual generation with friction identification approaches [4], CFD-FEA based approaches[5], Kalman filter approaches[6], Lateral Extraction Hydraulic Gradient approaches[7–10],

Fisher Discriminant Analysis approaches [11], and so on. The effectiveness of these leakage detection methods across various pipeline networks have also been broadly recognized. In addition, data-driven approaches provide a valuable alternative that functions like a "black box" and can effectively capture and learn the intricate relationships between fluid flow signals in pipelines without the need for a detailed understanding of underlying physical principles and mechanisms.

Artificial intelligence (AI) technology, recognized for its efficiency and reliability, has been widely applied across various fields [12], and its utility in the domain of water supply pipeline leak detection has demonstrated promising outcomes. Notably, Quinones-Grueiro et al. [13] illustrated the feasibility of Principal Component Analysis (PCA) as an unsupervised learning technique for this purpose, highlighting its practical application in identifying anomalies. In the same year, Quinones-Grueiro et al. [14] conducted a comparative study of different supervised learning methods, concluding that Support Vector Machines (SVM) exhibit superior performance and robustness in leak detection tasks. Yingchun Xie et al. [15] developed a convolutional neural network (CNN) model based on time-frequency distribution maps. Zhang Peng et al. [16] has proposed a new method for water pipeline leak detection based on a Pseudo-Siamese Convolutional Neural Network (PCNN). Additionally, autoencoders are a type of neural network that can be trained on unlabeled data and distinguishing attributes that may deviate from the normal state. This feature is very useful for pipeline fault detection [17,18].

For pipeline leak detection tasks, negative pressure waves have been proven to be the most effective and widely used method [19]. Because the negative pressure wave signals used in pipeline leak detection are a type of time series signal, Recurrent Neural Networks (RNNs) offer significant advantages compared to other models [20,21]. And Long Short-Term Memory (LSTM) has been widely applied in pipeline leak detection tasks. Its unique gating mechanism effectively captures both long-term and short-term dependencies in time series data, significantly enhancing the model's ability to predict and understand dynamic changes in sequences [22]. Kim et al. [23] proposed a method LSTM-RNN which can improve the accuracy and efficiency of leak source detection. Lei Yang et al. [24] proposed a method combining OPELM and Bidirectional LSTM, which can accurately detect leak events and significantly reduce the number of false alarms compared to existing methods. However, existing deep learning models have undergone a debatable development process, characterized by ever-increasing model sizes, deeper layers, and more network parameters. Nowadays, training a model often requires hundreds or even thousands of GPU days, significantly increasing the cost of AI usage and reducing its capability for widespread application in everyday life [25]. Recently, Ziming Liu et al. [26] proposed a groundbreaking new neural network architecture—the Kolmogorov-Arnold Network (KAN). Due to its high accuracy, strong interpretability, and fewer model parameters, KAN has caused an unprecedented sensation and is considered a potential alternative to the cornerstone of modern deep learning, Multilayer Perceptrons (MLPs). To address the limitations of traditional models in handling complex sequential patterns, Rémi Genet et al. [27] proposed the Temporal Kolmogorov-Arnold Networks (TKAN)—an innovative neural network architecture that integrates LSTM with the KAN network. This innovation enables more accurate and efficient multi-step time series predictions. The TKAN architecture has shown significant potential in fields requiring multi-step forecasting. However, to the best of the authors knowledge, the application of TKAN to real-time automated pipeline leakage detection has not been explored yet.

This study aims to propose an unsupervised automated pipeline leakage detector by integrating the Temporal Kolmogorov-Arnold Network (TKAN) with an Autoencoder (AE) to develop a hybrid model called TKAN-Autoencoder (TKAN-AE). Additionally, a novel anomaly sequence labeling method based on reconstruction error, combined with prior knowledge, is proposed to enhance the reliability and accuracy of automated leakage detection. On-site experiments were conducted in real pipelines in megacities such as Shanghai, with simulated leaks to build a benchmark dataset. The detector was trained on unlabeled data and compared against two widely used baseline time-series models as well as classical threshold-based labeling methods. The major contributions and innovations of this study are as follows:

(1) This paper presents a novel hybrid model called TKAN-AE, where the Temporal Kolmogorov-Arnold Networks (TKAN) is employed to capture complex temporal dependencies within the data, while the Autoencoder (AE) is used for compressing and reconstructing the input data. The model demonstrates a well-balanced performance in pipeline leak detection tasks, achieving an F1 score of 75%.

(2) Building on the traditional reconstruction error theory, we incorporate prior knowledge to deeply analyze the reconstruction error curve and the negative pressure wave characteristics of pipeline leaks. We propose a new anomaly sequence labeling method. Compared to traditional methods, this approach improves segment-level precision by 54.2% and has been experimentally validated to be applicable to real-world field datasets and various models, demonstrating strong robustness and generalizability.

(3) To the best of our knowledge, this is the first time that the lightweight TKAN network has been applied to real-world field datasets for urban water pipeline leak detection. This work provides a feasible solution for the future industrial-scale application of artificial intelligence in pipeline leak detection in real-world scenarios. This work provides a feasible solution for the future industrial-scale application of artificial intelligence in pipeline leak detection.

## 2. The Proposed Leak Detection Method

The whole structure of the proposed leak detector in this paper is illustrated in **Figure 1**, including data preprocessing, construction of TKAN-AE, and anomaly labeling. Initially, the negative pressure wave data, which is obtained from real city pipelines, undergoes preprocessing steps like basic noise reduction, applying sliding window averages, and normalization. This results in turning the original data into subsequence datasets. Next, the subsequences are compressed by the encoder into latent variables and then reconstructed by the decoder. Afterward, the reconstructed data is compared with the original dataset, and reconstruction loss for each individual point are measured. Next, we plot the reconstruction error curve and apply our proposed anomaly sequence labeling method to detect anomalies.

### 2.1. Data Preprocessing

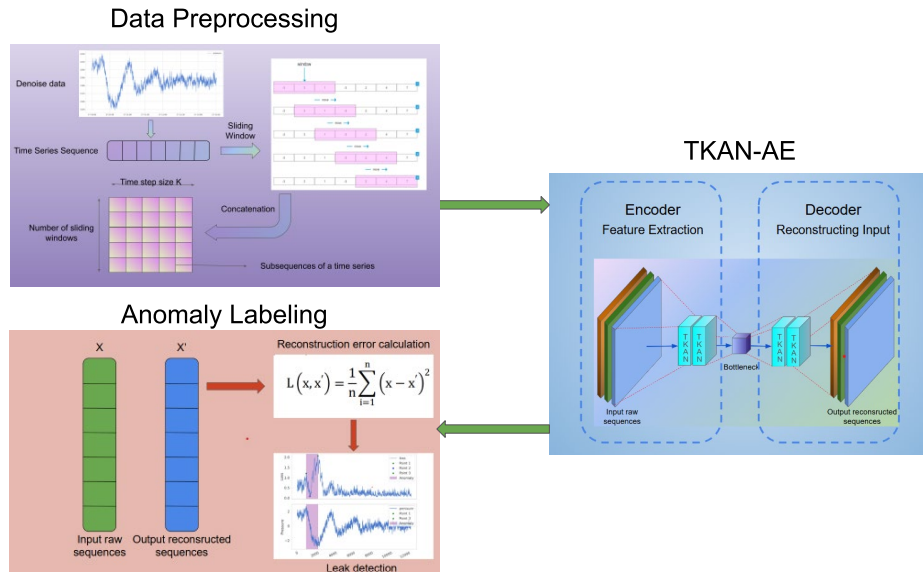
Data preprocessing including cleaning and denoising is considered one of the important parts in a machine learning process. It has been shown by Y Zhang et al. [28], that directly using data which is noisy or not clean for training neural networks may not be an ideal practice. Proper preprocessing can improve the training effectiveness and predictive performance of the model. The first step involves using the Sliding Window Average to further reduce random fluctuations in the signal Shtayat et al. [29], thereby enhancing its smoothness. The Sliding Window Average method can express as:

$$MA_t = \frac{1}{n} \sum_{i=t-n+1}^t x_i \quad (1)$$

where  $MA_t$  is the moving average at time  $t$ ,  $x_i$  is the value of the time series at time  $i$ ,  $n$  is the window size (number of periods to average over).

Next, these subsequences from the time series go through sliding window reshaping, which results in a matrix. This matrix then becomes the final form of input data for the model.





**Figure 1.** The overall structure of the detector.

After that, the second step is applying MinMax scaling normalization on the data. Because each data point may belong to a different pressure range, it becomes important to make sure that all training data remain within the same scale [30]. This is necessary to avoid having specific features with larger values affect the anomaly detection model more than others because of their higher magnitudes[31].

## 2.2. TKAN-AE

### 2.2.1. Autoencoder

Autoencoders perform anomaly detection by learning effective representations of data. Their encoder-decoder architecture adapts well to capturing the underlying structure of data and identifying anomalies. They are a type of neural network used for efficiently reconstructing unlabeled data, learning representations of a given dataset during training to filter out irrelevant aspects like noise [32,33]. In anomaly detection, the model learns patterns of normal processes, classifying any data that deviates from this pattern as anomalous [34].

Autoencoders consist of two main parts, as **Figure 2** illustrated:

The first part is the encoder, which maps the input to a latent representation space  $h$ , and the decoder maps information from this latent space back to a reconstructed input.

In the simplest architecture with a single hidden layer, the encoder of an autoencoder takes input  $x \in \mathbb{R}^d$  and maps it to the latent space  $h \in \mathbb{R}^p$ . Typically, the latent space  $h$  is represented as:

$$h = \sigma(Wx + b) \quad (2)$$

where  $\sigma$  is an activation function like sigmoid or ReLU,  $W$  is a weight matrix, and  $b$  is a bias vector, usually initialized randomly and updated gradually during training.

Next, the decoder takes the latent representation  $h$  and attempts to reconstruct the input of the encoder. In other words, the decoder aims to map the latent representation  $h$  back to a reconstructed input  $x'$ . Following previous notation conventions, this operation can be expressed as:

$$x' = \sigma'(W'h + b') \quad (3)$$

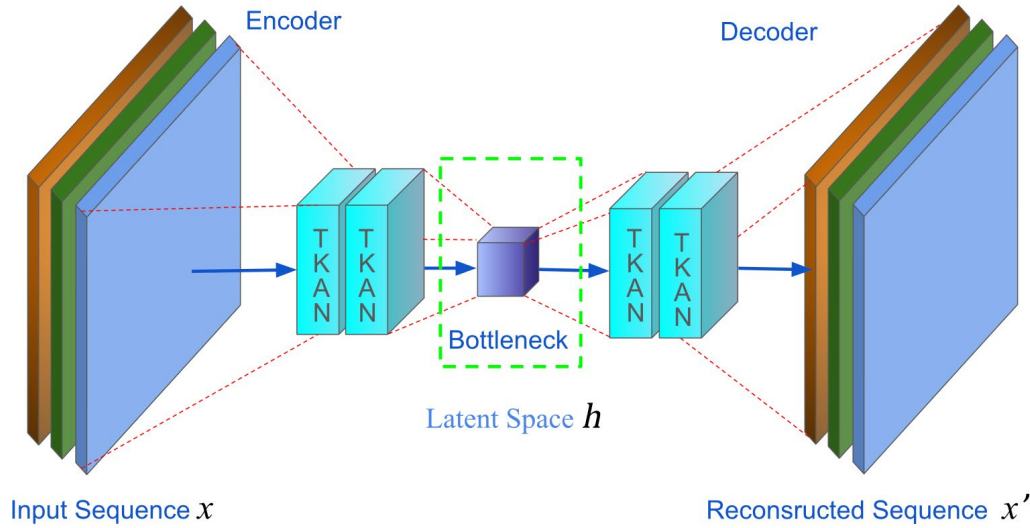
where  $\sigma'$  is also the activation function, and  $W'$  and  $b'$  are the weight matrix and bias vector specific to the decoder stage.

Ultimately, during the training process, the autoencoder minimizes the loss function, a process known as reconstruction error. Typically, the Mean Absolute Error (MAE) exhibits strong robustness

against outliers and can adapt to nonlinear data containing noise [35]. Therefore, we choose Mean Absolute Error (MAE) as our loss function, which is represented as follows:

$$L(x, x') = \frac{1}{n} \sum_{i=1}^n (x - x')^2 = \frac{1}{n} \sum_{i=1}^n \left( x - \sigma' \left( W' (\sigma(Wx + b)) \right) + b' \right)^2 \quad (4)$$

where  $L$  is used to measure the difference between the variables  $x$  and  $x'$ , which corresponds to the difference between the original input signal and the reconstructed output signal.



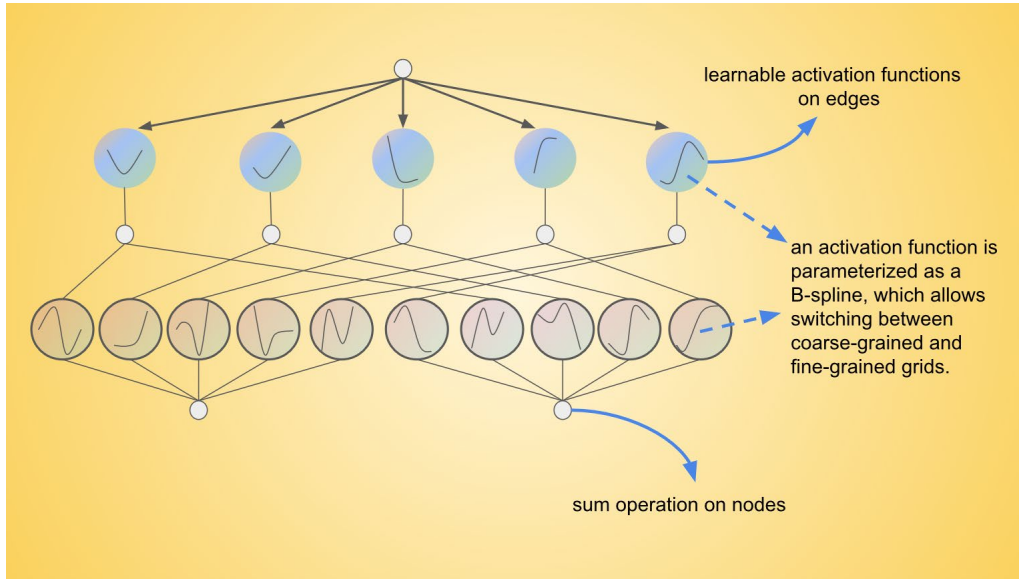
**Figure 2.** The structure of Autoencoder.

### 2.2.2. TKAN

TKAN is an improved version of the Kolmogorov-Arnold network architecture for time series, combining the classic looping and gating mechanisms of LSTM. This new architecture addresses the common long-term dependency issues found in RNNs. Compared to traditional models such as LSTM and GRU, TKAN excels in long-term prediction, demonstrating the ability to handle various situations and longer time periods [27].

The Kolmogorov-Arnold Network (KAN)'s structure is shown in **Figure 3**. In the presented diagram, the activation functions are not located at the nodes but are instead assigned to the connecting edges, with each edge associated with a learnable activation function parameterized as a B-spline. This configuration allows the activation functions to adapt dynamically between coarse-grained and fine-grained resolutions, unlike traditional Multi-Layer Perceptrons (MLPs), which employ fixed activation functions at each node. Kolmogorov-Arnold Networks (KANs) utilize these learnable B-spline functions along the edges. The signals transmitted between nodes are aggregated using a simple summation operation, meaning that the nodes themselves only perform summation without involving activation functions. This design results in faster computational speed and improved interpretability, as the complexity of the activation functions resides in the edges rather than the nodes. The KAN network is inspired by the Kolmogorov-Arnold representation theorem [35]. The Kolmogorov-Arnold representation theorem (Equation 5) states that any multivariate continuous function can be represented as a combination of univariate functions and addition operations,

$$f(x_1, \dots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left( \sum_{p=1}^n \phi_{q,p}(x_p) \right) \quad (5)$$



**Figure 3.** The detailed structure of the KAN network.

where  $\phi_{q,p}$  are univariate functions that map each input variable ( $x_p$ ), such  $\phi_{q,p}: [0,1] \rightarrow \mathbb{R}$  and  $\phi_q: \mathbb{R} \rightarrow \mathbb{R}$ . Since all the functions to be learned are univariate, Ziming Liu et al.[26] subsequently parameterize each one-dimensional function as a B-spline curve, employing learnable coefficients for the local B-spline basis functions. So, a KAN layer is

$$\Phi = \{\phi_{q,p}\}, \quad p = 1, 2, \dots, n_{in}, \quad q = 1, 2, \dots, n_{out}, \quad (6)$$

where  $\phi_{q,p}$  are parametrized function of learnable parameters and  $n_{in}$  means the number of inputs while  $n_{out}$  means the number of outputs. A shape of KAN is defined as:

$$[n_0, n_1, \dots, n_L], \quad (7)$$

where  $n_L$  is the number of nodes in the  $i^{\text{th}}$  layer of the computational graph. They denote the  $i^{\text{th}}$  layer by  $(l, i)$ , and the activation value of the  $(l, i)$  neuron by  $x_{l,i}$ . Between layer  $l$  and  $l+1$ , there are  $n_l n_{l+1}$  activation functions: the activation function that connects  $(l, i)$  and  $(l+1, j)$  is denoted by

$$\phi_{l,j,i}, \quad l = 0, \dots, L-1, \quad i = 1, \dots, n_l, \quad j = 1, \dots, n_{l+1}. \quad (8)$$

The activation value of the  $(l+1, j)$  neuron is simply the sum of all incoming post-activations, it can be written in matrix form:

$$\mathbf{x}_{l+1} = \begin{pmatrix} \phi_{l,1,1}(\cdot) & \phi_{l,1,2}(\cdot) & \cdots & \phi_{l,1,n_l}(\cdot) \\ \phi_{l,2,1}(\cdot) & \phi_{l,2,2}(\cdot) & \cdots & \phi_{l,2,n_l}(\cdot) \\ \vdots & \vdots & & \vdots \\ \phi_{l,n_{l+1},1}(\cdot) & \phi_{l,n_{l+1},2}(\cdot) & \cdots & \phi_{l,n_{l+1},n_l}(\cdot) \end{pmatrix} \mathbf{x}_l, \quad (9)$$

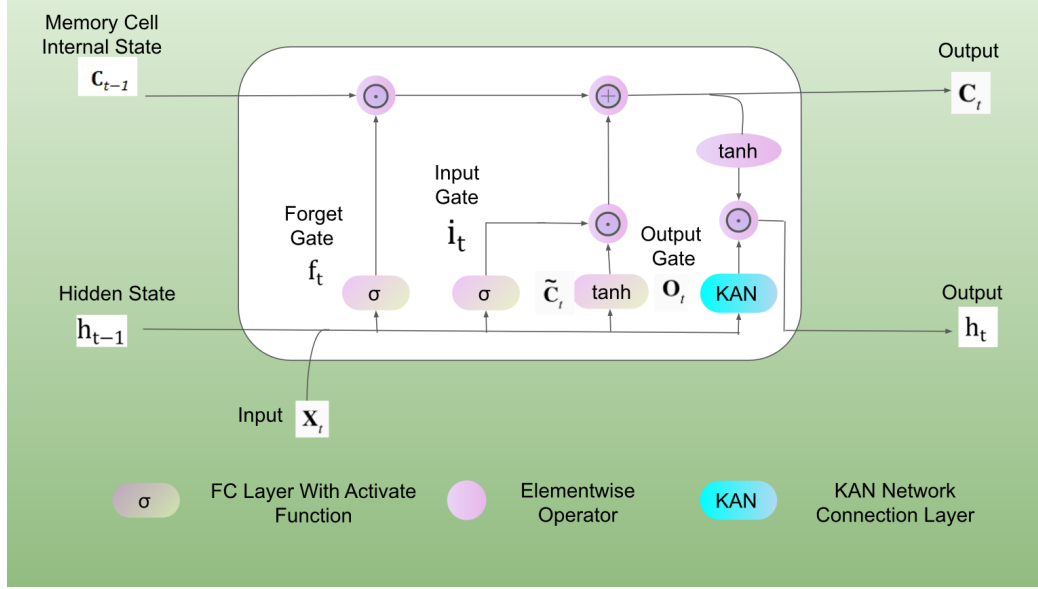
where  $\Phi_l$  is the function matrix corresponding to the  $l^{\text{th}}$  KAN layer and the notation  $(\cdot)$  represents a placeholder for the input variable of a function, indicating that this position can accept a variable as input. A general KAN network is a composition of  $L$  layers: given an input vector  $\mathbf{x}_0 \in \mathbb{R}^{n_0}$ , the output of KAN is

$$\text{KAN}(\mathbf{x}) = (\Phi_{L-1} \circ \Phi_{L-2} \circ \cdots \circ \Phi_1 \circ \Phi_0) \mathbf{x}. \quad (10)$$

where  $\Phi_1 \circ \Phi_0 = \Phi_1(\Phi_0(x))$ , represents the composition of functions.

Temporal Kolmogorov-Arnold Networks (TKAN) adapt the concept of Kolmogorov-Arnold Networks (KAN) to time series by integrating temporal management within neural networks. As shown in **Figure 4**, this approach combines the KAN architecture with a modified Long Short-Term

Memory (LSTM) unit. The learnable activation functions in KAN capture complex nonlinear patterns in time series data, while LSTM cells, with their gating mechanisms and memory units, effectively address the challenges of long-term dependencies and the vanishing/exploding gradient problem common in traditional RNNs. This enables the model to retain information about past events over extended time horizons, enhancing its ability to model long-range temporal dependencies. Although the intricate architecture of LSTM may slightly increase computational costs, this trade-off is justified in specific learning tasks such as pipeline leak detection, where high accuracy is paramount.



**Figure 4.** The detailed structure of the TKAN network.

We denote the input vector of dimension  $d$  as  $x_t$ . This unit uses several internal vectors and gates to manage information flow. The forget gate, with activation vector  $f_t$ ,

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f), \quad (11)$$

decides what information to forget from the previous state. The input gate, with activation vector denoted  $i_t$ ,

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \quad (12)$$

controls which new information to include.

Where  $\sigma$  represents the activation function,  $x_t$  denotes the input vector at the current time step,  $h_{t-1}$  represents the hidden state vector from the previous time step,  $W$  is the weight matrix connecting the input vector,  $U$  is the weight matrix connecting the hidden state vector from the previous time step  $h_{t-1}$ , and  $b_i$  denotes the bias vector.

The KAN network now embeds memory management at each layer:

$$\text{KAN}(x, t) = (\Phi_{L-1,t} \circ \Phi_{L-2,t} \circ \dots \circ \Phi_{1,t} \circ \Phi_{0,t})(x, t). \quad (13)$$

The output gate, with activation vector  $o_t$ ,

$$o_t = \sigma(\text{KAN}(\vec{x}, t)), \quad (14)$$

determines what information from the current state to output given  $\text{KAN}(\vec{x}, t)$ . The hidden state,  $h_t$ , captures the unit's output, while the cell state,  $c_t$  is updated such

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (15)$$

Where  $\odot$  represents Element-wise multiplication (Hadamard product) and  $\tilde{c}_t = \sigma(W_c x_t + U_c h_{t-1} + b_c)$  represents its internal memory. All these internal states have a dimensionality of  $h$ . The output denoted  $h_t$  is given by

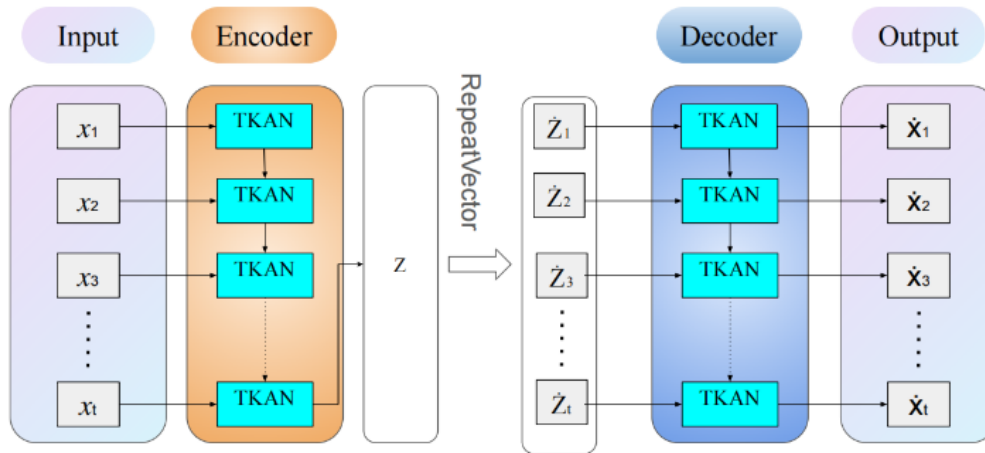


$$h_t = o_t \odot \tanh(c_t). \quad (16)$$

### 2.2.3. Proposed Anomaly Detection TKAN-AE Model

In this study, we focus on the negative pressure wave signals generated by pipeline leaks, which are time-series data. Therefore, we define this problem as a time-series anomaly detection task. Time-series anomaly detection aims to identify and detect abnormal patterns or behaviors in time-series data, which may indicate errors, unexpected events, or system failures[37].

To address the specific requirements of time-series anomaly detection, we developed a novel model named TKAN-AE (as illustrated in **Figure 5.**) The detailed structure of the proposed TKAN-AE network. This model integrates the Temporal Kolmogorov-Arnold Network (TKAN) within an autoencoder (AE) architecture, effectively capturing salient features of time-series data. The TKAN component combines the temporal gating mechanism of Long Short-Term Memory (LSTM) networks with the powerful function approximation capabilities of Kolmogorov-Arnold Networks (KAN), thus capturing both long-term and short-term dependencies inherent in time-series data. Meanwhile, the autoencoder's encoder-decoder structure facilitates the learning of efficient data representations, reconstructing the input to identify anomalies.



**Figure 5.** The detailed structure of the proposed TKAN-AE network.

In particular, during the encoding phase, the TKAN layer initially processes the input time series data. The primary objective of this layer is to maintain the structural coherence of the time series, thereby enabling in-depth analysis of temporal dependencies. The TKAN preserves the output of each timestep, allowing the model to effectively process temporal information in subsequent layers and capture long-term relationships across the sequence. Subsequent encoding layers aggregate the temporal information into a compact fixed-size feature vector, which serves as the basis for reconstructing the original time series during the decoding phase.

In the decoding phase, the TKAN layer's role is to restore the fixed-size feature vector, derived during encoding, back to the original time-series format. By expanding each element of the feature vector temporally, this layer reconstructs the output sequence, ensuring it matches the length of the original input. The reconstruction process occurs progressively, as each timestep is reconstructed using information encoded within the learned feature representation. By reconstructing each timestep with precision, the TKAN layer enables the model to effectively recover the complete time series from its compressed representation, which is critical for accurate reconstruction and the identification of anomalies.

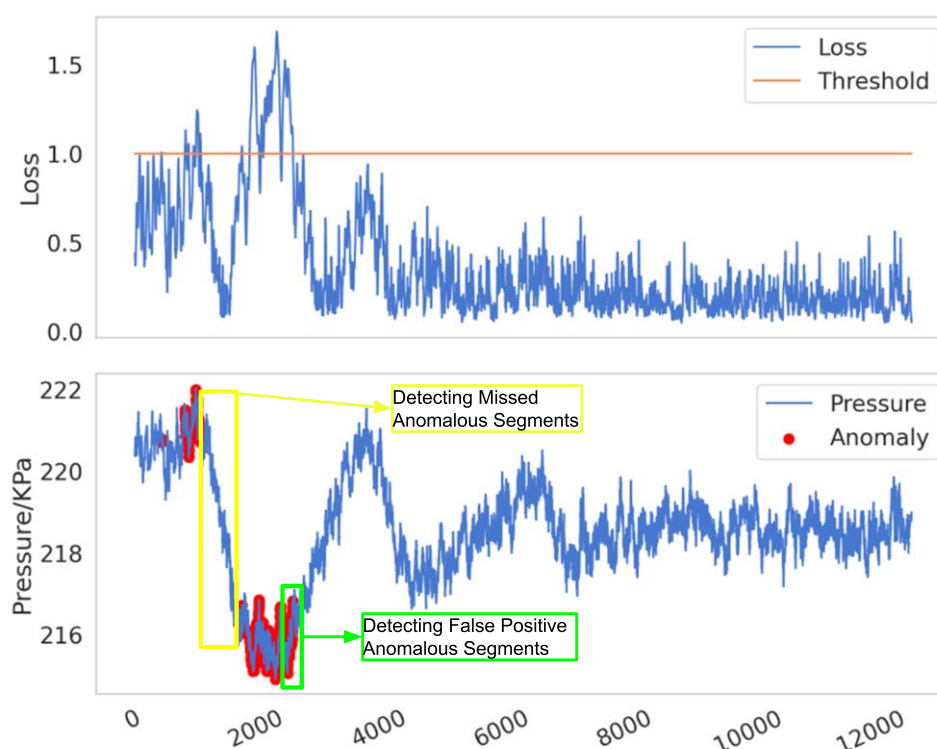
### 2.3. A Novel Anomaly Detection and Sequence Labeling Method for Pipeline Leak Detection

Reconstruction error is defined as the difference between the reconstructed data and the original data. In anomaly detection methods based on reconstruction error, it is generally assumed that the

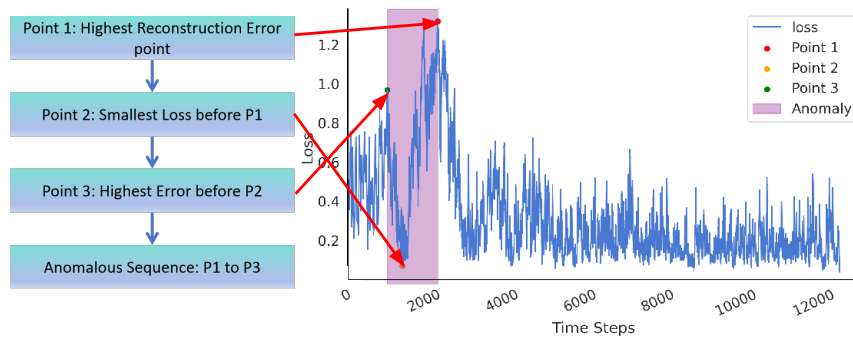
reconstruction error of normal data is small, while the reconstruction error of anomalous data is large. As illustrated in **Figure 6**, when a pipeline leak occurs, the reconstruction error (or loss) typically increases sharply in the initial phase, but may gradually decrease as the leak continues. This phenomenon arises because the model gradually learns the characteristics of the leakage process and attempts to reconstruct and predict it. As the negative pressure wave propagates through the pipeline system, the internal pressure of the pipeline rises significantly. For a model that has already learned the anomalous features, this sudden pressure increase leads to a sharp rise in the reconstruction error. However, in traditional threshold-based detection methods, data is classified as anomalous only when the reconstruction error exceeds a predefined threshold. Due to the decrease in reconstruction error during the leakage process, certain signals that should be identified as indicative of a leak may be overlooked in specific local regions, resulting in missed detections, as shown in the yellow box. Furthermore, the subsequent rise in pressure caused by the propagation of the negative pressure wave can lead to an increase in reconstruction error, thereby causing false positives, as shown in the green box. The issue of threshold selection has long been a challenge in anomaly detection. Research by Xue Yang et al. [38] has shown that fixed or static thresholds often fail to adapt to the dynamic characteristics of real-world data, resulting in false positives or missed detections.

To address this issue, we propose an improved approach that builds upon the traditional reconstruction error framework. By conducting a detailed analysis of the reconstruction error curve and incorporating prior knowledge along with the model's core principles, the goal is to enhance the performance of anomaly detection. The specific method is as follows:

As illustrated in **Figure 7**, The flowchart of anomaly detection and labeling methods for pipeline leaks., the specific detection procedure is as follows: we initially identify the point P1 corresponding to the maximum reconstruction error. Subsequently, within the sequence leading up to P1, we determine point P2, which represents the minimum loss value. Finally, we locate point P3, characterized by the maximum reconstruction error preceding P2. The sequence between points P1 and P3 is defined as the potentially anomalous region, which is highly indicative of leakage. The detailed experimental comparison process and results will be elaborated in the subsequent sections.



**Figure 6.** The issues in traditional threshold-based anomaly detection methods.

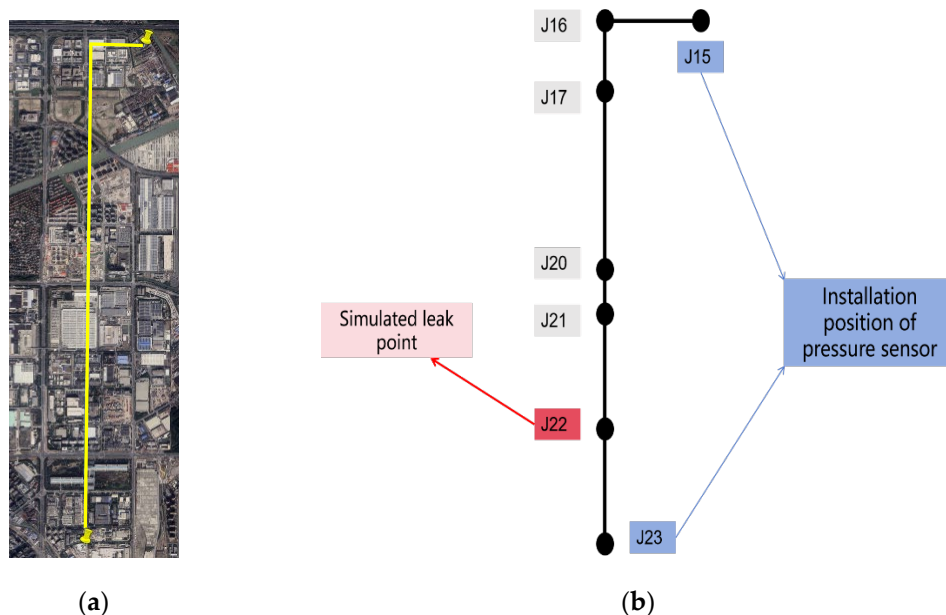


**Figure 7.** The flowchart of anomaly detection and labeling methods for pipeline leaks.

### 3. Benchmark Dataset

In this section, we constructed a benchmark dataset based on pressure signals collected from field experiments. Pressure signals, recognized as a convenient and stable indicator for pipeline leakage detection [33], are widely employed in leakage detection tasks. To ensure the practical relevance of our study, we deliberately avoided using simulated data generated by software such as Fluent or CFX, as well as data from laboratory-based pipeline systems. Instead, we utilized field leakage experiment data from actual urban water supply pipelines, as these data more accurately reflect real-world industrial applications. In contrast, simulated data often exhibit biases that are more aligned with specific experimental conditions, lacking the broader applicability to real-world usage scenarios.

The field experiments were conducted on water supply pipelines located underground at Shenjiang Road and Wuzhou Avenue in Pudong New District, Shanghai, China (The yellow line in **Figure 8a**). The total length of the pipeline involved in the experiment was 5,891 meters. High-frequency dynamic pressure sensors were installed at wells J15 and J23 (**Figure 8b**) to monitor pressure fluctuations. To simulate leak events, a valve located at well J22 was manipulated to induce leakage, with different valve openings representing varying leakage severities (**Figure 9a**) and the sensor installation diagram for the field site is shown in **Figure 9b**. Furthermore, the exact times for opening and closing the valves were recorded to provide reference data for subsequent model evaluation. The distance between J15 and J22 is approximately 4,981 meters, between J15 and J23 is about 5,891 meters, and between J22 and J23 is around 910 meters.



**Figure 8.** An overview of the pipeline from a satellite map and a topographic map perspective:(a) Overview Map of Pipelines (Real Urban Water Supply Pipelines in Shanghai); (b) Layout Map of Pipelines (Including Sensors and Simulated Leakage Locations).



**Figure 9.** On-site experiment illustrations: (a) Workers' operations with varying valve openings during leak simulation. (b) Pressure sensor installation.

To ensure realistic simulation of leakage scenarios and eliminate randomness, the experiments were conducted over multiple days and at different times of day, with sensor data continuously collected over an extended period, thereby ensuring data diversity and representativeness. An overview of the leakage data is provided in Table 1. Additionally, pressure signals under normal operating conditions were collected over several days to serve as reference data for healthy states. Based on these experiments, we found that the number of normal samples greatly exceeded the number of leakage samples. To address the potential imbalance in the dataset, which could affect model training, we ultimately selected 12 leakage samples and 200 normal samples to construct the dataset. This balanced selection of data aims to ensure that the model can effectively learn from the limited leakage data while preventing bias caused by the disproportionate number of normal samples. The raw pressure signals were processed using the data preprocessing methods outlined in Section 2.1. These signals were then segmented into subsequences of equal length. After extensive experimentation, we have selected a standard subsequence length of 13,000 data point to create our benchmark dataset. A shorter sequence would fail to capture the complete leak process, while a longer sequence might include pressure fluctuations associated with multiple states, which would reduce data consistency and hinder the learning process of the model. Moreover, for time series models, excessively long sequences significantly increase computational load and memory requirements. Thus, a sequence length of 13,000 data points is a suitable choice given our current computational capacity, striking a balance between computational efficiency and data representativeness. This dataset was subsequently used to train our model and to compare its performance with other benchmark models. An overview of the benchmark test dataset is presented in Table 2.

**Figure 10** provides an example of pressure signal data, illustrating that a sharp drop in pressure occurs when a pipeline experiences a leakage event. This phenomenon is primarily caused by the rapid outflow of fluid at the leakage point, resulting in a localized pressure reduction. Subsequently, due to the generation of a negative pressure wave and its propagation towards both ends of the pipeline, significant changes occur in the internal pressure dynamics of the pipeline. The propagation of the negative pressure wave induces fluid near the leakage point to move towards the low-pressure zone, leading to a transient pressure rise and forming a U-shaped waveform. Over time, as the negative pressure wave reflects and attenuates within the pipeline, the system eventually reaches a new steady-state pressure, which, in the short term, falls below the original operating pressure. This temporary reduction in steady-state pressure is due to the ongoing fluid loss at the leakage point.

**Table 1.** Properties of the Leakage Experiment Data.

Dataset properties	Value
Number of healthy samples	200

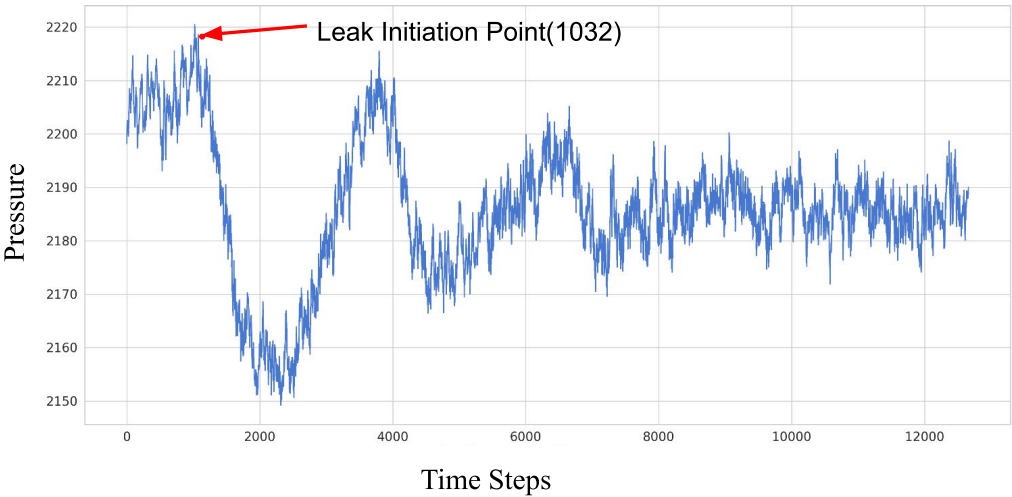
Number of samples with leakage	12
The signal length of a single sample	13,000-time steps
Total signal length	2,756,000-time steps
Sampling frequency (Hz)	500
Anomaly ratio	5.66%

<sup>1</sup>Total signal length=The signal length of a single sample\*total samples=13000\*212=2756000. <sup>2</sup>Anomaly Ratio: The proportion of samples labeled as anomaly points to the total number of samples in the dataset. This metric is used to quantify the rarity or prevalence of anomalies within the dataset. Anomaly Ratio=12/212=5.66%

**Table 2.** Benchmark dataset properties acquired from the experimental field tests.

Valve opening degree	Number of experiments	Simulated leakage area $S_L$ (dm <sup>2</sup> )	Pipe cross-sectional area $S_P$ (dm <sup>2</sup> )	$S_L/S_P$
30	3	1.05	1017.9	0.103%
45	3	1.57	1017.9	0.154%
60	3	2.36	1017.9	0.232%
90	3	3.14	1017.9	0.308%

<sup>1</sup>  $S_L / S_P$ : The primary function of this indicator is to reflect the proportion of the leakage area relative to the cross-sectional area of the pipe, quantifying the severity of the leakage.



**Figure 10.** A sample of pressure signal data.

4. Experiment Results and Analysis

4.1. Experiment Environment

Our model construction and experimental analyses were performed on Google Colaboratory (Colab). All experiments utilized Colab's Tesla T4 GPU, equipped with CUDA Version 12.2, and an Intel(R) Xeon(R) CPU @ 2.20GHz, paired with 12GB of RAM. The software version details are as follows:

- Python: 3.10.12



- TensorFlow: 2.15.0
- Keras: 2.15.0

Google Colaboratory is a cloud-based, interactive notebook environment developed by Google. It provides users with free computational resources, including CPUs, GPUs, and TPUs, allowing them to execute code directly in a web browser without requiring any local setup or installation. TensorFlow, an open-source machine learning framework, was used to facilitate the construction, training, evaluation, and saving of neural network models. Keras, integrated within TensorFlow, is a high-level neural network API that provides models, thereby simplifying the development of deep learning workflows in Python.

#### 4.2. Metrics For Model Performance

Several evaluation metrics are commonly used to assess the performance of classification machine learning models. For pipeline leakage time series anomaly detection, the following metrics are particularly suitable:

**Accuracy:** This is a fundamental metric that measures the proportion of correctly predicted instances out of the total number of instances. In general classification tasks, for the time series anomaly detection of pipeline leaks, a continuous leak anomaly segment is regarded as a whole (a positive sample). If the detector triggers an alarm at any point within the segment and identifies an anomaly, then the entire segment as a sample is considered successfully detected. However, evaluating the entirety of the sample does not adequately measure the performance of the detector, and subsequent work on leak localization heavily relies on the precision of detection. Therefore, we introduce a new metric—Segment-wise Precision[42,43].

**Segment-wise Precision:** This metric measures the proportion of actual anomaly points within a specific segment that has been detected as anomalous. In other words, after a sample is successfully detected as a whole, we continue to focus on how many points within it were correctly identified as anomalies.

**True Positives (TP):** The number of actual leakage instances correctly identified by the model as leaks.

**False Negatives (FN):** The number of actual leakage instances that the model failed to identify as leaks.

**False Positives (FP):** The number of non-leakage instances incorrectly classified by the model as leaks.

**True Negatives (TN):** The number of non-leakage instances correctly identified by the model as non-leaks.

These metrics are often visualized using a confusion matrix, which provides a comprehensive overview of the model's classification performance.

**Confusion Matrix:** A tool used to evaluate the performance of classification models by comparing the model's predictions to the actual outcomes. It includes four components: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). The confusion matrix helps to visualize the model's accuracy, false positive rate, and false negative rate, thus providing a holistic evaluation of its performance.

**Precision:** Precision measures the proportion of true positive predictions among all positive predictions made by the model. It reflects the reliability of the model's positive predictions.

**Recall:** Recall measures the proportion of actual positive instances that are correctly identified by the model. It reflects the model's ability to identify all relevant instances.

**F1-score:** The F1-score is the harmonic mean of Precision and Recall, providing a balanced evaluation of the model's classification performance, particularly in cases where there is an uneven class distribution.

These metrics are critical for a comprehensive assessment of pipeline leakage detection models in time series anomaly detection tasks. The mathematical expressions for these metrics are provided in Equations (17) to (21).

$$\text{Accuracy} = \frac{TP}{TP + FP + TN + FN} \quad (17)$$

$$\text{Segment - wise Precision} = \frac{|S_{\text{detected}} \cap S_{\text{true}}|}{|S_{\text{detected}}|} \quad (18)$$

$S_{\text{detected}}$  represents the set of detected anomalous segments, while  $S_{\text{true}}$  represents the set of actual anomalous segments.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (19)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (20)$$

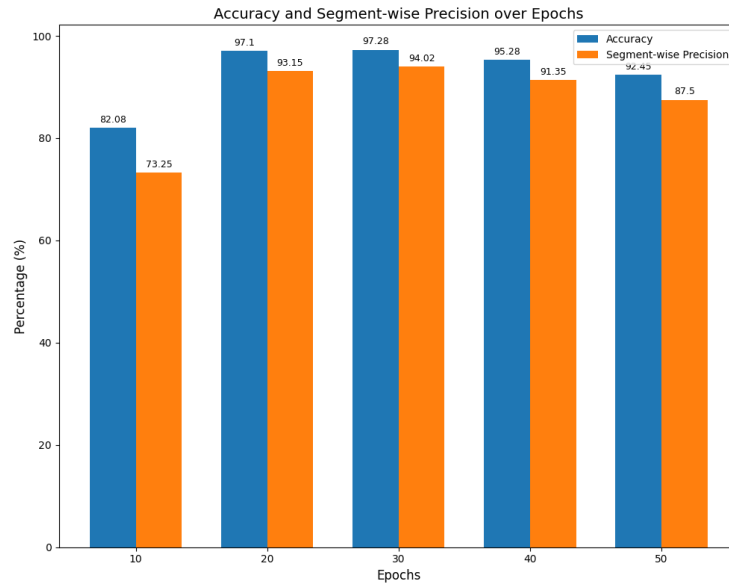
$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (21)$$

#### 4.3. Experiment Development

Building on the work of Ziming Liu et al. [26], which demonstrated that KAN networks effectively fit data using a minimal number of parameters and shallow network architectures, we applied similar principles to develop our TKAN-AE network. Both the encoder and decoder in our TKAN-AE network consist of a single TKAN layer, utilizing B-spline activation functions of orders ranging from 0 to 4. For benchmarking, we selected two widely used recurrent neural network (RNN) models for multistep prediction: the Gated Recurrent Unit (GRU) and the Long Short-Term Memory network (LSTM). We intentionally excluded the Transformer model from our comparison due to its extensive parameter requirements, primarily resulting from the multi-headed attention mechanism and numerous fully connected layers, which render it unsuitable for low-cost, large-scale deployment in industrial applications.

To ensure a fair comparison, we also constructed simplified autoencoder variants, namely LSTM-AE and GRU-AE, each using a single-layer LSTM or GRU architecture for both the encoder and decoder components. All models were trained using the Adam optimizer, an adaptive moment estimation method introduced by Kingma et al. [34], which combines the benefits of momentum and RMSProp algorithms. The loss function for all models was the Mean Absolute Error (MAE). To mitigate overfitting, dropout regularization was applied with a probability of 0.2, alongside a weight decay of 0.2, following recommendations by Kumar and Asri et al. [38,39]. Furthermore, to further compare the proposed anomalous sequence labeling method with traditional threshold-based approaches, we meticulously trained the traditional threshold method and, after multiple trials, selected a threshold value of 1 to ensure the highest possible accuracy.

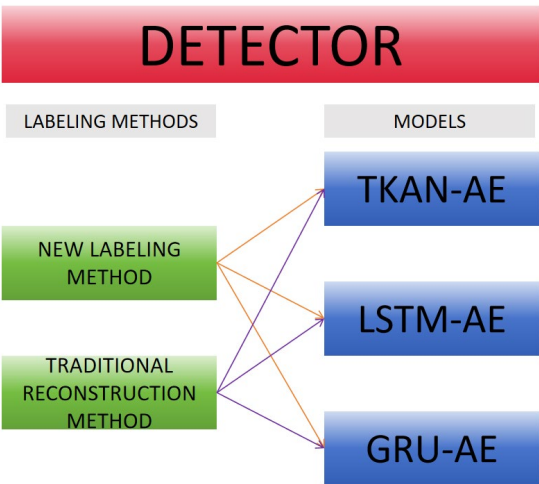
In line with the findings of Aburass et al. [42], we recognize that an excessive increase in the number of training epochs does not necessarily improve detection accuracy and may instead lead to higher computational costs. Their research highlights diminishing returns in performance as the number of epochs increases, particularly for models designed to detect anomalies in time series data, rather than solely focusing on model fitting. Therefore, we tested multiple training epoch configurations to determine the optimal number of epochs. As shown in **Figure 11**, the best performance was achieved at 30 epochs, but the improvement over 20 epochs was marginal, while computational costs increased by 33%. Based on this, we selected 20 epochs as the standard training epoch count for all subsequent comparative experiments.



**Figure 11.** Accuracy and Segment-wise Precision over Epochs.

#### 4.4. Results And Comparison

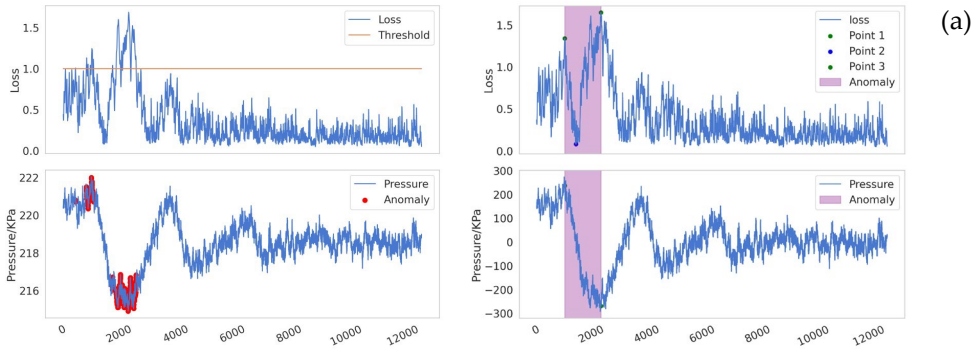
In this experiment, we compare the performance of six detectors, resulting from the combination of three models (TKAN-AE, LSTM-AE, GRU-AE) with two different anomalous sequence labeling methods (Figure 12). We have provided links to publicly available runnable code for comparative example experiments in the Supplementary Materials, allowing others to conduct reproducible experiments and further develop in-depth research based on our work. This comparison not only evaluates the performance of each model but also verifies the superiority and robustness of the proposed labeling approach. Figure 12 presents specific detection instances, where subfigures (a), (b), and (c) respectively illustrate the visual comparison between our proposed method and traditional methods for the three models. Through the study of specific leakage detection tasks, we observed that for the same model, the differences brought by different labeling methods are quite significant. The traditional labeling method based on reconstruction error thresholds tends to miss a substantial number of intermediate leakage segments, and it also incorrectly labels a short sequence as anomalous after the leakage has ended, leading to notable false negatives and false positives (Figure 13). In contrast, our proposed new labeling method effectively avoids these issues, significantly improving labeling accuracy. Furthermore, under the same labeling method, the detection capabilities of the three models are generally consistent. Figure 14 further provides a detailed comparison of segment-wise precision between the two labeling approaches. Our method outperforms traditional methods by 54.2% in segment-wise precision.

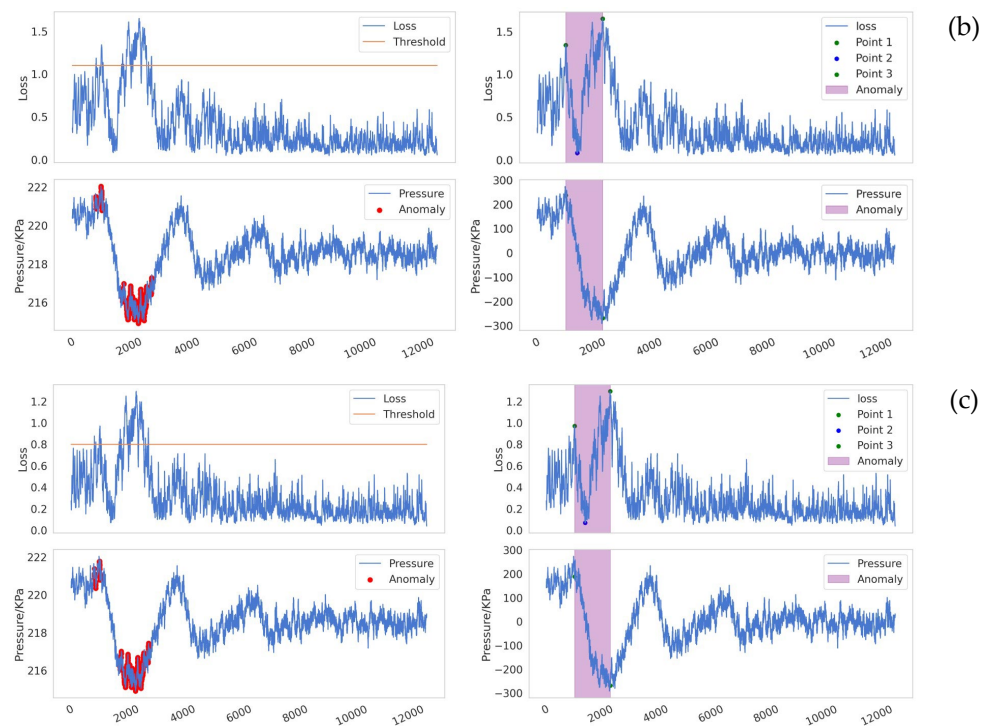


**Figure 12.** The composition of the six detectors in the comparative experiment.

From the detection instances (Figure 13), it is evident that all six detectors perform well and meet the requirements for industrial applications. Notably, TKAN-AE, along with the other two established models, maintains strong performance even when there are abrupt changes in the original data, demonstrating the models' robustness in complex environments. Additionally, our proposed labeling method has been successfully applied across different models, providing strong experimental support for its generalization capabilities, while also reducing the false negatives and false positives commonly seen in traditional methods.

Table 3 provides a comparative analysis of the three models discussed above, with a focus on ensuring fairness by using the proposed new labeling method for all models. In conjunction with the confusion matrices of the three models shown in Figure 15(a)(b)(c), it is evident that all three models demonstrate strong performance in anomaly detection accuracy on real-world datasets, as well as in identifying specific anomalous sequences effectively. TKAN achieves a balanced performance, with both Precision and Recall being well-aligned, resulting in an F1-score of 75%, which underscores the model's overall stability. In contrast, LSTM shows a higher recall rate, but its lower precision compared to TKAN suggests a tendency toward false positives. Additionally, GRU's relatively lower recall rate indicates a potential issue with false negatives, highlighting its limitations in capturing certain anomalies. Overall, each model has its unique advantages. As an emerging model, TKAN demonstrates such balanced performance without extensive engineering tuning, which undoubtedly provides a promising outlook for its future applications. At the same time, the success of TKAN opens up new possibilities for replacing traditional fully connected layer-based networks (such as MLP).

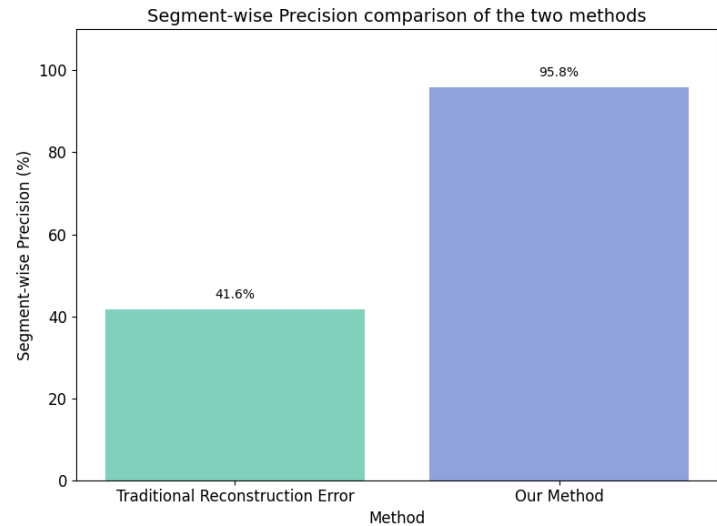




**Figure 13.** The performance of six detection methods on leakage sample segments.

In terms of memory usage (measured by total number of parameters), the autoencoder (AE) network based on the TKAN and GRU architectures has fewer parameters. However, with respect to training speed on a GPU, classical time series models significantly outperform the TKAN network. It is important to note that the KAN network, being a recently introduced model, has not yet undergone optimization for efficiency or specific adaptation to GPU architectures, which contributes to its relatively poor GPU compatibility. Moreover, from a technical standpoint, the inclusion of learnable activation functions enhances model interpretability but incurs higher evaluation costs compared to fixed activation functions, leading to slower training speeds. This concern was also highlighted by Liu Ziming, the author of the KAN model.

Interestingly, when training on a CPU, the TKAN network surpasses traditional LSTM models in terms of training speed. This is likely because the CPU architecture is better suited to handle the complex, sequential computations involved. Notably, the TKAN model even outperforms the GRU, which is well known for its lightweight nature and fast training capabilities.





**Figure 14.** Segment-wise Precision of two methods for specific detection instances.

**Table 3.** Performance comparison between the proposed networks and benchmark timeseries models across 8 metrics.

Model	Accuracy(%)	Segment-wise Precision(%)	Precision(%)	Recall(%)	F1- score(%)	Total Parameters	GPU Cost(s)	CPU Cost(s)
TKAN-AE	97.1	93.1	75.0	75.0	75.0	73544	188.9	125.8
LSTM-AE	98.1	93.0	68.7	91.6	78.3	96331	23.2	162.0
GRU-AE	96.6	92.8	72.7	66.7	69.5	73631	12.8	150.7

<sup>1</sup> **Total Parameters** :The total parameter count represents the sum of all trainable parameters in a deep learning model, serving as an indicator of the model's complexity. A larger number of parameters generally enhances the model's learning capacity, enabling it to capture more intricate patterns and details. However, an excessive number of parameters may lead to overfitting, particularly when the dataset is insufficient. Additionally, larger models require greater computational resources, resulting in higher training and inference costs. Therefore, the total parameter count is a crucial metric for evaluating both the complexity of the model and the associated computational resource demands.

5. Discussion

Although the proposed method introduces a novel approach for RNN-based models, which traditionally rely heavily on MLP architectures, and demonstrates strong performance on real-world pipeline datasets, several limitations remain:

1. Data Scarcity: Despite utilizing a real-world pipeline dataset rarely employed in previous studies, the scarcity of anomaly samples presents a significant challenge, largely due to the high costs and safety considerations associated with leakage experiments. Moreover, the completely unsupervised learning approach employed in this study did not fully maximize the use of these rare anomaly samples, which continues to be a challenge in the field of anomaly detection. Potential solutions include:

(1)Transfer Learning: By combining laboratory pipeline datasets with real-world datasets, transfer learning can effectively leverage existing knowledge to enhance model performance in new environments at reduced costs.

(2)Semi-Supervised Learning: As demonstrated by Guansong Pang et al.[23], semi-supervised learning is one of the most effective methods in anomaly detection. This approach combines a small amount of labeled anomaly data with a large volume of unlabeled normal data for training. By learning the characteristics of normal data, semi-supervised learning can more accurately identify anomalies, addressing the inefficiency of unsupervised learning in utilizing known anomaly information while also avoiding the reliance on large amounts of labeled data typical of fully supervised methods.

2. Model Adaptability: The proposed model is tailored specifically for anomaly detection in a particular urban pipeline network. Unlike fluid-based models, our method benefits from not requiring knowledge of physical parameters such as pipeline geometry, fluid properties, or pressure, instead relying directly on monitoring data to train the leakage detection model. These monitoring data inherently contain physical information about the pipeline network. However, if the configuration of the pipeline network changes, the model must be retrained. Recent advances in federated learning have shown promise in addressing this adaptability issue. Federated learning integrates resources and updates models in real time, while reducing computational costs through distributed computing. More importantly, in an era where data security is of paramount concern, federated learning's robust data privacy protection capabilities have gained considerable attention.

3. Model Interpretability: While the proposed model takes advantage of the strong fitting capability and reduced parameter complexity of the KAN network, it does not fully exploit another critical feature of KAN—its interpretability. This underutilized feature is closely tied to the task of leak point localization in pipeline leakage detection. Accurate localization of leaks depends on determining the velocity of negative pressure waves, which is often estimated using empirical formulas or complex fluid numerical simulations involving challenging partial differential equation solutions. Addressing these interpretability issues, particularly in the context of leak point localization, is an important direction for future work.

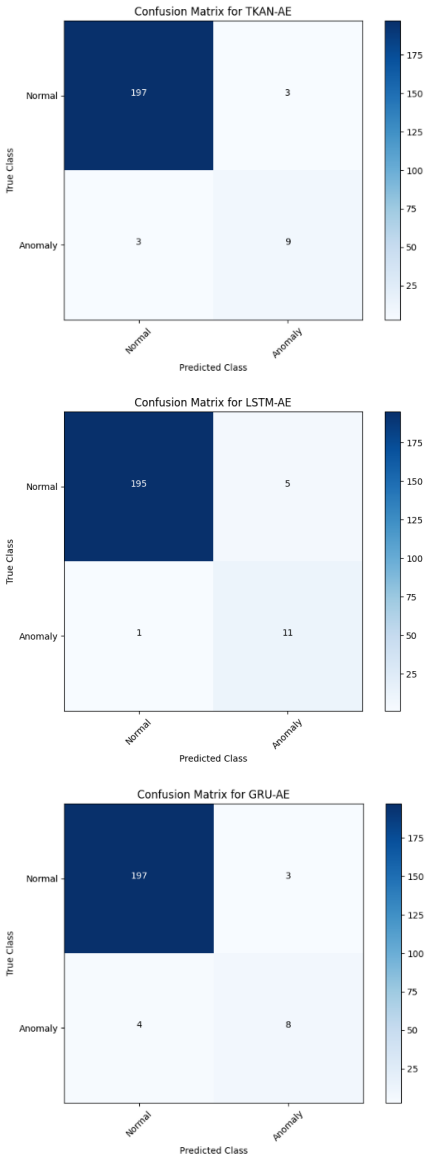


Figure 15. The confusion matrices of three models.

6. Conclusions

- 1.The proposed TKAN-AE model demonstrates exceptional performance and stability, effectively identifying signal features associated with pipeline leaks. Compared to traditional time series models, TKAN-AE significantly reduces the number of parameters, making it particularly suitable for large-scale, cost-efficient industrial deployment. Furthermore, this model offers a novel alternative to traditional fully connected networks, such as MLP.
- 2.The improved anomaly sequence labeling method substantially enhances the accuracy of time series anomaly detection, achieving a segment-level detection precision of 93.1% for specific

anomalous sequences. The application of this method across three different models further confirms its strong generalization capability and robustness.

3. This study validates the proposed detector using real-world pipeline leakage experimental data from a megacity. The results demonstrate superior performance and robustness, providing strong support for its potential application in industrial settings.

**Supplementary Materials:** The code used as demonstration in this paper has now been open-sourced on GitHub. It can be run directly on Colab to reproduce the experimental results presented in this work. The link to access it is: <https://github.com/wuhyyy/TKAN-AE>

**Author Contributions:** Author Contributions: Conceptualization, Hengyu Wu; methodology, Hengyu Wu; software, Hengyu Wu; investigation, Zhu Jiang; resources, Zhu Jiang; data curation, Jian Cheng; writing—original draft preparation, Hengyu Wu; writing—review and editing, Zhu Jiang, Xiang Zhang; supervision, Xiang Zhang; project administration, Xiang Zhang. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Sichuan Science and Technology Program, grant numbers 24NSFSC1460, the Further Research on Leakage Alarm and Precise Location of Large Diameter Raw Water Pipeline, grant numbers 232209, and Xihua University Science and Technology In-novation Competition Project for Postgraduate Students, grant numbers YK20240217.

## References

1. Climate-ADAPT, "Water restrictions and consumption cuts," 2023. [Online]. Available: <https://climate-adapt.eea.europa.eu/en/metadata/adaptation-options/water-restrictions-and-consumption-cuts>. Accessed: Aug. 23, 2023.
2. Horsburgh J S, Leonardo M E, Abdallah A M, et al. Measuring water use, conservation, and differences by gender using an inexpensive, high frequency metering system[J]. *Environmental modelling & software*, 2017, 96: 83-94.
3. *China Urban and Rural Construction Statistical Yearbook*, Ministry of Housing and Urban-Rural Development, 2023. [Online]. Available: <https://www.stats.gov.cn/sj/ndsj/2023/indexeh.htm>.
4. Pahlavanzadeh, F., Khaloozadeh, H. & Forouzanfar, M. Online fault detection and localization of multiple oil pipeline leaks using model-based residual generation and friction identification. *Int. J. Dynam. Control* **12**, 2615–2628 (2024). <https://doi.org/10.1007/s40435-024-01386-8>
5. Abuhatira A A, Salim S M, Vorstius J B. CFD-FEA based model to predict leak-points in a 90-degree pipe elbow[J]. *Engineering with Computers*, 2023, 39(6): 3941-3954.
6. Delgado-Aguinaga J A, Puig V, Becerra-López F I. Leak diagnosis in pipelines based on a Kalman filter for Linear Parameter Varying systems[J]. *Control Engineering Practice*, 2021, 115: 104888.
7. Pujades E, Vázquez-Suñé E, Culí L, et al. Hydrogeological impact assessment by tunnelling at sites of high sensitivity[J]. *Engineering Geology*, 2015, 193: 421-434.
8. Carrera R, Verde C, Cayetano R. A SCADA expansion for leak detection in a pipeline[J]. *Sensors*, 2015, 2300(2320): 2340.
9. Lopezlena R, Sadovnychiy S. Pipeline leak detection and location using boundary feedback estimation: case study[J]. *Journal of Pipeline Systems Engineering and Practice*, 2019, 10(3): 04019017.
10. Torres L, Verde C, Molina L. Leak diagnosis for pipelines with multiple branches based on model similarity[J]. *Journal of Process Control*, 2021, 99: 41-53.
11. Romero-Tapia G, Fuente M J, Puig V. Leak localization in water distribution networks using Fisher discriminant analysis[J]. *IFAC-PapersOnLine*, 2018, 51(24): 929-934.
12. Y. Xu, X. Liu, X. Cao, C. Huang, E. Liu, S. Qian, X. Liu, Y. Wu, F. Dong, and C.-W. Qiu, "Artificial intelligence: A powerful paradigm for scientific research," *The Innovation*, vol. 2, no. 4, p. 100179, 2021.
13. Quinones-Grueiro M, Bernal-de Lázaro J M, Verde C, et al. Comparison of classifiers for leak location in water distribution networks[J]. *IFAC-PapersOnLine*, 2018, 51(24): 407-413.
14. Quiñones-Grueiro M, Verde C, Prieto-Moreno A, et al. An unsupervised approach to leak detection and location in water distribution networks[J]. *International Journal of Applied Mathematics and Computer Science*, 2018, 28(2): 283-295.
15. Xie Y, Xiao Y, Liu X, et al. Time-frequency distribution map-based convolutional neural network (CNN) model for underwater pipeline leakage detection using acoustic signals[J]. *Sensors*, 2020, 20(18): 5040.
16. Zhang P, He J, Huang W, et al. Water pipeline leak detection based on a pseudo-siamese convolutional neural network: Integrating handcrafted features and deep representations[J]. *Water*, 2023, 15(6): 1088.
17. Cody R A, Tolson B A, Orchard J. Detecting leaks in water distribution pipes using a deep autoencoder and hydroacoustic spectrograms[J]. *Journal of Computing in Civil Engineering*, 2020, 34(2): 04020001.

18. Wang X, Guo G, Liu S, et al. Burst detection in district metering areas using deep learning method[J]. *Journal of Water Resources Planning and Management*, 2020, 146(6): 04020031.
19. Tian C H, Yan J C, Huang J, et al. Negative pressure wave based pipeline leak detection: Challenges and algorithms[C]//*Proceedings of 2012 IEEE international conference on service operations and logistics, and informatics*. IEEE, 2012: 372-376.
20. Hewamalage H, Bergmeir C, Bandara K. Recurrent neural networks for time series forecasting: Current status and future directions[J]. *International Journal of Forecasting*, 2021, 37(1): 388-427.
21. Mienye I D, Swart T G, Obaido G. Recurrent neural networks: A comprehensive review of architectures, variants, and applications[J]. *Information*, 2024, 15(9): 517.
22. Hua Y, Zhao Z, Li R, et al. Deep learning with long short-term memory for time series prediction[J]. *IEEE Communications Magazine*, 2019, 57(6): 114-119.
23. Kim H, Park M, Kim C W, et al. Source localization for hazardous material release in an outdoor chemical plant via a combination of LSTM-RNN and CFD simulation[J]. *Computers & Chemical Engineering*, 2019, 125: 476-489.
24. Yang L, Zhao Q. A novel PPA method for fluid pipeline leak detection based on OPELM and bidirectional LSTM[J]. *IEEE Access*, 2020, 8: 107185-107199.
25. Hu X, Chu L, Pei J, et al. Model complexity of deep learning: A survey[J]. *Knowledge and Information Systems*, 2021, 63: 2585-2619.
26. Liu Z, Wang Y, Vaidya S, et al. Kan: Kolmogorov-arnold networks[J]. *arXiv preprint arXiv:2404.19756*, 2024.
27. Genet R, Inzirillo H. Tkan: Temporal kolmogorov-arnold networks[J]. *arXiv preprint arXiv:2405.07344*, 2024.
28. Zhang Y, Wallace B. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification[J]. *arXiv preprint arXiv:1510.03820*, 2015.
29. Shtayat A, Moridpour S, Best B, et al. Application of noise-cancelling and smoothing techniques in road pavement vibration monitoring data[J]. *International Journal of Transportation Science and Technology*, 2024, 14: 110-119.
30. Zhang X, Shi J, Yang M, et al. Real-time pipeline leak detection and localization using an attention-based LSTM approach[J]. *Process Safety and Environmental Protection*, 2023, 174: 460-472.
31. Ruff L, Kauffmann J R, Vandermeulen R A, et al. A unifying review of deep and shallow anomaly detection[J]. *Proceedings of the IEEE*, 2021, 109(5): 756-795.
32. Hinton G E, Zemel R. Autoencoders, minimum description length and Helmholtz free energy[J]. *Advances in neural information processing systems*, 1993, 6.
33. Aljameel S S, Alabbad D A, Alomari D, et al. Oil and Gas Pipelines Leakage Detection Approaches: A Systematic Review of Literature[J]. *International Journal of Safety & Security Engineering*, 2024, 14(3).
34. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
35. Li J, Malialis K, Polycarpou M M. Autoencoder-based anomaly detection in streaming data with incremental learning and concept drift adaptation[C]//*2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2023: 1-8.
36. Liu, H., & Chen, C. (2019). Multi-objective data-ensemble wind speed forecasting model with stacked sparse autoencoder and adaptive decomposition-based error correction. *Applied Energy*, 251, 113400. <https://doi.org/10.1016/j.apenergy.2019.113400>
37. A. N. Kolmogorov, "On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition," *Dokl. Akad. Nauk SSSR*, vol. 114, no. 5, pp. 953-956, 1957.
38. A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," *ACM Comput. Surv.*, vol. 54, no. 3, pp. 1-33, 2021.
39. Li J, Zheng Q, Qian Z, et al. A novel location algorithm for pipeline leakage based on the attenuation of negative pressure wave[J]. *Process Safety and Environmental Protection*, 2019, 123: 309-316.
40. Kumar U, Bhar A. Studying and analysing the effect of weight norm penalties and dropout as regularizers for small convolutional neural networks[J]. *Int J Eng Res Technol (IJERT)*, 2021, 10: 47-51.
41. B. El Asri, and M. Rhanoui, "Mechanism of Overfitting Avoidance Techniques for Training Deep Neural Networks," in *Proc. ICEIS*, 2022, pp. 418-427.
42. S. Aburass, "Quantifying Overfitting: Introducing the Overfitting Index," *arXiv:2308.08682*, 2023.
43. Pang G, Shen C, Cao L, et al. Deep learning for anomaly detection: A review[J]. *ACM computing surveys (CSUR)*, 2021, 54(2): 1-38.
44. Lavin, Alexander, and Subutai Ahmad. "Evaluating real-time anomaly detection algorithms – the Numenta anomaly benchmark." *14th International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 2015, <https://arxiv.org/pdf/1510.03336>.
45. Zamanzadeh Darban, Z., Webb, G. I., and Pan, S. "Time series anomaly detection: Survey and benchmark." *ACM Computing Surveys*, 2024, <https://dl.acm.org/doi/abs/10.1145/3691338>.

46. Pang, G., Shen, C., Cao, L., & van den Hengel, A. (2020). Deep learning for anomaly detection: A review. *ACM Computing Surveys (CSUR)*, 53(5), <https://doi.org/10.48550/arXiv.2007.02500>

**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.