

# Abnormal State Detection using Memory-augmented Autoencoder technique in Frequency-Time Domain

**Haoyi Zhong, Yongjiang Zhao and Chang Gyoon Lim\***

Department of Computer Engineering, Chonnam National University  
Yeosu 59626, South Korea  
[e-mail: cglim@jnu.ac.kr]

\*Corresponding author: Chang Gyoon Lim

*Received September 25, 2023; revised November 28, 2023; accepted January 11, 2024;  
published February 29, 2024*

---

## **Abstract**

With the advancement of Industry 4.0 and Industrial Internet of Things (IIoT), manufacturing increasingly seeks automation and intelligence. Temperature and vibration monitoring are essential for machinery health. Traditional abnormal state detection methodologies often overlook the intricate frequency characteristics inherent in vibration time series and are susceptible to erroneously reconstructing temperature abnormalities due to the highly similar waveforms. To address these limitations, we introduce synergistic, end-to-end, unsupervised Frequency-Time Domain Memory-Enhanced Autoencoders (FTD-MAE) capable of identifying abnormalities in both temperature and vibration datasets. This model is adept at accommodating time series with variable frequency complexities and mitigates the risk of overgeneralization. Initially, the frequency domain encoder processes the spectrogram generated through Short-Time Fourier Transform (STFT), while the time domain encoder interprets the raw time series. This results in two disparate sets of latent representations. Subsequently, these are subjected to a memory mechanism and a limiting function, which numerically constrain each memory term. These processed terms are then amalgamated to create two unified, novel representations that the decoder leverages to produce reconstructed samples. Furthermore, the model employs Spectral Entropy to dynamically assess the frequency complexity of the time series, which, in turn, calibrates the weightage attributed to the loss functions of the individual branches, thereby generating definitive abnormal scores. Through extensive experiments, FTD-MAE achieved an average ACC and F1 of 0.9826 and 0.9808 on the CMHS and CWRU datasets, respectively. Compared to the best representative model, the ACC increased by 0.2114 and the F1 by 0.1876.

---

**Keywords:** Abnormal State Detection, Memory Augmentation, Autoencoder, Frequency-Time Domain.

## 1. Introduction

As Industry 4.0 emerges and Industrial Internet of Things (IIoT) technology matures, contemporary manufacturing is evolving towards increased intelligence and automation. The IIoT incorporates sensors, control mechanisms, and analytical tools into industrial machinery, facilitating real-time data gathering, remote oversight, and predictive upkeep [1]. While these advancements enhance productivity and the efficiency of resource utilization, they also present considerable challenges in maintaining the safety and reliability of such systems. Under these circumstances, monitoring of temperature and vibration has become critical parameters in evaluating the health status of machinery. These metrics offer early alerts to engineers and maintenance staff, enabling them to proactively intervene before issues worsen. For instance, steam leakage in steam traps compromises system efficiency and leads to energy wastage. By monitoring the temperature and vibrations of the steam traps, such leaks can be detected, allowing for prompt maintenance and resource-saving [2, 3]. Hence, it is imperative to devise efficient and precise methods for detecting temperature and vibration abnormalities.

Conventional supervised approaches for abnormal state detection depend on a fully annotated training set comprising both normal and abnormal samples. These techniques usually employ standard classification algorithms such as Support Vector Machines (SVM) [4], Decision Trees (DT) [5], or Neural Networks [6]. While these methods excel in a variety of application settings, their general applicability is constrained by the availability of labeled data. Practically, acquiring abnormal samples is often a challenging task due to their rarity [7].

In contrast to supervised methods for abnormal state detection, unsupervised techniques can function without the need for labeled samples and primarily consist of statistical, distance-based, and density-based methods. For instance, K-Nearest Neighbors (K-NN) [8] and Isolated Forest (IF) [9] are renowned unsupervised algorithms. While these techniques do not necessitate labels, they frequently require manual tuning of parameters, which poses challenges in high-dimensional or volatile data settings. Moreover, their high computational complexity leads to substantial resource and time consumption, and their sensitivity to noise results in elevated rates of false alarms [10].

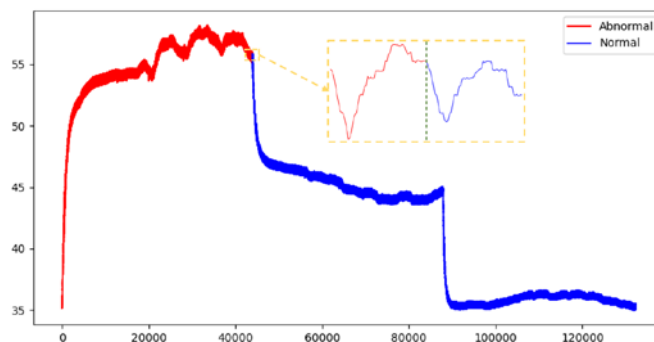


Fig. 1. Comparison of abnormal and normal samples in CMHS dataset.

Lately, methods of abnormal state detection that rely on reconstruction loss have become a promising field. These approaches utilize deep learning models such as autoencoders (AE) to learn normal data patterns, using reconstruction loss (the discrepancy between original and reconstructed data) as a measure for abnormal scoring. In contrast to traditional techniques, these methods can self-learn complex data patterns without manual feature engineering or parameter adjustments, typically demonstrating robustness and scalability in high-dimensional

environments [11]. Nonetheless, challenges persist in employing AE-based approaches due to the subsequent issues:

(1) In the context of industrial equipment's temperature data, the variation patterns between normal and abnormal samples within each sampling period are highly alike. This resemblance can lead AE's occasional good generalizing capabilities to mistakenly categorize abnormal samples as normal [12], as shown in Fig. 1. A green dashed line segments the temperature data gathered by the sensor in every cycle. The blue curve marks the normal samples, while the red curve signifies the abnormal ones.

(2) AE used for abnormal state detection is generally crafted to reconstruct input data within the time domain. Yet, the majority of vibration data are not stationary and have features like a broad dynamic scope and rapid frequency shifts. Abnormalities usually are not a matter of value increase or decrease at a specific time but more about changes in frequency during the sample period, as shown in Fig. 2. Fig. 2 (a) displays normal vibrational signals and Fig. 2 (c) reveals the spectral graph of those normal signals post-Fast Fourier Transformation (FFT) [13]. Fig. 2 (b) represents abnormal vibrational signals and Fig. 2 (d) showcases their spectral graph following FFT. As can be observed, the normal signals are rich in low-frequency elements, while the abnormal signals have a considerable number of high-frequency elements.

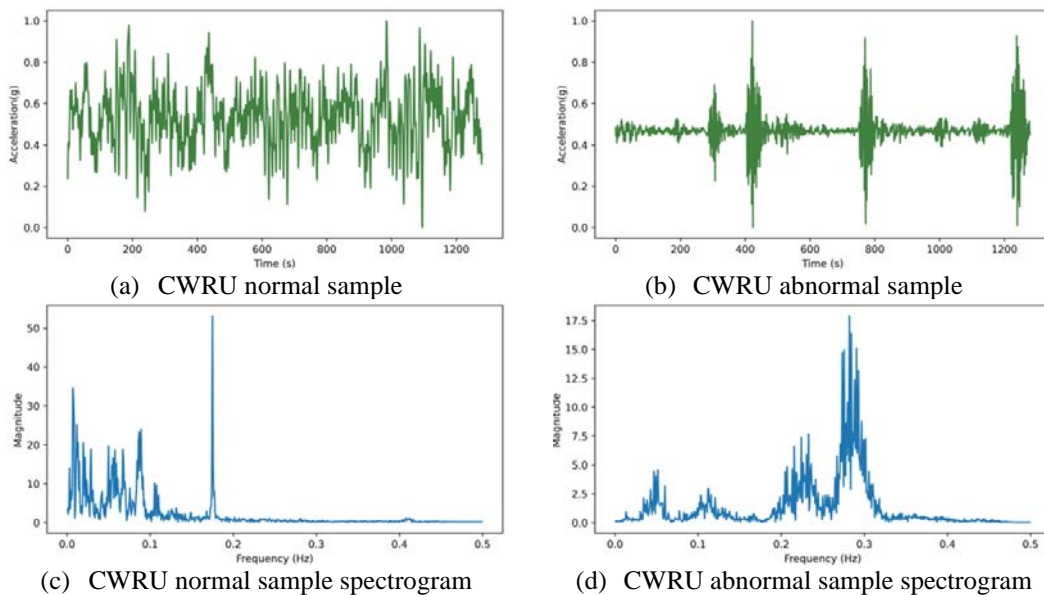


Fig. 2. Comparison of normal and abnormal vibration data and their spectrum plot after Fast Fourier Transform in the CWRU dataset.

To tackle the aforementioned challenges, we introduce a new unsupervised abnormal state detection model tailored for temperature and vibration time series, named FTD-MAE (Frequency-Time Domain Memory-augmented Autoencoders). Provided an input time series, the model processes it through both time domain and frequency domain branches, resulting in two kinds of reconstruction losses.

For the time domain branch, the input is encoded into a latent representation through two linear layers. This latent representation serves as a query to weight each item in memory, generating a latent representation most similar to normal samples. Ultimately, the generated latent representation is utilized by the decoder to create a reconstructed time series.

In the frequency domain branch, the input is initially transformed into a spectrogram using Short-Time Fourier Transform (STFT) [14]. Subsequently, convolutional layers encode this spectrogram into a latent representation. Similar to the time domain branch, this latent representation also serves as a query to weight each item in memory, generating a latent representation most similar to normal samples. Finally, the generated latent representation is decoded to produce the reconstructed image.

The time domain branch focuses on detecting abnormalities in data with simple frequencies. On the other hand, the frequency domain branch aims to identify abnormalities in data with complex frequencies. Given that unseen data may have different levels of complexity compared to the training data, it becomes necessary to calculate the weights. These weights correspond to each of the two reconstruction losses after they have been obtained. These weights are derived from the ratio of the two losses and a Spectral Entropy test conducted on the unseen data [15]. Then the weighted losses are summed to arrive at the final abnormal score.

In summary, this paper makes the following major contributions:

- We propose an autoencoder model that integrates frequency and time information for abnormal state detection in temperature and vibration time series data. It is an unsupervised, end-to-end model, capable of efficiently handling data with varying frequency complexities.
- We introduce a memory-augmented method that can encode and constrain the latent representations of normal samples, making the reconstructed samples less prone to generalization.
- Experiments on two real datasets demonstrate the effectiveness of the FTD-MAE model, outperforming other representative algorithms. The results emphasize the benefits of using a memory module and a dual-branch architecture. Additionally, ablation studies and sensitivity tests for hyperparameters were conducted, further validating the effectiveness and robustness of the proposed method.

## 2. Related Work

In recent years, the field of time-series abnormal state detection has received widespread attention from the academic community [16]-[18]. Classical techniques for detecting abnormalities (values that significantly deviate from many observations) can be categorized into several types. These include methods based on distance metrics [8, 19], methods employing density calculation [20, 21], isolation-based methods [9, 22], and strategies based on statistical inference [23, 24].

### 2.1 Reconstruction-based time-series abnormal state detection

Recently, deep learning-based abnormal state detection methods have started to gain popularity. Compared to deep learning methods, traditional methods have shortcomings in aspects like feature learning, as they may require manual feature engineering, which is both time-consuming and potentially inaccurate. Deep learning models can automatically learn useful features from the data without relying on domain expertise [25]. One approach focuses on using autoencoders with reconstruction errors for abnormal state detection. Examples include AE, Variational Autoencoders (VAE) [26], autoencoders based on Recurrent Neural Networks [27], and autoencoders based on GANs [28]. However, there are issues with reconstruction-based abnormal state detection. The main reasons are twofold. First, AE has strong generalization capabilities, which makes it easy for abnormalities in waveform-similar

data like temperature data to be reconstructed, thereby affecting the performance of abnormal state detection. Second, for non-stationary data like vibration data, AE tends to overlook frequency characteristics.

## 2.2 Preprocessing methods for frequency complex signals

Signals with complex frequency components have characteristics such as multiple frequency components, susceptibility to noise interference, and nonlinear properties. Preprocessing these types of signals is a common problem in signal processing and time series analysis, the primary aim of which is to make the signal or data smoother for easier subsequent analysis or processing. Common methods include differencing, moving averages [29], wavelet transform [30], Hilbert-Huang transform (HHT) [31] and STFT. Among them, only wavelet transform, HHT, and STFT are time-frequency graph transform methods that can map 1 dimensional data to 2 dimensions, and we compare these three methods in Section 4.7. In comparison to the differencing method, moving averages, wavelet transformations and HHT, STFT offers an intuitive bi-dimensional representation of time and frequency, which enables clearer capturing of the signal's time and spectral attributes. Therefore, we opt to use STFT for the analysis of the signal.

## 2.3 Evaluation of frequency complexity

While the STFT can reduce data complexity, it also inherently causes a certain level of data loss. Hence, data exclusively processed by STFT may not be fully reliable. In order to strengthen the model's robustness, we use a weighted blend of both STFT-processed and non-STFT-processed data. It's important to note that setting static weights is impractical since the features of unseen data cannot be predicted. Therefore, it is necessary to dynamically adjust these weights by continually evaluating the frequency complexity of both the original and unseen data. There are multiple ways to evaluate the frequency complexity of data, including statistical approaches [32], Renyi Entropy [33] and Spectral Entropy [15]. In this study, we opt for the Spectral Entropy. Compared to other methods, Spectral Entropy does not rely on basic assumptions and has fewer data requirements. It offers a quantitative way to assess the uncertainty or complexity of the spectral distribution in time series data.

## 2.4 Memory networks

Weston et al. introduced Memory Networks in 2014 [34]. This is a memory model oriented towards external content, applicable to a range of tasks including question answering and language modeling. Later, Sukhbaatar et al. proposed an end-to-end version, making it easier to train [35]. Kumar et al. proposed Dynamic Memory Networks (DMN) as a variant of Memory Networks. DMN integrates external memory components with Recurrent Neural Networks (RNN). This enables the model to initially handle the input through RNN and subsequently store the outcome in the memory modules [36].

# 3. Methodology

Our proposed model can be deployed on a cloud server to receive temperature or vibration signals sent by sensors. When the reconstruction error exceeds a set threshold, it is considered an abnormal state. This is when engineers and maintenance personnel are notified, enabling proactive intervention before the problem worsens.

### 3.1 Overview

The proposed FTD-MAE model consists of two branches: the frequency domain branch and the time domain branch. The frequency domain branch is made up of four key components: an STFT module (transforming the time series into a 2D spectrogram), an encoder (encoding inputs and queries), a decoder (reconstructing the input), and a memory module (memorizing the latent representation of normal samples). The encoder consists of three convolutional layers and the decoder consists of four linear layers. The time domain branch resembles the frequency domain branch but lacks the STFT component. Both the encoder and decoder consist of two linear layers. As shown in Fig. 3, the input first enters the time domain branch (green line) where the encoder captures its latent representation. Utilizing this latent representation as a query, the memory module learns the distance between memories via attention mechanisms and then passes it on to the decoder for reconstruction. The original input traverses the frequency domain branch (blue line), initially undergoing STFT transformation into a spectrogram, then follows the same steps as the time domain branch to generate a reconstructed image. The frequency complexity coefficient of the input is determined by Spectral Entropy, which is then used to set the weights for the reconstructed time series and image, ultimately yielding an abnormal score.

During the training process, the time domain and frequency domain branches are trained separately. Optimization is carried out on the encoder and decoder to minimize the reconstruction error and the memory content is simultaneously updated to capture the latent representation of the normal data.

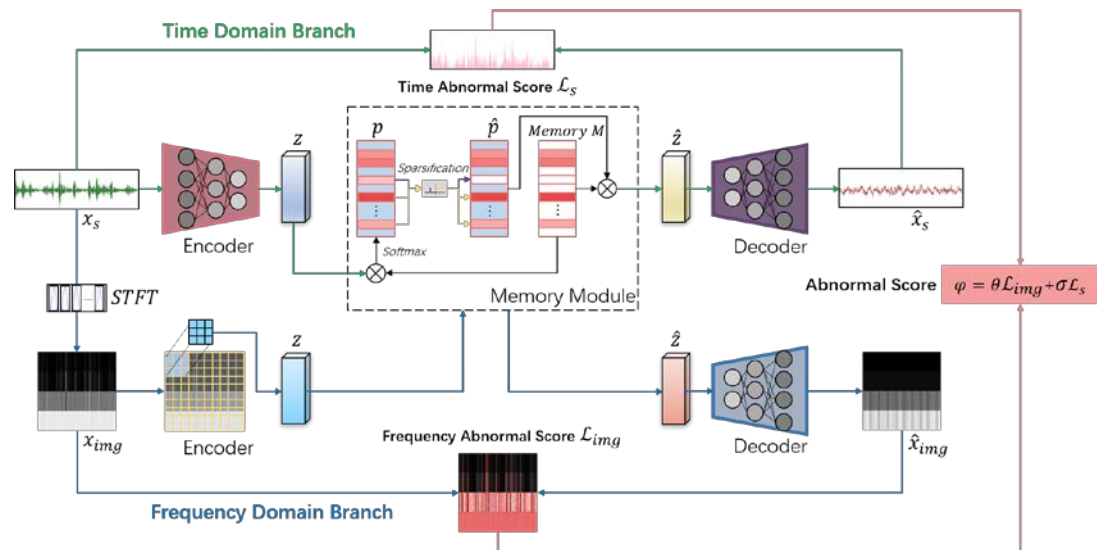


Fig. 3. An overview of the FTD-MAE.

### 3.2 Time-series Data Preprocessing

In vibration data, larger numerical values can expand the range of scales, rendering the model insensitive to minor variations in smaller numerical values, which could result in the neglect of their crucial properties. Hence, considering the sensor context information under the same timestamp, it is necessary to standardize the raw data for all temporal dimensions.

The raw data  $D \in \mathbb{R}^{n \times T}$  is a matrix that consists of  $n$  samples of time series. The length of each time series is  $T$ , which signifies the data gathered during each sampling cycle. For every

data point  $d$ , the maximum and minimum values in its corresponding column are utilized as the parameters for its normalization. The normalized data  $x$  is shown in (1).

$$x = \frac{d - \text{Min}_{col}(d, D)}{\text{Max}_{col}(d, D) - \text{Min}_{col}(d, D)} \quad (1)$$

where each column represents a time series segment.  $\text{Min}_{col}(d, D)$  refers to the minimum value in the column in which  $d \in D$  is located.  $\text{Max}_{col}(d, D)$  refers to the maximum value in the column in which  $d \in D$  is located.

In temperature data with each row represents a time series segment, normalizing along the columns can result in substantial alterations to some time series waveforms, leading to a loss of their inherent properties. Therefore, for temperature time series, it is necessary to normalize along the rows to maintain the waveform while reducing the range. The normalized data  $x$  is shown in (2).

$$x = \frac{d - \text{Min}_{row}(d, D)}{\text{Max}_{row}(d, D) - \text{Min}_{row}(d, D)} \quad (2)$$

where  $\text{Min}_{row}(d, D)$  refers to the minimum value in the row in which  $d \in D$  is located.  $\text{Max}_{row}(d, D)$  refers to the maximum value in the row in which  $d \in D$  is located.

While this method of data normalization enhances the stability of the algorithm for abnormal state detection, it may cause the model to fit larger temporal trends due to similar trends across various temperature data segments, thus neglecting fluctuations at smaller time scales. Hence, the time domain branch of FTD-MAE attenuates the trends in temperature time series and pays more attention to short-term fluctuations, while the frequency domain branch does not dampen the trends due to the scale issues of its window function. (Preprocessing performance analysis is detailed in the section 4.6). The equation below is utilized to elucidate the procedure for detrending preprocessing.

$$J(\alpha, \beta) = \sum_{t=1}^T (x(t) - \alpha - \beta t)^2 \quad (3)$$

where  $J(\alpha, \beta)$  is the minimization objective function, and the values of  $\alpha$  and  $\beta$  need to be estimated by least squares to minimize  $J(\alpha, \beta)$ .  $x(t) \in \mathbb{R}^T$  is the original input. (4) and (5) can be derived by taking the partial derivatives of  $J(\alpha, \beta)$  with respect to  $\alpha$  and  $\beta$  and setting them to zero.

$$\frac{\partial J}{\partial \alpha} = -2 \sum_{t=1}^T (x(t) - \alpha - \beta t) = 0 \quad (4)$$

$$\frac{\partial J}{\partial \beta} = -2 \sum_{t=1}^T t(x(t) - \alpha - \beta t) = 0 \quad (5)$$

Solving these two equations will result in (6) and (7).

$$\alpha = \frac{\sum_{t=1}^T x(t) - \beta \sum_{t=1}^T t}{T} \quad (6)$$

$$\beta = \frac{T \sum_{t=1}^T tx(t) - \sum_{t=1}^T x(t) \sum_{t=1}^T t}{T \sum_{t=1}^T t^2 - (\sum_{t=1}^T t)^2} \quad (7)$$

After obtaining the estimated values of  $\alpha$  and  $\beta$ , substituting them into (8) yields the output trend  $y(t)$ .

$$y(t) = x(t) - (\alpha + \beta t) \quad (8)$$

### 3.3 Model Structure

#### 3.3.1 Short-time Fourier Transform

As shown in Fig. 4, when the time series is input into the frequency domain branch of the model, it first undergoes STFT to produce a two-dimensional image containing time and corresponding frequencies. The time series is then decomposed into  $M$  sub-sequences with overlapping sections through a sliding Hamming window. The expression is shown in (9).

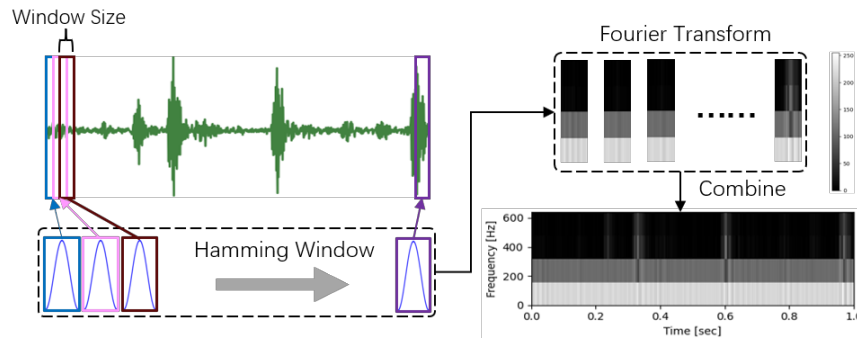


Fig. 4. STFT workflow implementation with Hamming Window.

$$M = 1 + \frac{M_{ori} - nperseg}{nperseg - noverlap} \quad (9)$$

where  $M_{ori}$  denotes the number of samples of the original signal, the window length is  $nperseg$  and the window overlap length is  $noverlap$ . This allows each subsequence to smoothly converge to zero at both ends, thus reducing the spectral leakage due to non-periodicity, and the expression for the Hamming window is shown in (10).

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) \quad (10)$$

where  $w(n)$  is the value of the window function at moment  $n$ ,  $N$  is the total length of the window function and  $n$  is the index of each discrete time point of the window function.

Window length selection can be made according to the characteristics of the signal. For signals with a fast change trend (temperature), a shorter window can be used to better capture changes in time. For signals that trend more slowly (vibration), a longer window can be used for more accurate frequency analysis.

The aggregation of Fourier transformations for each sub-sequence constitutes the outcome of the STFT, as indicated in (11).



$$STFT\{x[n]\}(m, \omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n} \quad (11)$$

where  $x[n]$  is the input signal,  $w[n-m]$  is the window function,  $m$  is the time domain index of the window centre,  $\omega$  is the frequency,  $STFT\{x[n]\}(m, \omega)$  is the STFT of the signal at time domain index  $m$  and frequency  $\omega$ .

### 3.3.2 Encoder and Decoder

The encoder is responsible for extracting the latent representation from the input, which is then used as a query to fetch relevant units stored in memory. The decoder is trained to reconstruct samples based on memory and queries. The formulations for the encoder and decoder are presented in (12) and (13).

$$\mathbf{z} = f_e(\mathbf{x}) \quad (12)$$

$$\hat{\mathbf{x}} = f_d(\hat{\mathbf{z}}) \quad (13)$$

where  $\mathbf{z}$  denotes the latent representation of the output of the input sample  $\mathbf{x}$  (input time series or input image) after passing through the encoder  $f_e(\cdot)$ . Unlike standard autoencoders,  $\mathbf{z}$  in each branch of FTD-MAE is not equal to  $\hat{\mathbf{z}}$ .  $\mathbf{z}$  is passed through the retrieval memory to get latent representation  $\hat{\mathbf{z}}$ , and then  $\hat{\mathbf{z}}$  is passed through the decoder  $f_d(\cdot)$  to get the reconstruction sample  $\hat{\mathbf{x}}$  (reconstructed time series or reconstructed images).

### 3.3.3 Memory Module

The encoder transforms the input samples into a latent representation  $\mathbf{z} \in \mathbb{R}^K$ . The memory  $M \in \mathbb{R}^{N \times K}$  is composed of  $N$  memory items  $m \in \mathbb{R}^K$ , each with a fixed dimension of  $K$ . Within FTD-MAE, the value of  $K$  is different for each branch. The attention addressing vector  $\mathbf{p} \in \mathbb{R}^N$  is calculated as the inner product of the latent representation and each memory item, intending to gauge the match between the latent representation and each individual memory unit. *Softmax* is then applied to the match scores between each memory item and the latent representation, yielding similarity distances that are key to capturing important characteristics of normal samples in memory. The formulation for the similarity distance associated with each memory unit is presented in (14).

$$p_i = \text{Softmax}(\mathbf{z}m_i^T) = \frac{\exp(\mathbf{z}m_i^T)}{\sum_{j=1}^N \exp(\mathbf{z}m_j^T)} \quad (14)$$

where  $p_i$  denotes the  $i$ -th attention addressing vector, and  $m_i$  denotes the  $i$ -th memory item.

A higher value of  $p_i$  means that the addressing vector corresponding to that value is more similar to the normal sample, and a lower value of  $p_i$  means less similarity. Using these addressing vectors treated with *Softmax* to reconstruct the latent representations  $\hat{\mathbf{z}}$  with memory will make the latent representations closer to the normal samples. The expression is shown in (15).

$$\hat{\mathbf{z}} = \mathbf{p}M = \sum_{i=1}^N p_i m_i \quad (15)$$

While backpropagation for parameter updates may reduce the magnitude of addressing vectors with low relevance to normal samples, when the number of addressing vectors hits a certain size, even low-magnitude addressing vectors can still reconstruct abnormal samples through linear combinations. To address this, we incorporate the addressing adjustment technique proposed by Gong et al [37], as shown in (16), which limit the size of the addressing vector  $p_i$ .

$$\hat{p}_i = \frac{\max(p_i - \lambda, 0) \cdot p_i}{|p_i - \lambda| + \epsilon} \quad (16)$$

where  $\lambda$  is a sparsity threshold that allows  $p_i$  less than  $\lambda$  to become 0. In order to prevent the denominator from being zero, a positive number  $\epsilon$  nearly equal to zero is added. With the addition of the addressing correction technique, (15) needs to be modified to (17).

$$\hat{\mathbf{z}} = \hat{\mathbf{p}}M = \sum_{i=1}^N \hat{p}_i m_i \quad (17)$$

### 3.4 Anormal State Detection

#### 3.4.1 Loss Function

In the course of training, the frequency domain and time domain branches are individually trained, with the training set comprising only normal samples. The objective of training is to have the reconstructed samples closely approximate the normal samples; therefore, we employ mean squared error as the loss function. The formulae are shown in (18) and (19).

$$\mathcal{L}_s(x_s, \hat{x}_s) = \frac{1}{n} \sum_{j=1}^n (x_{s_j} - \hat{x}_{s_j})^2 \quad (18)$$

$$\mathcal{L}_{img}(x_{img}, \hat{x}_{img}) = \frac{1}{d} \sum_{j=1}^d (x_{img_j} - \hat{x}_{img_j})^2 \quad (19)$$

where  $x_s$  is the input time series and  $\hat{x}_s$  is the reconstructed time series.  $n$  denotes the number of time points in the time series.  $x_{img}$  is the input image and  $\hat{x}_{img}$  is the reconstructed image.  $d$  denotes to the number of pixel points in the image.

#### 3.4.2 Weighted Abnormal Score

Faced with various kinds of data, the performance of the frequency domain and time domain branches can differ. When the input data has low frequency complexity, the time domain branch becomes more sensitive to amplitude fluctuations at individual time points; whereas if the input data has high frequency complexity, the frequency domain branch is better at capturing the time series' frequency characteristics. Hence, assessing the frequency complexity of the input time series is crucial. In this study, we utilize Spectral Entropy as the metric for frequency complexity. Initially, the Fast Fourier transform of the signal  $x(T)$  is computed to

obtain its frequency representation denoted as  $S(\mathbf{f})$ ,  $\mathbf{f} \in \{0,1,2, \dots, T-1\}$ .  $T$  denotes the total number of time points.

$$S(\mathbf{f}) = \text{FFT}\{x(T)\} = \sum_{t=0}^{T-1} x(t) \cdot e^{-\frac{2\pi i}{T} \mathbf{f} t}, \mathbf{f} \in \{0,1,2, \dots, T-1\} \quad (20)$$

where  $\text{FFT}\{\cdot\}$  refers to the Fast Fourier transform.  $i$  is an imaginary unit.

Calculate the magnitude  $M(\mathbf{f})$  of the  $f$ -th frequency, followed by normalization in the frequency dimension to obtain  $p(\mathbf{f})$ .

$$M(\mathbf{f}) = |S(\mathbf{f})| \quad (21)$$

$$p(\mathbf{f}) = \frac{M(\mathbf{f})}{\sum_{i=0}^{T-1} M(i)} \quad (22)$$

The weighting factor  $\theta$  can be calculated by substituting the normalized frequencies into (23).

$$\theta = - \sum_{\mathbf{f}=0}^{T-1} p(\mathbf{f}) \log_2(p(\mathbf{f}) + \epsilon) \quad (23)$$

where  $\theta$  denotes the Spectral Entropy, which is a scalar whose larger value indicates a higher frequency complexity of the input signal.  $\epsilon$  is a very small positive number.

Although the weights of the image reconstruction loss  $\mathcal{L}_{img}$  and the sequence reconstruction loss  $\mathcal{L}_s$  can be controlled by  $\theta$ , the size of the two losses themselves affects the weights, so an additional parameter  $\sigma$  is needed to scale the two weights to a close range.

$$\sigma = \frac{\mathcal{L}_{img}^*}{\mathcal{L}_s^*} \quad (24)$$

where  $\mathcal{L}_{img}^*$  denotes the final threshold of the frequency domain branch of the model selected on the validation set.  $\mathcal{L}_s^*$  refers to the final threshold selected on the validation set for the time domain branch of the model. The final abnormal score  $\varphi$  is shown in (25).

$$\varphi = \theta \mathcal{L}_{img} + \sigma \mathcal{L}_s \quad (25)$$

## 4. Experiment Results

### 4.1 Datasets

In this paper, the Condition Monitoring of Hydraulic Systems (CMHS) dataset was selected for temperature data [38] and the Case Western Reserve University (CWRU) bearing dataset was selected for vibration data [39].

The CMHS was obtained by means of a hydraulic test bed. The test bench consists of a primary working circuit connected through a tank and a secondary cooling and filtering circuit, where the system cycle repeats a constant load cycle (duration 60 seconds) and measures

process values such as pressure, volume flow and temperature, while quantitatively changing the state of four hydraulic components (coolers, valves, pumps and accumulators) [38]. In this study, we choose data where the ‘stability flag’ is 1 and aside from the cooler, all other components are largely normal, to represent abnormal states, and we consider data where the cooler is functioning normally and the ‘stability flag’ is 0 as representing normal states. Finally, a portion of the data is randomly selected for experimentation.

In terms of vibration data, Case Western Reserve University utilized a 2-horsepower Reliance electric motor for collecting acceleration data at different positions relative to the bearings of the motor. Artificial faults were induced on the inner raceway, rolling elements, and outer raceway with diameters ranging from 0.007 to 0.040 inches. The faulty bearings were then reinstalled into the testing motor and vibration data were logged when the motor was under loads ranging from 0 to 3 horsepower and with rotational speeds from 1797 to 1720 RPM [39]. In this study, data specifically came from tests under a 3-horsepower motor load, with a fault size of 0.021 inches. Details of the dataset are shown in **Table 1**.

**Table 1.** Description of the dataset

Dataset	Abnormal Ratio	Quantities	Length
CMHS	33.19%	2205	60
CWRU	42.92%	664	1280

## 4.2 Metrics

In this study, we used Accuracy and F1-Score as metrics to evaluate the performance of the model. Accuracy is the proportion of the number of samples that are correctly classified to the total number of samples. This metric provides an intuitive depiction of the model's level of accuracy in classifying the entire dataset. The expression is shown in (26).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{N}} \quad (26)$$

where TP refers to True Positives, TN refers to True Negatives, and N refers to the total sample size. Nonetheless, the CMHS and CWRU datasets have certain data imbalances. Relying solely on accuracy as an evaluation metric is insufficiently rigorous, so we have also included an additional metric, F1 Score.

The F1-Score is the harmonic mean of Precision and Recall and is a composite metric. It is used in cases where there is a category imbalance in the dataset and can balance the effects of FP (False Positives) and FN (False Negatives).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (27)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (28)$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (29)$$

### 4.3 Experiment Environment Settings

On the CMHS dataset, the FTD-MAE frequency domain branch has an STFT window size of 3, a per-minute sampling frequency of 60, and produces 100x100 images. The encoder has three convolutional layers (The convolution kernel is 3 and the step size is 2), while the decoder has four linear layers (The number of neurons is 128, 256, 512,  $100^2$  respectively). The size of the memory module is 2000 with a sparsity threshold of 0.0025. Only normal samples are included in the training and validation sets. The ratio of positive samples in the training, validation, and test sets is 7:1:2. The training batch size is 32, the learning rate is  $2e-4$ , and the number of epochs is 200. The time domain branch of FTD-MAE uses two linear layers for both encoding (The number of neurons is 64, 128 respectively) and decoding (The number of neurons is 128, 60 respectively), with a memory module size of 100 and a sparsity threshold of 0.008. The data partitioning method is identical to that of the frequency domain branch. The batch size for training is 8, the learning rate is  $1e-4$ , and the epoch is 50.

On the CWRU dataset, the frequency domain branch of FTD-MAE uses an STFT window size of 8, with a sampling rate of 1280 per minute, generating images of size  $256*256$ . The encoder and decoder are the same as CHMS, but the number of neurons in the last layer of the decoder is  $256^2$ . The size of the memory module is 2000, with a sparsity threshold of 0.0025. Only normal samples are included in the training and validation sets. The ratio of positive samples in the training, validation, and test sets is 7:1:2. The training batch size is 32, with a learning rate of  $1e-4$ , and the epoch is 50. The encoder and decoder in the FTD-MAE time domain branch consist of two linear layers. The number of encoder neurons is 400, 800 respectively. The number of decoder neurons is 800, 128 respectively. The storage module size is 500, sparsity threshold is 0.0025, and the dataset is partitioned in the same manner as the frequency domain branch. The training batch size is 8, the learning rate is  $1e-4$ , and the number of epochs is 50.

### 4.4 Comparison of FTD-MAE with Representative Models

In this experiment, we compare FTD-MAE with six other representative models, including AE, VAE, Isolation Forest, PCA (Principal Component Analysis) [40], OC-SVM [41] and LSTM (Long Short-Term Memory) [42]. The encoders and decoders of AE, VAE and the time domain branch of FTD-MAE are the same. ACC and F1-Score results of FTD-MAE as compared to other representative models are displayed in Table 2. It can be seen that the ACC and F1-Score of FTD-MAE on the CMHS dataset are 0.9734 and 0.9668, respectively, which are 2.66% and 3.75% higher than the best representative model. On the CWRU dataset, FTD-MAE records an ACC and F1-Score of 0.9917 and 0.9948, surpassing the best representative model by 39.61% and 33.76% respectively.

**Table 2.** Performance of FTD-MAE and six representative models

Model	CMHS		CWRU	
	ACC	F1	ACC	F1
Isolation Forest	0.9263	0.9016	0.1773	0.2774
PCA	0.9427	0.9251	0.1607	0.2628
OC-SVM	0.8486	0.8348	0.2881	0.4473
LSTM	0.9468	0.9293	0.4792	0.5104
AE	0.9448	0.9256	0.5956	0.6572
VAE	0.9366	0.9136	0.4626	0.4921
FTD-MAE	<b>0.9734</b>	<b>0.9668</b>	<b>0.9917</b>	<b>0.9948</b>

#### 4.5 Applicability Experiments for the Proposed Module

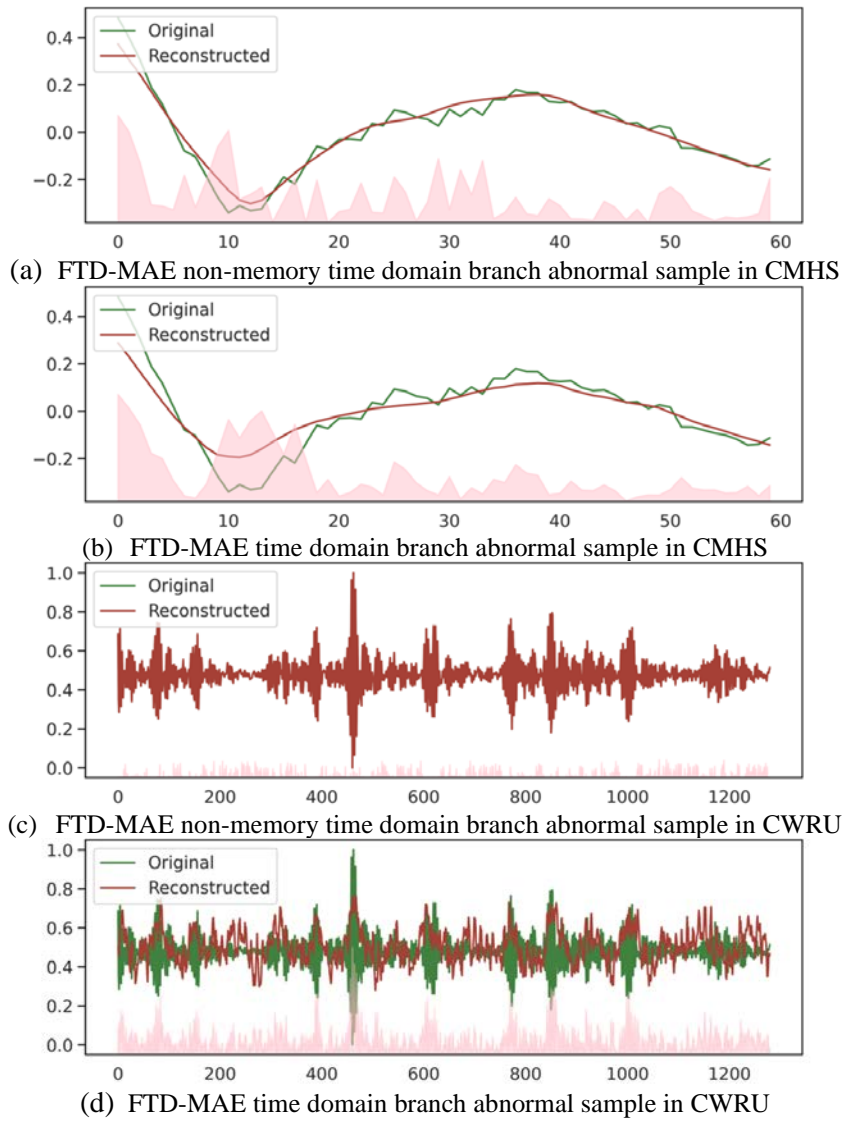
FTD-MAE-F means that FTD-MAE uses only frequency domain branches for inference. FTD-MAE-T means that FTD-MAE uses only time domain branches for inference. The non sparse means that sparsity threshold is not used in the model's memory module to restrict memory items. The non memory means that no memory module is used between the encoder and decoder, it is equivalent to regular AE.

From [Table 3](#), it can be seen that for CHMS data with lower frequency complexity, FTD-MAE-T outperforms FTD-MAE-F. This is because when frequency complexity is low, abnormalities in the data are more likely to occur in short-term fluctuations or sudden changes, rather than distributed across the entire frequency range. FTD-MAE-T excels in capturing these short-term abnormal shifts since it analyzes data directly within its time domain, allowing for more accurate pinpointing and identification of local abnormal. Conversely, FTD-MAE-F focuses mainly on features in the frequency domain, making it more suitable for handling data with high-frequency complexity or obvious periodicity. In such data, abnormalities are likely to manifest as sudden changes in frequency components, therefore, FTD-MAE-F performs better in CWRU data with high-frequency complexity. Without a sparsity threshold, the performance of both FTD-MAE-F and FTD-MAE-T declines, with FTD-MAE-T experiencing the most significant drop in performance on the CWRU dataset. This is because too many memory items can better reconstruct abnormal samples. FTD-MAE's overall performance suffers when it lacks a memory module, because AE's generalization is strong. Without a memory module to regulate reconstruction, even abnormal samples can be effectively reconstructed.

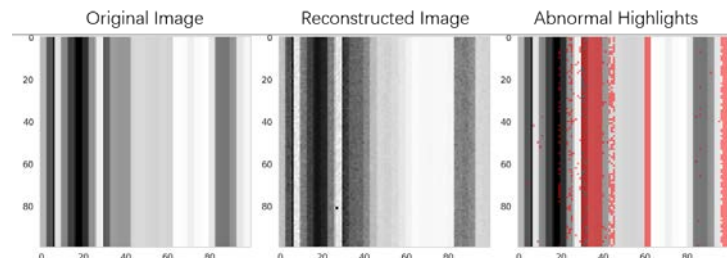
**Table 3.** Various experimental results applying FTD-MAE with the datasets

Model	CMHS		CWRU	
	ACC	F1	ACC	F1
FTD-MAE-F	0.9387	0.9197	0.9834	0.9896
FTD-MAE-T	0.9611	0.9491	0.6898	0.7596
FTD-MAE-F-non sparse	0.9141	0.8833	0.9695	0.9811
FTD-MAE-T- non sparse	0.9591	0.9459	0.4598	0.4909
FTD-MAE- non sparse	0.9693	0.9614	0.9668	0.9792
FTD-MAE-F-non memory	0.9243	0.8986	0.9806	0.9879
FTD-MAE-T-non memory	0.9448	0.9256	0.5956	0.6572
FTD-MAE-non memory	0.9468	0.9293	0.9806	0.9879
FTD-MAE	<b>0.9734</b>	<b>0.9668</b>	<b>0.9917</b>	<b>0.9948</b>

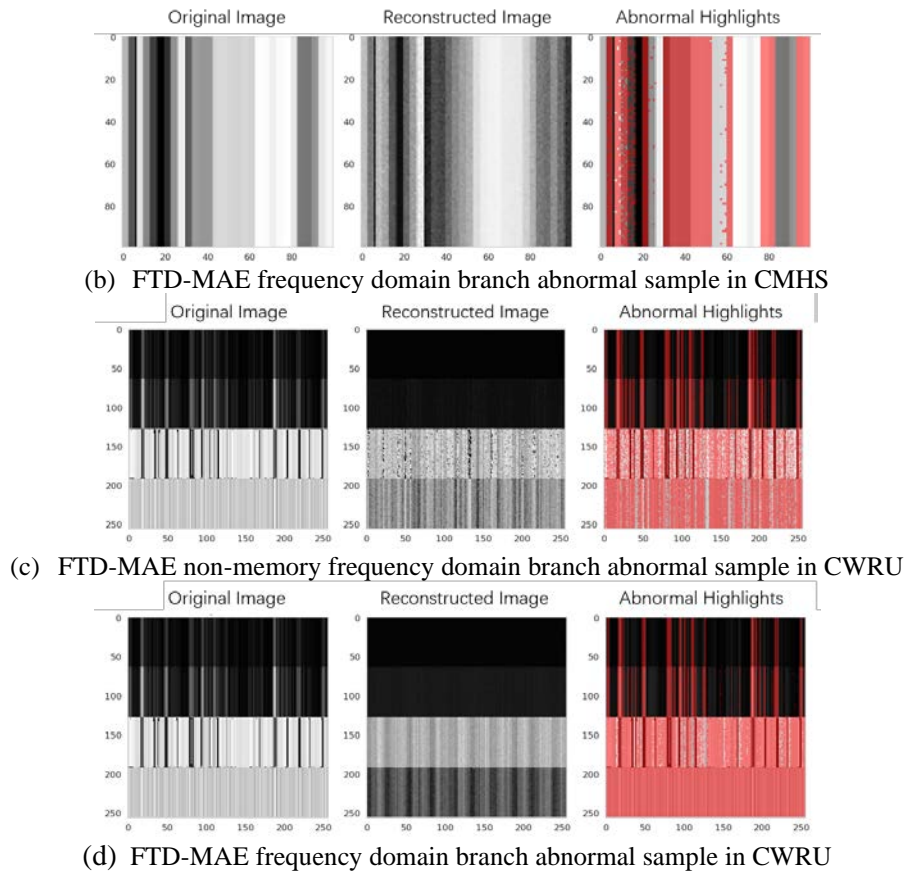
[Fig. 5](#) and [Fig. 6](#) illustrate the difference in abnormal sample reconstruction between the frequency and time domain branches of FTD-MAE, both with and without the memory module. [Fig. 5](#) only utilizes the time domain branch of FTD-MAE. It can be seen from the figure that without the memory module, the model can generalize well across the abnormal samples in both datasets, especially evident in the vibration data. On the other hand, adding a memory module leads to significant differences. [Fig. 6](#) only employs the frequency domain branch of FTD-MAE. Without the memory module, this model overgeneralizes the CMHS data, resulting in too small reconstruction errors for the abnormal samples. After adding the memory module, the reconstruction errors for both datasets increase correspondingly.



**Fig. 5.** Comparison of reconstructed abnormal samples in FTD-MAE time domain branch with and without memory module.



(a) FTD-MAE non-memory frequency domain branch abnormal sample in CMHS



**Fig. 6.** Comparison of reconstructed abnormal samples in FTD-MAE frequency domain branch with and without memory module.

In summary, FTD-MAE-T excels at capturing time domain features, while FTD-MAE-F specializes in capturing frequency characteristics. Together, they make the model robust to different types of data. The memory module and sparsity threshold jointly constrain the model's generalization, thereby further enhancing its robustness.

#### 4.6 Temperature Data Preprocessing

Since the trends of individual temperature time series are closely aligned, the model could potentially fit the data trends at large time scales, neglecting variations at smaller scales. Therefore, we apply trend attenuation in processing these temperature time series. The frequency domain branch doesn't attenuate the trend, as its window function lowers time resolution and makes it challenging to detect short-term fluctuations. MinMax-Scale corresponds to (2), while Detrend corresponds to (8). We compared the two preprocessing methods on the CMHS temperature dataset, and the results are shown in [Table 4](#). The frequency domain branch had a higher score when data was processed using MinMax-Scale, while the time domain branch scored higher when data was processed using Detrend.

#### 4.7 Time-frequency Map Transformation Method

In this section, we compare the performance of three time-frequency map transformation methods, wavelet transform, Hilbert-Huang transform (HHT) and STFT, on the FTD-MAE



**Table 4.** Temperature data preprocessing comparison

Preprocessing	FTD-MAE-T		FTD-MAE-F		FTD-MAE-T-non memory		FTD-MAE-F-non memory	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1
MinMax-Scale	0.9162	0.8864	<b>0.9387</b>	<b>0.9197</b>	0.8998	0.8588	<b>0.9243</b>	<b>0.8986</b>
Detrend	<b>0.9611</b>	<b>0.9491</b>	0.9039	0.8698	<b>0.9448</b>	<b>0.9256</b>	0.8957	0.8571

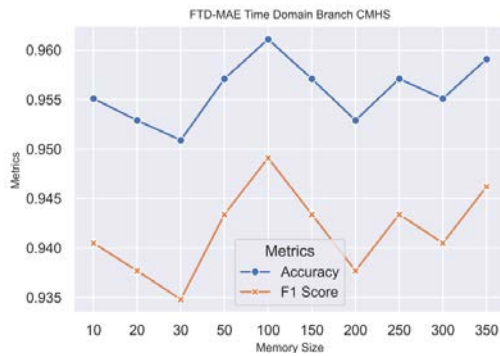
frequency domain branch. The result is shown in **Table 5**. From the table, it's apparent that on complex frequency vibration dataset, STFT, wavelet transform, and HHT perform similarly. However, for temperature dataset with trends, wavelet transform and HHT show significantly lower ACC than STFT. This is likely because, despite their variable time-frequency resolution, wavelet transform and HHT may struggle to maintain a consistent time-frequency resolution like STFT in signals with long-term trends, impacting result accuracy.

**Table 5.** Performance of FTD-MAE frequency domain branch on two datasets with different time-frequency map transformation methods

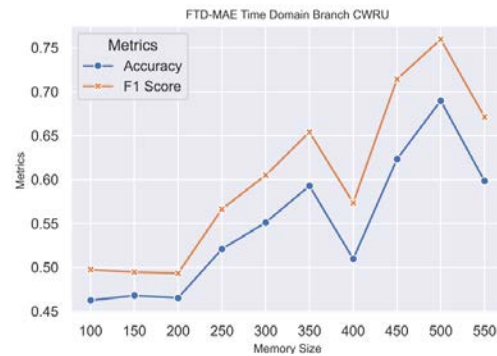
Methods	CMHS		CWRU	
	ACC	F1	ACC	F1
wavelet transform	0.8299	0.9072	0.9815	0.9866
HHT	0.8402	0.9131	<b>0.9842</b>	<b>0.9901</b>
STFT	<b>0.9387</b>	<b>0.9197</b>	0.9834	0.9896

## 4.8 Memory Size

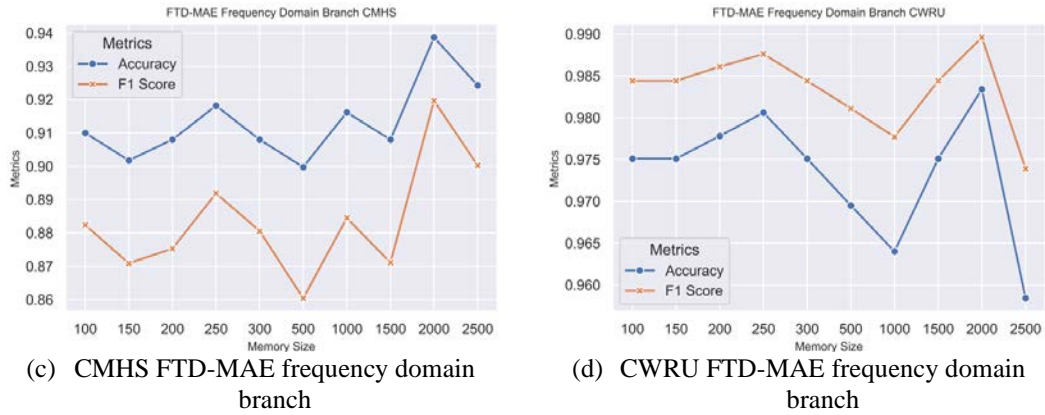
The size of the memory module affects the reconstruction quality. If the memory module is too small, the model will not be effective in reconstructing any samples; if it is too large, it may result in better reconstruction of abnormal samples. Hence, we evaluated the size of each memory module, as shown in **Fig. 7**. The selection of the memory size is highly correlated with the data complexity. For temperature time series with low complexity, only a small memory is needed to achieve better reconstruction of normal samples and suppression of abnormal samples (**Fig. 7 a**). Relative to the more complex vibrational time series, the optimal memory size reaches 500, far exceeding the temperature time series (**Fig. 7 b**). Unlike time series, image data after STFT is more complex, so the optimal memory size is 2000 (**Fig. 7 c and d**).



(a) CMHS FTD-MAE time domain branch



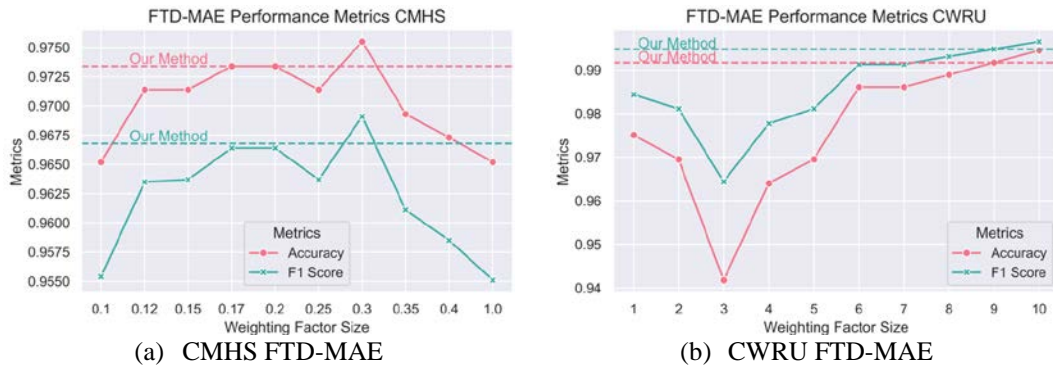
(b) CWRU FTD-MAE time domain branch



**Fig. 7.** Performance of each branch of FTD-MAE on CMHS and CWRU datasets with different memory sizes.

#### 4.9 Weighting Factor

Given the differing frequency complexities of time series, the weights for the FTD-MAE time domain and frequency domain branches should also vary. Manually setting thresholds is often laborious and may not be reliable when the frequency complexity of the input data is not known. We use Spectral Entropy as the weighting coefficient (the higher the coefficient, the higher the frequency complexity) to dynamically adjust the weight between the two branches, thereby improving the model's robustness. To validate the effectiveness of this method, we have manually configured some scaling coefficients for comparison, as illustrated in **Fig. 8**. The Spectral Entropy-weighted coefficients approach the highest accuracy on both datasets.



**Fig. 8.** Performance of FTD-MAE on CMHS and CWRU datasets with different weighting factor sizes.

## 5. Conclusion

Due to the waveform similarity in temperature time series and the high-frequency complexity in vibration time series within industrial equipment, we propose a new FTD-MAE model. The model is composed of frequency domain and time domain branches, effectively capturing the frequency characteristics in high-frequency vibration data and short-term fluctuations in low-frequency temperature data. A memory module between the branch encoder and decoder enables FTD-MAE to learn normal data patterns, minimizing over-generalization on abnormal

samples. Additionally, the weights of the branches can be dynamically adjusted using Spectral Entropy, thereby adapting to time series with different frequency complexities and reducing computational costs. After rigorous testing, FTD-MAE demonstrated remarkable performance on the CMHS and CWRU datasets, with an average ACC of 0.9826 and an average F1 score of 0.9808, respectively. When compared to the top-performing model, there was an enhancement of 0.2114 in average ACC and 0.1876 in average F1 score.

Although FTD-MAE has achieved good performance, its performance improvement is more dependent on each branch itself. The association of two branches is only in the inference part of the model, which balances the decisions of the branches through the weight factor. In the future, we will utilize the features of each branch in the training process, perform feature fusion, and eventually reconstruct a time series directly.

### Acknowledgement

This research was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-002)

### References

- [1] P. K. Malik, R. Sharma, R. Singh, A. Gehlot, S. C. Satapathy, W. S. Alnumay, D. Pelusi, U. Ghosh, and J. Nayak, "Industrial Internet of Things and its Applications in Industry 4.0: State of The Art," *Computer Communications*, vol. 166, pp. 125–139, Jan. 2021. [Article \(CrossRef Link\)](#)
- [2] G. Toh and J. Park, "Review of vibration-based structural health monitoring using deep learning," *Applied Sciences*, vol. 10(5), p. 1680, Mar. 2020. [Article \(CrossRef Link\)](#)
- [3] M. Alarcón, F. M. Martínez-García, and F. C. Gómez de León Hijes, "Energy and maintenance management systems in the context of industry 4.0. Implementation in a real case," *Renewable and Sustainable Energy Reviews*, vol. 142, p. 110841, May 2021. [Article \(CrossRef Link\)](#)
- [4] H. Dhiman, D. Deb, S. M. Muyeen, and I. Kamwa, "Wind Turbine Gearbox Anomaly Detection Based on Adaptive Threshold and Twin Support Vector Machines," *IEEE Transactions on Energy Conversion*, vol. 36, no. 4, pp. 3462–3469, Dec. 2021. [Article \(CrossRef Link\)](#)
- [5] A. K. Pathak, S. Saguna, K. Mitra, and C. Ahlund, "Anomaly Detection using Machine Learning to Discover Sensor Tampering in IoT Systems," in *Proc. of ICC 2021 - IEEE International Conference on Communications*, Montreal, QC, Canada, 2021. [Article \(CrossRef Link\)](#)
- [6] J. Gao, X. Song, Q. Wen, P. Wang, L. Sun, and H. Xu, "RobustTAD: Robust Time Series Anomaly Detection via Decomposition and Convolutional Neural Networks," *Cornell University - arXiv, Cornell University - arXiv*, Feb. 2020. [Article \(CrossRef Link\)](#)
- [7] M. Thill, W. Konen, H. Wang, and T. Bäck, "Temporal convolutional autoencoder for unsupervised anomaly detection in time series," *Applied Soft Computing*, vol. 122, p. 107751, Nov. 2021. [Article \(CrossRef Link\)](#)
- [8] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proc. of ACM SIGMOD Record*, pp. 427–438, Jun. 2000. [Article \(CrossRef Link\)](#)
- [9] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in *Proc. of 2008 Eighth IEEE International Conference on Data Mining*, Pisa, Italy, 2008. [Article \(CrossRef Link\)](#)
- [10] A. Toshniwal, K. Mahesh, and J. R., "Overview of Anomaly Detection techniques in Machine Learning," in *Proc. of 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, 2020. [Article \(CrossRef Link\)](#)
- [11] O. I. Provotar, Y. M. Linder, and M. M. Veres, "Unsupervised Anomaly Detection in Time Series Using LSTM-Based Autoencoders," in *Proc. of 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT)*, Kyiv, Ukraine, 2019. [Article \(CrossRef Link\)](#)

- [12] G. Spigler, "Denoising Autoencoders for Overgeneralization in Neural Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 4, pp. 998–1004, Apr. 2020. [Article \(CrossRef Link\)](#)
- [13] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation*, vol. 19, pp. 297–301, 1965. [Article \(CrossRef Link\)](#)
- [14] F. Jurado and J. R. Saenz, "Comparison between discrete STFT and wavelets for the analysis of power quality events," *Electric Power Systems Research*, vol. 62, no. 3, pp. 183–190, Jul. 2002. [Article \(CrossRef Link\)](#)
- [15] Z. Wang, J. Zhou, J. Wang, W. Du, J. Wang, X. Han, and G. He, "A Novel Fault Diagnosis Method of Gearbox Based on Maximum Kurtosis Spectral Entropy Deconvolution," *IEEE Access*, vol. 7, pp. 29520–29532, Jan. 2019. [Article \(CrossRef Link\)](#)
- [16] A. A. Cook, G. Misirli, and Z. Fan, "Anomaly Detection for IoT Time-Series Data: A Survey," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6481–6494, Jul. 2020. [Article \(CrossRef Link\)](#)
- [17] X. Xia, X. Pan, N. Li, X. He, L. Ma, X. Zhang, and N. Ding, "GAN-based Anomaly Detection: A Review," *Neurocomputing*, vol. 493, pp. 497–535, Jul. 2022. [Article \(CrossRef Link\)](#)
- [18] M. Hassan, M. Rehmani, and J. Chen, "Anomaly Detection in Blockchain Networks: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 289–318, 2023. [Article \(CrossRef Link\)](#)
- [19] W. A. Chaovalitwongse, Y.-J. Fan, and R. C. Sachdeo, "On the Time Series K-Nearest Neighbor Classification of Abnormal Brain Activity," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 37, no. 6, pp. 1005–1016, Nov. 2007. [Article \(CrossRef Link\)](#)
- [20] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," in *Proc. of ACM SIGMOD Record*, pp. 93–104, Jun. 2000. [Article \(CrossRef Link\)](#)
- [21] J. Baihong, C. Yuxin, L. Dan, P. Kameshwar, and S.-V. Alberto, "A One-Class Support Vector Machine Calibration Method for Time Series Change Point Detection," in *Proc. of IEEE Conference Proceedings, IEEE Conference Proceedings*, Jan. 2019. [Article \(CrossRef Link\)](#)
- [22] S. Zhong, S. Fu, L. Lin, X. Fu, Z. Cui, and R. Wang, "A novel unsupervised anomaly detection for gas turbine using Isolation Forest," in *Proc. of 2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*, San Francisco, CA, USA, 2019. [Article \(CrossRef Link\)](#)
- [23] V. Aggarwal, V. Gupta, P. Singh, K. Sharma, and N. Sharma, "Detection of Spatial Outlier by Using Improved Z-Score Test," in *Proc. of 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, 2019. [Article \(CrossRef Link\)](#)
- [24] F. E. Grubbs, "Sample Criteria for Testing Outlying Observations," *The Annals of Mathematical Statistics*, vol. 21, no. 1, pp. 27–58, 1950. [Article \(CrossRef Link\)](#)
- [25] K. Choi, J. Yi, C. Park, and S. Yoon, "Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines," *IEEE Access*, vol. 9, pp. 120043–120065, Jan. 2021. [Article \(CrossRef Link\)](#)
- [26] P. Matias, D. Folgado, H. Gamboa, and A. Carreiro, "Robust Anomaly Detection in Time Series through Variational AutoEncoders and a Local Similarity Score," in *Proc. of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies, Online Streaming, --- Select a Country ---*, 2021. [Article \(CrossRef Link\)](#)
- [27] H. Liu, J. Zhou, Y. Zheng, W. Jiang, and Y. Zhang, "Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders," *ISA Transactions*, vol. 77, pp. 167–178, Jun. 2018. [Article \(CrossRef Link\)](#)
- [28] B. Du, X. Sun, J. Ye, K. Cheng, J. Wang, and L. Sun, "GAN-Based Anomaly Detection for Multivariate Time Series Using Polluted Training Set," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 12, pp. 12208–12219, 2023. [Article \(CrossRef Link\)](#)
- [29] D. J. Bartholomew, G. E. P. Box, and G. M. Jenkins, "Time Series Analysis Forecasting and Control," *Operational Research Quarterly (1970-1977)*, vol. 22, no. 2, p. 199, Jun. 1971. [Article \(CrossRef Link\)](#)
- [30] C. Chakrabarti, M. Vishwanath, and R. M. Owens, "Architectures for wavelet transforms: A survey," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 14, no. 2, pp. 171–192, Nov. 1996. [Article \(CrossRef Link\)](#)

- [31] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," in *Proc. of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, pp. 903–995, Mar. 1998. [Article \(CrossRef Link\)](#)
- [32] A. Gupta, P. Mishra, C. Pandey, U. Singh, C. Sahu, and A. Keshri, "Descriptive statistics and normality tests for statistical data," *Annals of Cardiac Anaesthesia*, vol. 22, no. 1, pp. 67-72, Jan. 2019. [Article \(CrossRef Link\)](#)
- [33] R. G. Baraniuk, P. Flandrin, A. J. E. M. Janssen, and O. J. J. Michel, "Measuring time-frequency information content using the Renyi entropies," *IEEE Transactions on Information Theory*, vol. 47, no. 4, pp. 1391–1409, May 2001. [Article \(CrossRef Link\)](#)
- [34] J. Weston, S. Chopra, and A. Bordes, "Memory networks," in *Proc. of International Conference on Learning Representations (ICLR)*, 2015. [Article \(CrossRef Link\)](#)  
S. Sukhbaatar, a. szlam, J. Weston, and R. Fergus, "End-to-end memory networks," in *Proc. of Neural Information Processing Systems*, 2015. [Article \(CrossRef Link\)](#)
- [35] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher, "Ask Me Anything: Dynamic Memory Networks for Natural Language Processing," *arXiv: Computation and Language*, *arXiv: Computation and Language*, Jun. 2015. [Article \(CrossRef Link\)](#)
- [36] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. Van Den Hengel, "Memorizing Normality to Detect Anomaly: Memory-Augmented Deep Autoencoder for Unsupervised Anomaly Detection," in *Proc. of 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 2019. [Article \(CrossRef Link\)](#)
- [37] N. Helwig, E. Pignanelli, A. Schtze, "Condition monitoring of hydraulic systems," *UCI Machine Learning Repository*, 2018. [Online]. Available: <https://archive.ics.uci.edu/dataset/447/condition+monitoring+of+hydraulic+systems>
- [38] Case Western Reserve University Bearing Data Center, Dec. 2019. [Online]. Available: <https://engineering.case.edu/bearingdatacenter>
- [39] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of PCA for traffic anomaly detection," in *Proc. of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, San Diego California USA, pp. 109-120, 2007. [Article \(CrossRef Link\)](#)
- [40] B. Schölkopf, RobertC. Williamson, AlexanderJ. Smola, J. Shawe-Taylor, and J. Platt, "Support Vector Method for Novelty Detection," *Neural Information Processing Systems, Neural Information Processing Systems*, Nov. 1999. [Article \(CrossRef Link\)](#)
- [41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Article \(CrossRef Link\)](#)



**Haoyi Zhong** is currently pursuing an M.S. degree in computer engineering with the Chonnam National University, Yeosu, South Korea. His research interests include machine learning, computer vision and IoT.



**Yongjiang Zhao** received M.S. degrees from the Chonnam National University, Yeosu, South Korea, in 2021. He is currently pursuing an Ph.D. degree in computer engineering with the Chonnam National University, Yeosu, South Korea. His research interests include machine learning and reinforcement learning.



**ChangGyoon Lim** received the Ph.D. degree from the Department of Computer Engineering, Wayne State University, USA, in 1997. Since September 1997, he has been working for major in computer engineering with Chonnam National University, Yeosu, South Korea, as a professor. He was the Director of Home Robot Center, Gwangju Techno Park. He also leads various research projects of his interest areas. Current areas of interest include machine learning, energy optimization, IoT, big data, virtual power plants, digital twins, and blockchain. He is a member of Korean Society for Internet Information. Simultaneously, he acts as the committee member of several public and private regional institutions. He currently serves as Chairman of the Board of Coding Insight and Chairman.