

## Article

# SSA-SL Transformer for Bearing Fault Diagnosis under Noisy Factory Environments

Seoyeong Lee and Jongpil Jeong \* 

Department of Smart Factory Convergence, Sungkyunkwan University, Suwon 16419, Korea; smaxtoday@naver.com

\* Correspondence: jpjeong@skku.edu

**Abstract:** Among the smart factory studies, we describe defect detection research conducted on bearings, which are elements of mechanical facilities. Bearing research has been consistently conducted in the past; however, most of the research has been limited to using existing artificial intelligence models. In addition, previous studies assumed the factories situated in the bearing defect research were insufficient. Therefore, a recent research was conducted that applied an artificial intelligence model and the factory environment. The transformer model was selected as state-of-the-art (SOTA) and was also applied to bearing research. Then, an experiment was conducted with Gaussian noise applied to assume a factory situation. The swish-LSTM transformer (SI transformer) framework was constructed by redesigning the internal structure of the transformer using the swish activation function and long short-term memory (LSTM). Then, the data in noise were removed and reconstructed using the singular spectrum analysis (SSA) preprocessing method. Based on the SSA-SI transformer framework, an experiment was performed by adding Gaussian noise to the Case Western Reserve University (CWRU) dataset. In the case of no noise, the SI transformer showed more than 95% performance, and when noise was inserted, the SSA-SI transformer showed better performance than the comparative artificial intelligence models.

**Keywords:** bearing fault diagnosis; singular spectrum analysis; transformer; under noisy factory environments



**Citation:** Lee, S.; Jeong, J. SSA-SL Transformer for Bearing Fault Diagnosis under Noisy Factory Environments. *Electronics* **2022**, *11*, 1504. <https://doi.org/10.3390/electronics11091504>

Academic Editors: Hamed Habibi, Afef Fekih, Simani Silvio and Amirmehdi Yazdani and Adam Glowacz

Received: 22 March 2022

Accepted: 30 April 2022

Published: 7 May 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Among smart factory-related studies, demand for machine condition monitoring research is increasing. This is because increased downtime due to machine failures can cause enormous economic losses. Therefore, the demand for better methods of diagnostic monitoring of machine conditions continues to ensure the machine operates stably. The method described in this paper mainly uses a deep-learning-based artificial intelligence methodology. This paper studies a deep learning approach to diagnose bearing defects among various machines and components [1–5].

The most important part of the machines in the factory is the motor. A defect in the motor, which is an electrical device, can cause fatal damage to the factory. Among the defects in the motor, defects in the bearing appear most frequently. Motor defects are found mainly due to overcurrent. The inspection of bearing defects caused by this overcurrent is a major concern in this paper. Most simply, a bearing is a device that supports an axis. Upon further refinement, a bearing is a mechanical element that supports the rotational axis and reciprocating movement in a certain position while being used in a machine, resulting in accurate and smooth movement. That is, the bearing induces the shaft to rotate smoothly and serves to support the load, determines the position of the rotation shaft, and maintains the position even if the load changes. As such, bearings are a very important factor for mechanical facilities with axes. For this reason, previous studies have focused on the diagnosis of bearing abnormalities [6–10].

The following describes the flow of previous studies on bearing defect detection. In the beginning, statistical approaches or machine learning methodologies were mainly used based on bearing data. The classification process for defects was solved using various machine learning methods, such as support vector machine (SVM) and XGBoost [11–13]. However, as deep learning methods have emerged since the research of [14], a deep learning approach has been introduced in earnest to research bearing abnormalities. Since the introduction of deep learning, the following studies have been conducted. Artificial intelligence neural network models of the recurrent neural network (RNN) series were used as a method of dealing with time series data from bearing vibration data [15,16]. In an effort to improve performance of the artificial intelligence model, most of the studies converted one-dimensional data into two-dimensional data after preprocessing and approached it using the convolutional neural network (CNN) model [17].

However, research on the diagnosis of bearing defects confirmed that the previous directions were insufficient. There is the need to apply the latest trend of artificial intelligence models to research more diverse data preprocessing methods and address the lack of consideration, such as in actual factories. First, it was found that research on detecting bearing defects incorporating the latest trend of artificial intelligence modeling, the transformer model, was insufficient. The transformer model, which has shown overwhelming performance in the field of text generation [18], is now used in various fields such as image classification. Likewise, the transformer model shows overwhelming performance in other fields. Currently, several studies are known to have produced SOTA performance [19,20]. Second, vibration data to be used in research on the diagnosis of bearing defects can be preprocessed in various ways. Several studies have confirmed the use of preprocessing methods such as wavelet and fast Fourier transform (FFT) [14,21]. However, in many studies, only preprocessing techniques were used to ensure the classification accuracy. There is a need to research preprocessing techniques for processing noise. Third, most of the public bearing data did not contain noise mixed inside the data. However, the actual factory environment generates various obstacles in the process of data collection. Obstructive elements produce noise. Therefore, clean bearing data research may not be suitable for an actual factory environment. Previously, there was also a paper that evaluated the performance of the framework by adding noise to the data in consideration of the factory environment [22,23]. However, it did not show what noise and how much noise was inserted. Therefore, this paper was conducted to supplement the limitations of the existing studies mentioned.

The following are the contributions of this paper. First, we propose the SSA-SI transformer framework. SSA is a data preprocessing technique and SI transformer is an artificial intelligence model that has developed the existing vanilla transformer model. Singular spectrum analysis (SSA) is an algorithm that decomposes time series data into multiple subseries and then reconstructs the data [24,25]. SI transformer stands for swish-LSTM transformer. We used the swish activation function in the paper, and then designed and inserted the LSTM block into the feed-forward and self-attention internal linear layer within the encoder of the transformer. It was possible to hypothesize that SSA would be an appropriate method for bearing fault diagnostics in that it is an algorithm that considers the noise in time series data, and the SI transformer showed higher performance than the existing vanilla transformer. The second contribution is the assumption of the factory situation. Previous studies related to bearing abnormality detection have already demonstrated more than 95% classification-related accuracy. Therefore, this experiment was also predictable in that a high level of accuracy could be obtained. This figure was not considered realistic. This is because there are many variables in factories. This is because when data are collected through sensors in actual factories, many field studies say that they are suffering from noise generation. Therefore, in this work, we insert noise into clean data, then separate the degree of noise. The main goal is to see how robust the framework proposed in the paper is to noise. Engineers and follow-up researchers will be able to obtain more practical research results. A summary of the contributions is as follows.

1. We propose the SSA-SI transformer framework. We re-examine noise-resistant SSA preprocessing techniques and demonstrate the robust performance of existing SI transformer algorithms through various metrics.
2. Experiments were conducted assuming a realistic factory situation. The actual factory environment was implemented by mixing noise with the bearing dataset, and experiments were conducted. Engineers and researchers will be able to look at more realistic results through this paper.

This paper is composed as follows. Section 2 introduces previous bearing fault diagnostic studies, LSTM, SSA algorithm, and transformer artificial intelligence models. Section 3 describes the proposed SSA-SL transformer framework. Section 4 provides a description of the dataset and the experimental environment, how SSA deals with noise, a description of the swish activation function, and a description of the results of the three experiments.

## 2. Related Works

### 2.1. Bearing Fault Diagnosis

The process of bearing defect formation is mainly as follows. After microcracks occur inside the bearing, internal microcracks agglomerate and surface damage occurs. Subsequently, due to a lack of lubricant, contact between the bearing surfaces, the metal is damaged, or an abnormally excessive external force is applied to the bearing. Most defects appear in the outer race, inner race, or ball. Defects can be classified according to diameter. Previous studies on these bearing defects are as follows [6–10]. Many of the initial classifications proposed framework or more advanced research by using an autoencoder [26,27]. Other studies suggest methods of preprocessing bearing data to boost performance [14,21,28]. Attempts to transform one-dimensional data into two dimensions are also widely used. This paper shows high accuracy using the CNN artificial intelligence model after transforming one-dimensional bearing signal data into two dimensions [29]. There are also previous studies that improve classification performance by utilizing CNN artificial intelligence neural networks for 2D data images [17]. In addition to this, there are various methodologies for resolving noise in the bearing fault diagnosis process. There was a methodology that used empirical mode decomposition (EMD) to remove noise in the bearing fault diagnosis process [30]. In [31], we could refer to a method robust to noise through the second-order stochastic resonance method. In [32], it is suggested that MCKD, MOMEDA, and CYCBD can be used to more effectively extract characteristic factors for bearing defect detection.

### 2.2. SSA Algorithm

This paper seeks to present the singular spectrum algorithm (SSA) method, which is a relatively unnoticed preprocessing method [24,25]. SSA is a methodology for decomposing time series data into a simple form of components. SSA was developed as shown in Figure 1. In the decomposition process, data are run through embedding and singular vector decomposition (SVD) processes [33], and in the reconstruction process, they are run through grouping and averaging processes before ending.

The decomposition process is subjected to (a) embedding and (b) SVD processes in Figure 1. In the embedding process, one-dimensional time series data is converted into a trajectory matrix. This trajectory matrix is also called the Hankel matrix. Suppose there is a one-dimensional time series of length  $N$ . Let  $F = (f_0, \dots, f_N)$ . In the embedding process, it appears as the definition of the lagged version of time series  $F$ . The matrix depends on  $L$  ( $L \leq I/2$ ), which is the window length and is randomly determined. Let  $K = I - L + 1$ , the Hankel matrix, be defined as (1).

$$X = [X_1, \dots, X_K] = \begin{bmatrix} f_0 & f_1 & f_2 & \cdots & f_{K-1} \\ f_1 & f_2 & f_3 & \cdots & f_K \\ f_2 & f_3 & f_4 & \cdots & f_{K+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{L-1} & f_L & f_{L+1} & \cdots & f_T \end{bmatrix} \quad (1)$$

The trajectory matrix is a Hankel matrix, and all elements along the diagonal ( $i + j = \text{const}$ ) are symmetrically identical. The SVD process in (b) is as follows. In the SVD process, the Hankel matrix is changed to the sum of the two-dimensional bi-orthogonal elementary matrices (2).

$$X_i = X_1 + \cdots + X_K \quad (2)$$

The elementary matrix  $X_i$  is defined by  $X_i = s_i U_i V_i^T$ .  $s_i$  is the  $i$ th singular value of  $X$ ,  $U_i$  is the  $i$ th left singular vector of  $X$ , and  $V_i$  is the  $i$ th singular vector of  $X$ .  $s_i, U_i, V_i$  is called the  $i$ th eigentriple of SVD. Until now, it was the decomposition process of the SSA algorithm.

After deposition, the reconstruction process continues. In reconstruction, grouping (c) and averaging (d) processes are performed. In grouping (c), the result of SVD (b) is applied. It is selected by  $r$  out of  $d$  eigentriples, and  $r$  is the parameter of the SSA algorithm.  $X_I = X_{i1} + X_{i2} + \cdots + X_{ir}$ .  $X_I$  is related to the signal of  $F$ , and the remaining  $(d - r)$  eigentriples represent the error term  $\epsilon$ .

Finally, the process of averaging (d) is described. The  $r$  groups selected in the grouping (c) process are used.  $I = X_{i1} + X_{i2} + \cdots + X_{ir}$  is reconstructed into a time series  $y_{i1}, y_{i2}, \dots, y_{ir}$  through a Hankelization process or through a diagonal averaging process. If the  $k$ th term among the reconstructed time series is  $i + j = k + 1$ , the  $k$ th term can be obtained by averaging  $Z$ . Let Hankelization equal  $H()$ , which means that  $H(Z)$  is a time series of length  $T$  reconstructed in matrix  $Z$ . As a result, the following equation is produced.

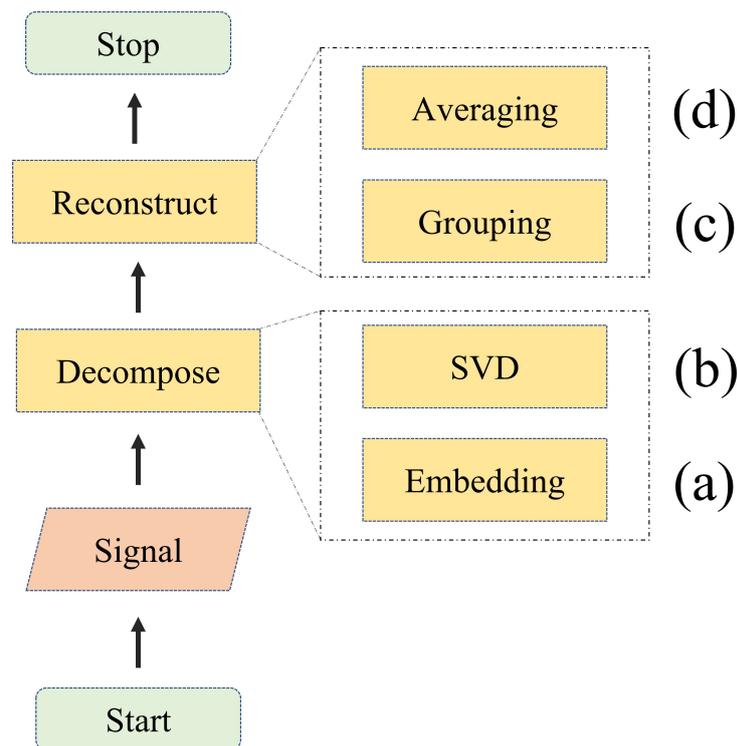
$$y = H(X_{i1}) + H(X_{i2}) + \cdots + H(X_{ir}) + e \quad (3)$$

In Formula (3),  $y$  means the reconstructed time series that is the goal of SSA.

The SSA algorithm was used in the following directions in previous studies. There have been previous studies in which SSA algorithms were used to predict time series. SSA was used as a preprocessing algorithm for time series prediction using the LSTM model [34]. In addition, similar studies have been used as a predictive model preprocessing method for machine defect detection [35].

### 2.3. LSTM

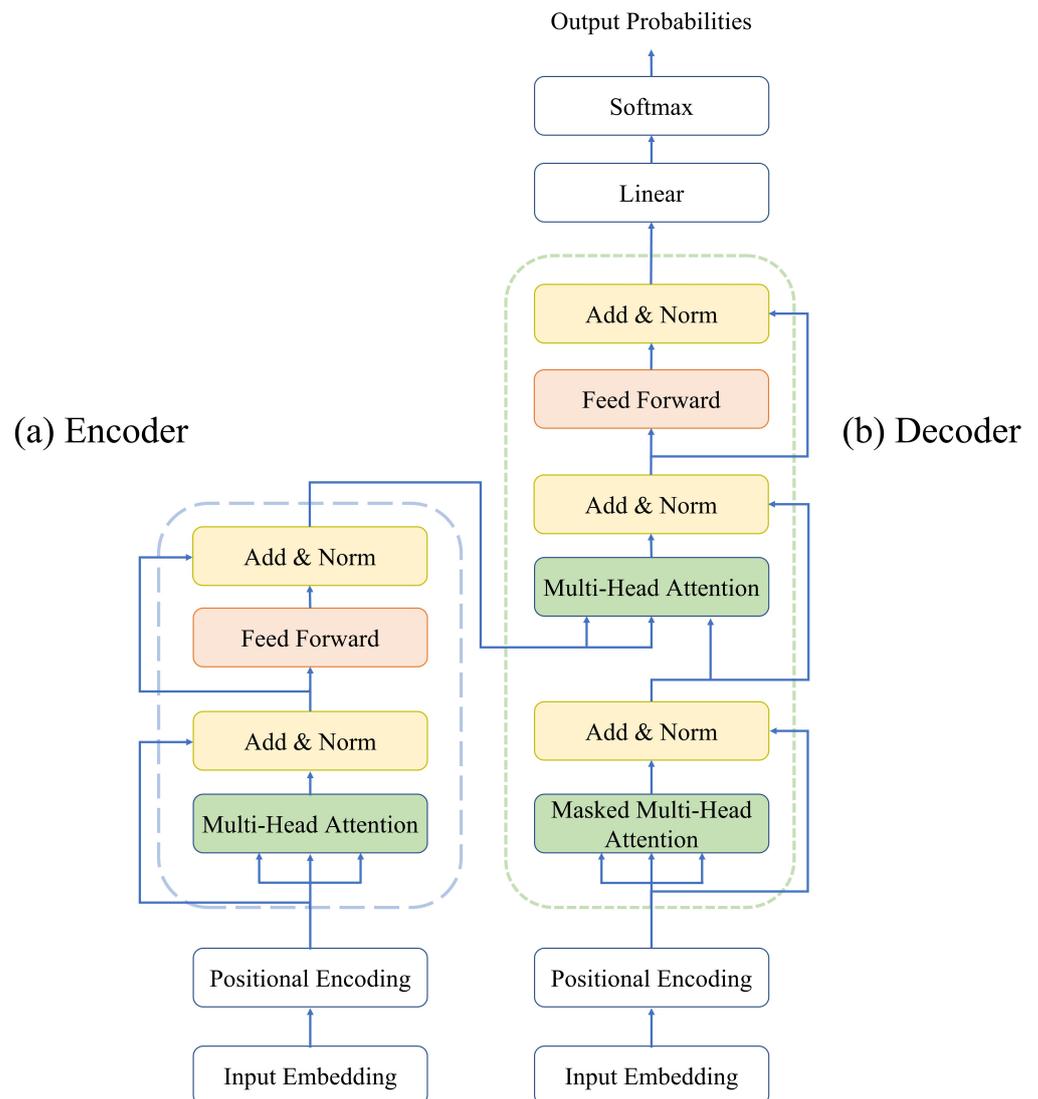
Previously, the recurrent neural network (RNN) artificial intelligence model was used for research related to time series data. However, there was a limit to RNN. Therefore, the developed long short-term memory neural network (LSTM) was developed to compensate for the shortcomings of RNN. LSTM solves the slope loss problem of RNN. LSTM can learn events from a distant past and process both high-frequency signals and low-frequency signals. The advantage of LSTM is that it has excellent time series data processing performance [36]. There have been previous studies using the LSTM model as a method of classifying time series data [37]. In addition, the LSTM model was also used in this paper, showing high performance by using it with multi-variate time series data [38].



**Figure 1.** The structure of SSA algorithm, (a) shows embedding process block, (b) shows SVD process block, (c) shows grouping process block and (d) shows averaging process block.

#### 2.4. Vanilla Transformer

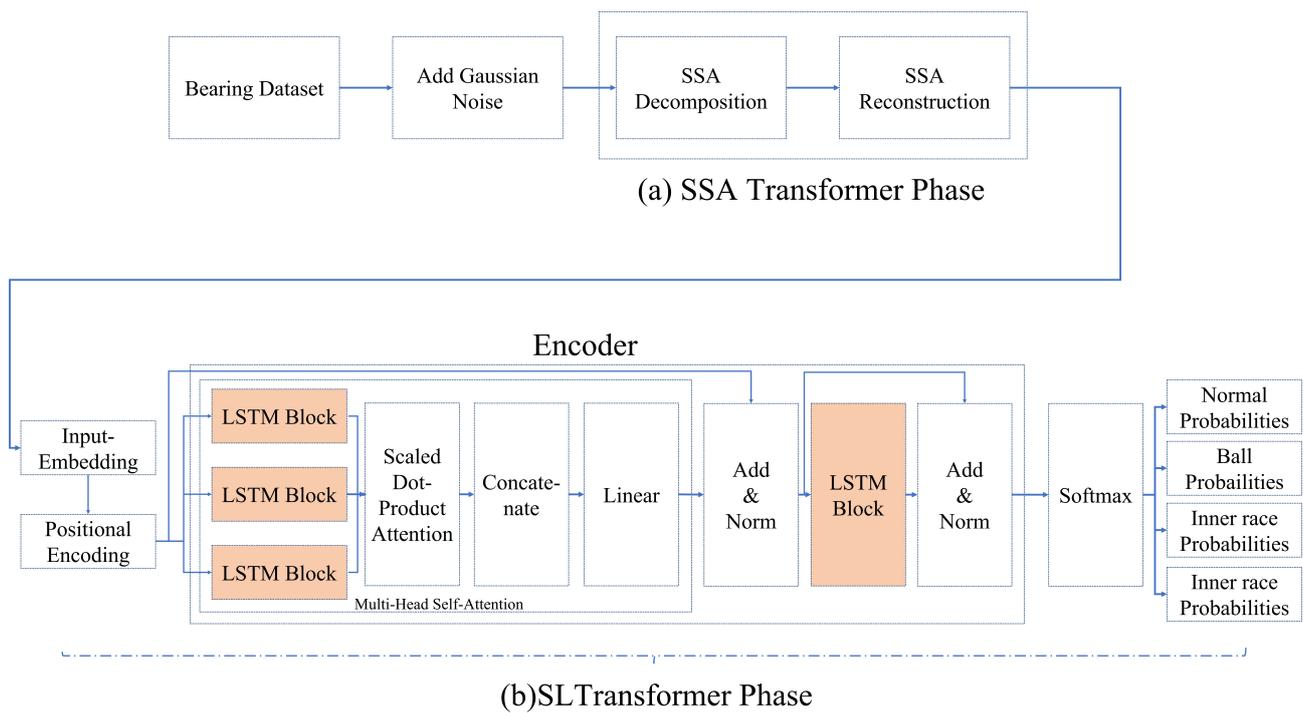
Vanilla transformers became famous by emphasizing the importance of self-attention [18], and they immediately occupy SOTA performance in the field of text generation. Subsequently, vanilla transformers are being used in various fields such as text classification and vision classification. The biggest feature of the transformer is that it removed the recurrent characteristics from the existing RNN-structured artificial intelligence architecture and took the characteristics of the attention module. The transformer model is mainly divided into encoder and decoder, depending on the purpose of use. Figure 2a denotes an encoder and Figure 2b denotes a decoder. The encoder serves to encode text as a presentation of numbers. In this case, the presentation may be referred to as embedding or features. For a given input, the input is encoded by applying a bidirectional self-attention. Moreover, the decoder functions to decode the presentations from the encoder. The decoder uses masked self-attention, to perform one-way auto-regressive learning. In this paper, bidirectional encoder representations from transformers (BERT) was mainly used [39] and applied to text classification tasks using the encoder model separately. BERT has laid the foundation for classification using an encoder by changing various approaches, such as using a Gaussian error linear unit (Gelu) activation function instead of rectified linear unit (Relu) activation [40,41]. Since then, approaches using transformer encoders have been studied in various fields.



**Figure 2.** The architecture of the vanilla transformer.

### 3. SSA-SI Transformer

The SSA-SI transformer is a framework born from the following ideas. Assuming a complex factory situation, we use an SSA algorithm that is excellent for noise decomposition and reconstruction. In addition, the transformer model, which currently shows the performance of SOTA, has been transformed into a model suitable for time series data. Among the layers inside the transformer, the linear layer entering the attention block and the multi-layer perceptron (MLP) block after the attention block were designed assuming an LSTM-oriented block. In Figure 3, one can refer to the framework and overall experiment presented in this paper, which can be separated into two processes. The first is the SSA transform phase where bearing data are mixed with noise. Denoising work is carried out through SSA modification, decompose, and reconstruct processes. The noise was mixed with white Gaussian noise. In the SI transformer phase, the swish activation function and the LSTM artificial intelligence model are used inside the transformer encoder. The highlight of the encoder in Figure 3 is a key part of this framework. The MLP block has been replaced with the LSTM, and the details are described in Section 3.2. After passing through the layers, the probability value for the data is derived through the softmax function. In addition, the artificial intelligence neural network determines what kind of bearing defect is present through the probability value.



**Figure 3.** The framework of SSA-SI transformer.

3.1. SSA Transformation

The reasons for focusing on SSA in this paper are as follows. SSA is used to reproduce and cope with the factory environment, which is the main purpose of the paper. SSA mainly describes the noise extraction of the SSA algorithm in Section 3.1. As explained initially in Section 2, SSA decomposes time series data into several subseries. SSA assumes that various components are extracted during the decomposition process, and noise is included in the components.

There are two parameters of the SSA algorithm, the window-length  $L$  and the number of components  $r$ . This may be adjusted according to the user’s discretion. In general, the window length is determined according to  $L = T/2$  while the number of components  $r$  is determined by the following method. The general criterion is to select  $r$  of the  $d$  components so that the sum of the contributions to  $\lambda_i/\Gamma$  ( $\Gamma = \sum_{i=1}^d \lambda_i$ ) is at least a predefined threshold. In general, since the noise component has a low contribution, the purpose of SSA transformation is to reconstruct time series data after removing the portion with a low contribution. In general, to observe trends and noise well, a large value of  $L$  must be selected. However, if the trend of time series data is too complex, it can be extracted only by considering a large number of eigentriples. Therefore, the goal of the SSA transformation phase can be seen as removing noise and extracting normal trends.

3.2. SL Transformer

In the SI transformer section, The SI transformer artificial intelligence model is designed with the following hypotheses. There are two main explanations. First, it explains whether LSTM is composed of the linear layer (Figure 4b) or the feed-forward layer (Figure 4a) as the main component, as shown in Figure 4d.

The SI transformer is a methodology that modifies the encoder portion of the vanilla transformer. The biggest change was the inclusion of the swish-LSTM structure, shown in Figure 4a,b. Conventional vanilla transformer constructed layers are noted in Figure 4a,b with linear MLP to extract specific data features. However, in this paper, it was judged that it is more appropriate to replace MLP with LSTM for time series data.

A detailed description is as follows. In the left encoder block of Figure 4, the linear layers receiving K, V, and Q inside the multi-head self-attention were changed. LSTM and dropout were configured in the corresponding linear layer, and the configuration shown in the right figure (d) of Figure 4 was attempted. Through this process, we tried to apply the recurrent property to the linear layer. In addition, it was judged that the swish activation function would be more suitable for time series data than Relu, an activation function used in vanilla transformer. Swish is an activation function proposed by Google. Swish is known as an active function with a smoother interface than the Relu activation function. Swish is known to converge artificial intelligence models well when learning time series data in various studies [39,42].

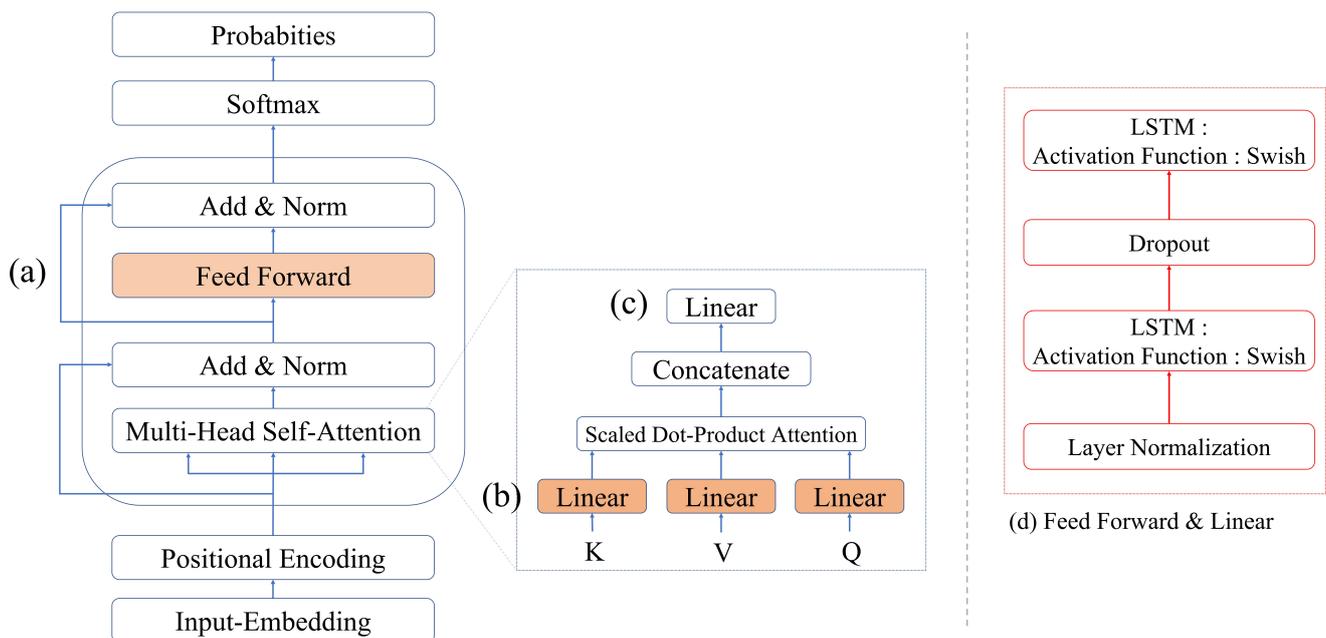


Figure 4. The architecture of the SI transformer.

In Figure 4, the LSTM network is applied to the linear network only (a) and (b) among linear networks (a), (b), and (c). First, (b) is a part that receives features of data from positional encoding as key (K), value (V), and query (Q). This Q describes a hidden state in the decoder shell, respectively, while K and V denote a hidden state in the encoder shell. When used for classification purposes, the meaning of Q used as a decoder is greatly reduced. Therefore, it was judged that the LSTM network with recurrent properties was suitable for the linear network at the self-attention entrance used for classification purposes.

On the other hand, (c) does not apply this to the linear network. From the data in Q, K, and V, self-attention determines which part of the time series is close to the core feature. In addition, the sum of the determined weights is (c), which is then sent back to the linear network. Since the time attribute is already considered, it could be assumed that applying LSTM again would increase the bias of classification.

Finally, (d) is the structure of the block applied to (a) and (b). We explain why blocks that replace MLP are configured in this way. First, the method of constructing the neural network again after dropping out immediately after normalization was referenced in the paper [43]. This paper evaluated the order of normalization and dropout.

At the end of Section 3, the overall process is explained through Figure 5 before moving on to Section 4. In Figure 5, it can be seen that there is an important process for each step: (a) represents the data received from the sensor. As for the data, the acceleration (b) represents the data acquisition system. The contents of (a) and (b) are shown in Section 4.1. A close investigation of the collection process of the CWRU public dataset was written. Then (c) shows the exploratory data analysis (EDA) process. EDA mainly identifies whether

there are no data missing values or statistically problematic data. CWRU data are known as data suitable for learning with artificial intelligence because there is no problem in kurtosis or skewness in existing studies. If there are too many missing data or a tendency to suspect intentional manipulation occurring in (c), the data are collected again by going back to step (b). Then, (d) is the process of preprocessing through SSA. In this process, the optimal L value for applying the SSA algorithm is found. This is introduced in Section 4.2, and in (e), the process of learning artificial intelligence comes out. Preprocessed data go through EDA again to see if they are suitable for artificial intelligence learning. After that, the data are divided into training, test, and validation. Hyperparameters for model learning are selected based on the divided data. Hyperparameters are shown in Table 1. In addition, scores are scored according to the metric selected in this study. However, if performance differs from the hypothesis, or if results differ significantly from similar prior studies, it returns to the process of EDA and checks again to see if the data are defective. The main goal of the experiment is to repeat this process to check whether a metric score with a hypothesis setting appears.

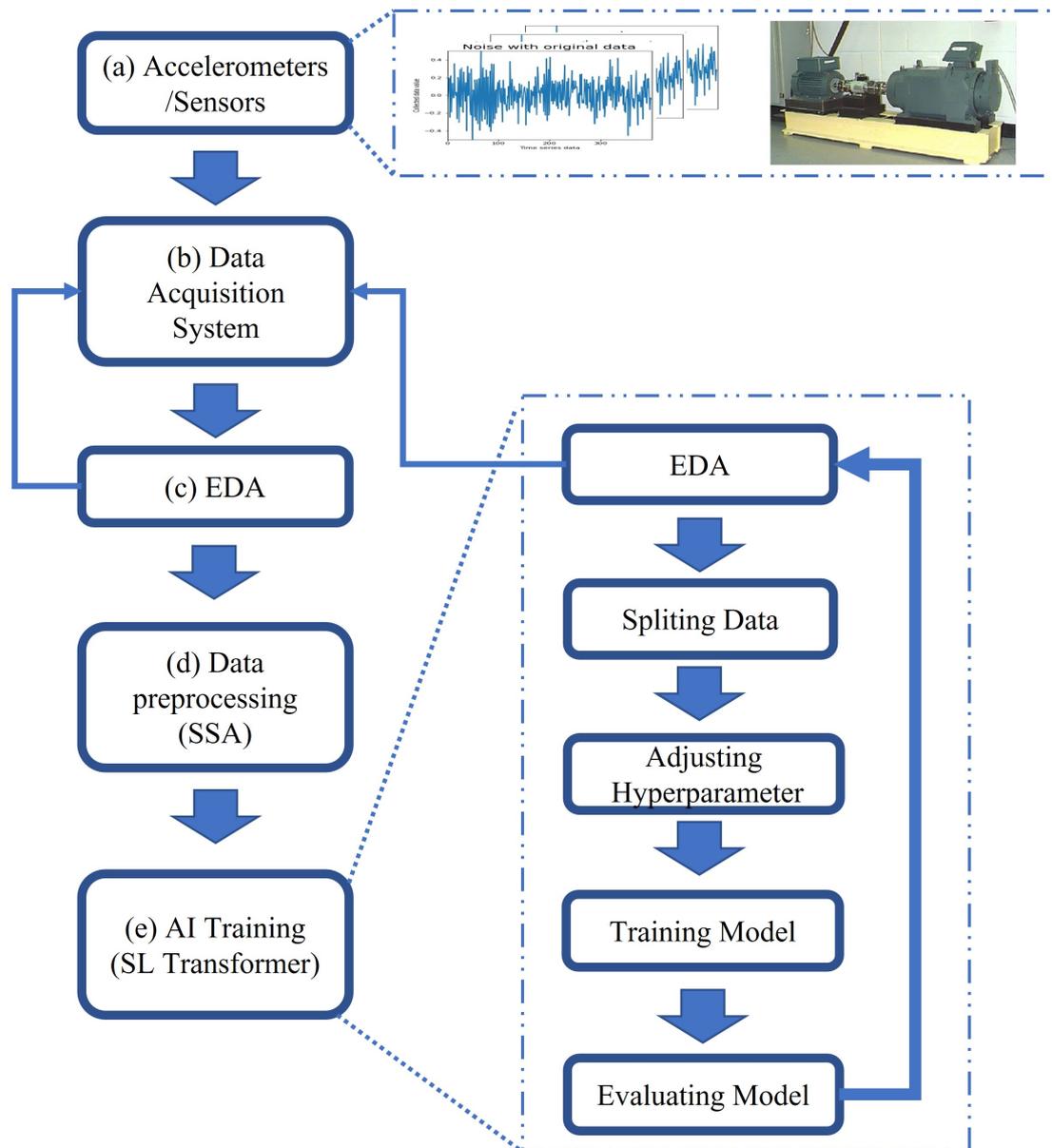


Figure 5. The block diagram of the proposed research process.

**Table 1.** The optimization structure and hyperparameters for the proposed SI transformer.

Hyperparameter	Value
Input size	[224, 224, 3]
Batch size	128
Max epochs	500
Learning rate	$5 \times 10^{-5}$
Label smoothing rate	0.1
Number of encoder layers N	6
Hidden dimension	256
Optimizer	Adam
Dropout rate	0.2
Position encoding	1D

## 4. Experiment and Results

### 4.1. Experiment Settings

The dataset used in this paper is called the Case Western Reserve University (CWRU) bearing dataset. The CWRU dataset is a supervised learning-based dataset that divides normal and faulty bearings. In these data, a total of three defects are classified, which include bearing inner ball defect, inner race defect, and outer race defect. Bearing vibration data were recorded through a load of 0–3 horsepower (motor speed 1797 to 1720 RPM). The defect test was measured using the mechanical facility shown in Figure 6a. As shown in Figure 6 above, the test stand consists of 2 bearings (left), a 2 hp motor (left), a torque transducer and encoder (center), a dynamometer (right), and control electronics (not shown). Figure 6b shows the cross-section view of the simulator.

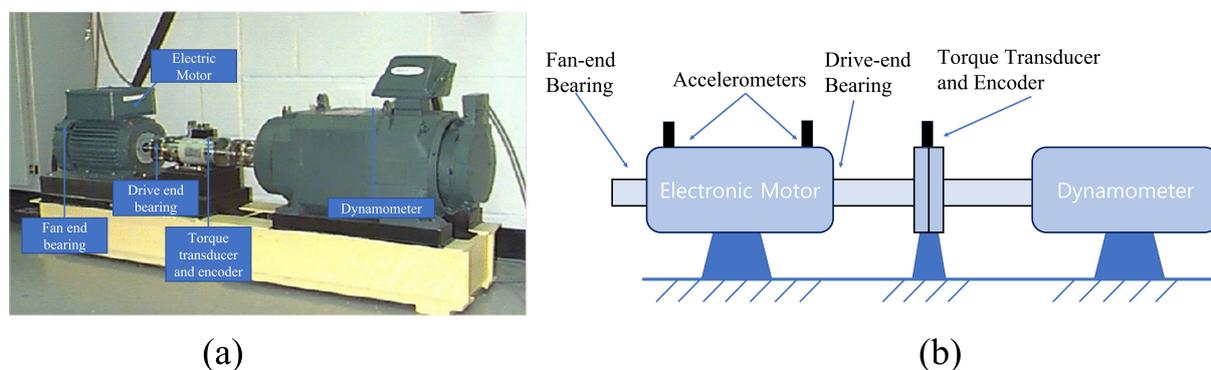
**Figure 6.** (a) The bearing simulator of CWRU and (b) its cross-section view.

Figure 7 shows the appearance of the rolling bearings of SKF6205-HC5C3. Most of the defects appear in the outer race, the inner race, and the ball. The defect was created by a process created by the electrical device called electro-discharge machining (EDM). EDM is thermo-electric machining process in which material removal takes place through the process of controlled spark generation between a pair of electrodes which are submerged in dielectric medium. Ball, inner race, and outer race defects were created in the bearing data through the EDM process. Figure 8 shows the EDM device. Figure 8a shows the servo motor that makes the tool electrode rotate (b). As the electrode rotates, a spark is generated by the electric device. In the EDM process, (c) is composed of a liquid that acts as a kind of stabilizer. Oil or water is used to operate it. The distance to the object is adjusted as in (d) to determine the degree of damage to the object. The dataset contains 4244 normal cases, 4860 ball defects, 4862 inner race defects, and 8529 outer race defects. While assuming the data structure, we considered applying the data augmentation method. However, this paper successfully reproduced the factory environment. Data augmentation was not attempted as the experiment was conducted assuming a class-imbalanced situation.

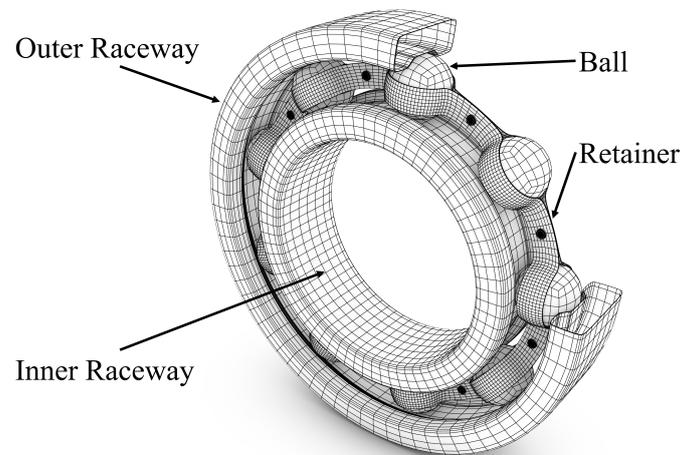


Figure 7. Components of rolling bearing.

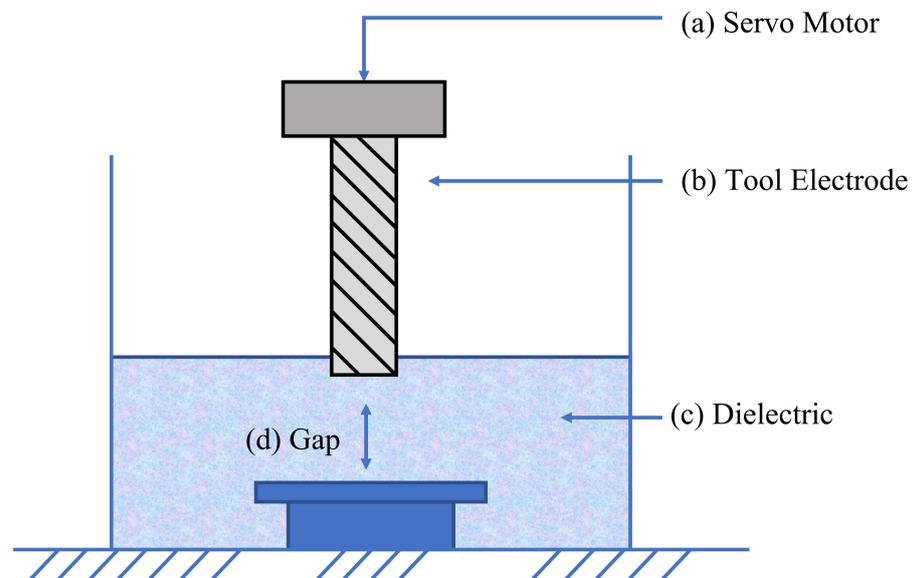
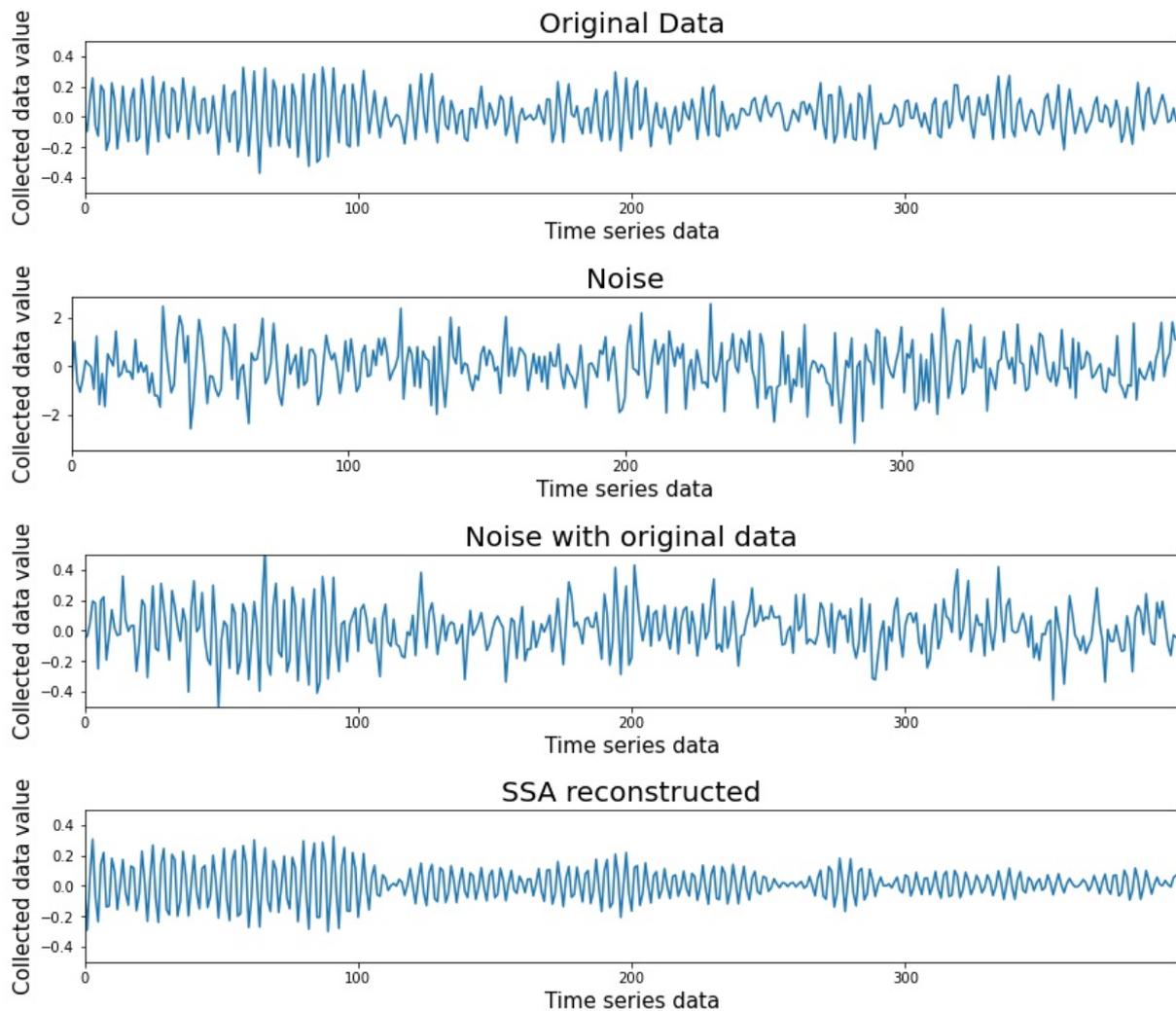


Figure 8. Schematic of an electric discharge machining (EDM) machine tool.

The hardware used in the experiment included Intel core I7-9750h CPU @ 2.60 GHz 2.59 GHz 32.0 GB RAM, Nvidia GeForce GTX in 1650. The same experiment was conducted using Google Colab Pro. Google Colab Pro is known for GPU, e.g., T4 or P100 specs. Both environments gave good results. However, it should be noted that the degree of convergence of the artificial intelligence neural network may vary depending on the GPU performance when learning the artificial intelligence neural network.

#### 4.2. Denoising with SSA Algorithm

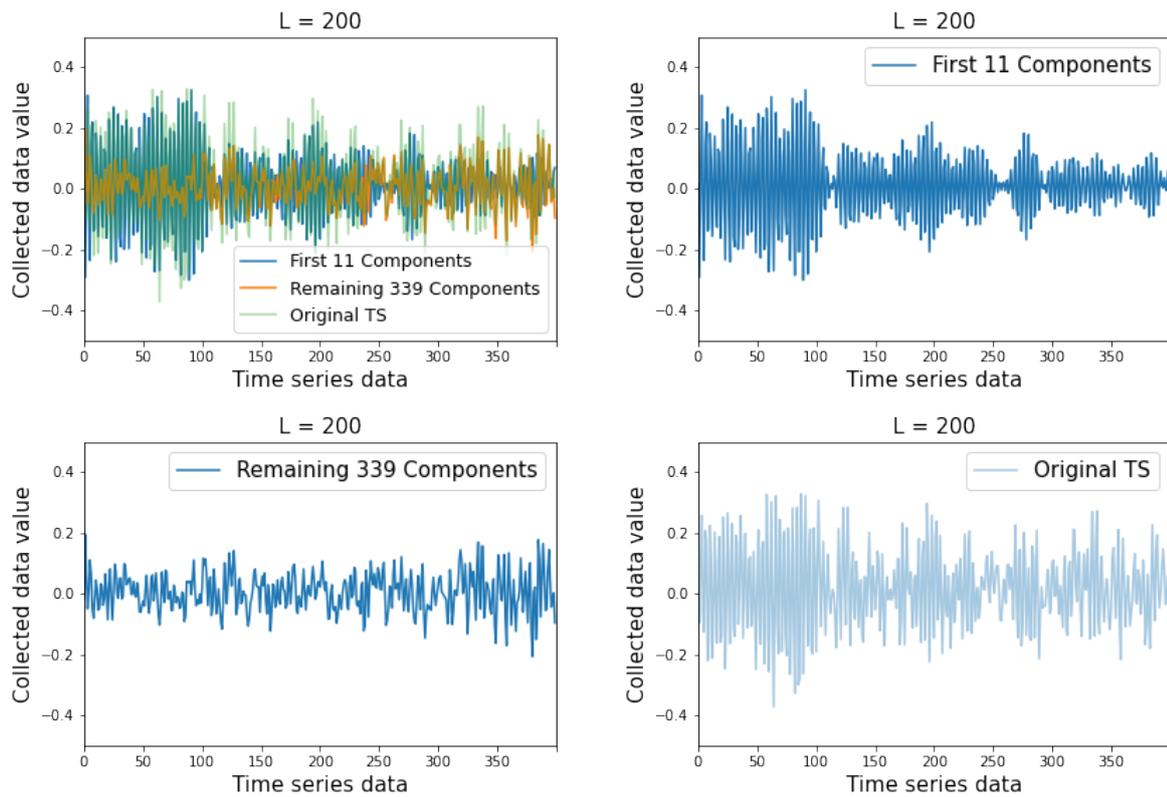
The four graphs in Figure 9 show reconstructed diagrams by original data, noise, noise with original data, and SSA algorithm, respectively. Original data brought an example of an inner race defect. As for noise, white Gaussian noise was presented. The third graph combines these two factors, and the fourth graph is the reconstruction of the data in the third graph through the SSA algorithm.



**Figure 9.** Data comparison diagram, original data (inner race), noise, noise with original data, and reconstructed data from above.

Looking at the data reconstructed with the SSA algorithm at the bottom of Figure 9, it can be seen that the original data are well reconstructed by removing noise. However, it was necessary to look at a more detailed process. Figure 10 shows the noise cancellation process based on the SSA algorithm operating principle described in Section 3.1.

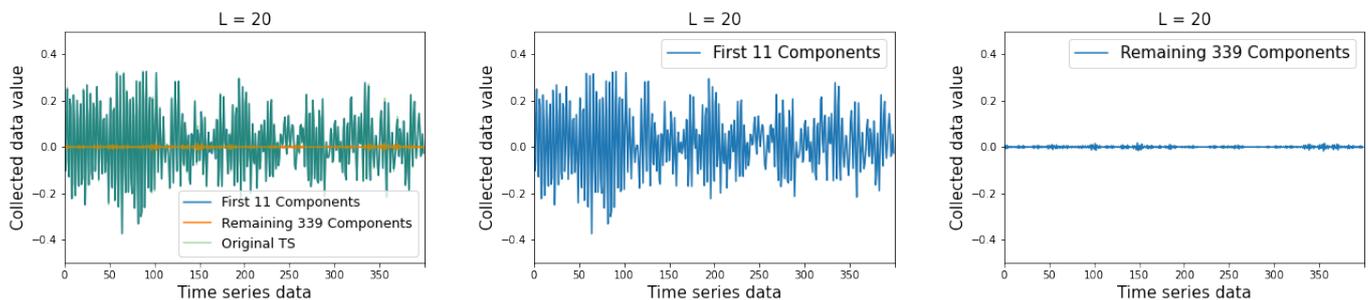
The upper right corner of Figure 10 shows the first 11 components. The SSA algorithm decomposes signal data into various time series data first and selects the most important factor among the decomposed data, which can be decided at the user's discretion. As described in Section 3.1, it was observed that it is important to select 90% of the subseries elements in the SSA algorithm grouping process. The notation in the lower left represents the remaining data in the subseries. In the reconstruction process of the SSA algorithm, these residual data are determined as noise and are not used. SSA algorithms certainly had advantages in removing noise over other preprocessing methods. However, there was an aspect of excessive data reconstruction in the process of removing and reconstructing subseries. Interesting results were found in the third experiment with the SSA algorithm. Accuracy performance was measured higher in the noisy third experiment than in the first experiment without noise. This may be proof that the SSA algorithm makes the pattern clearer. However, this is only a hypothesis, and more algorithm research seems to be needed.



**Figure 10.** Figure shows how SSA decomposes noise-mixed data. The top right, it shows the first 11 elements, the bottom left—maintaining 339 components, and the bottom right—the original time series (TS), and the top left shows mixture of components.

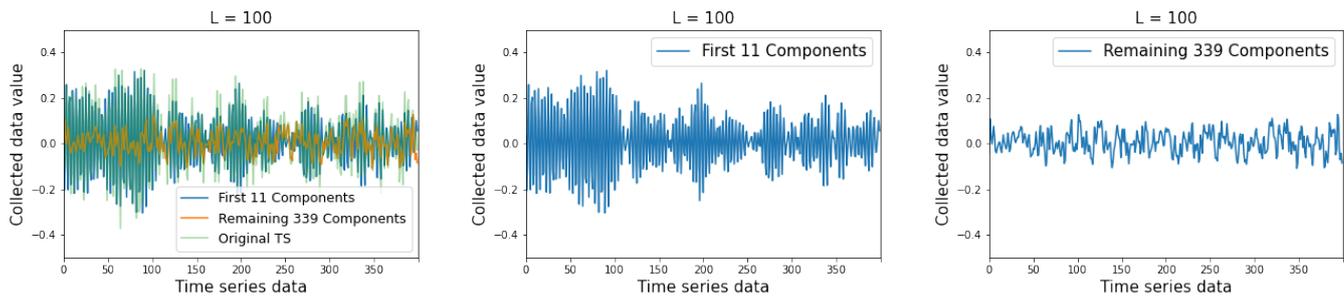
The L size in Figure 10 is set to 200. This parameter is a very important number when using SSA. The process of estimating the appropriate L was as follows.

The figures in Figure 11 are when L is 20. In the figure on the right of Figure 11, it can be seen that the noise was not properly resolved because the number of remaining components was too small. In addition, in the middle figure of Figure 11, it can be seen that the data are derived as in the original TS mixed with noise.



**Figure 11.** The result of SSA when L is 20, the middle shows first 11 components, the right shows remaining 339 components, the left shows mixture of components.

Figure 12 shows that in a situation where L is 100, many more values of remaining components are detected than when L is 20. In the middle figure of Figure 12, it can be seen that the reconstructed data of the first 11 components are in fact very similar to the reconstructed data when L is 200. However, as can be seen from the Figure 12 right side figure, it was confirmed that the noise was still less decomposed. In this way, we can find the value of L.



**Figure 12.** The result of SSA when L is 100, the middle shows first 11 components, the right shows remaining 339 components, the left shows mixture of components.

Even without going through the above method, the method used as  $L = T/2$  is most widely known. However, there is no guarantee that the L value derived from  $T/2$  during the decomposition of SSA necessarily distinguishes trend from other components well. Therefore, it is the process of finding the best parameter for the data to find the optimal L value that decomposes the trend and other components well, considering the possible cases among values below the L value.

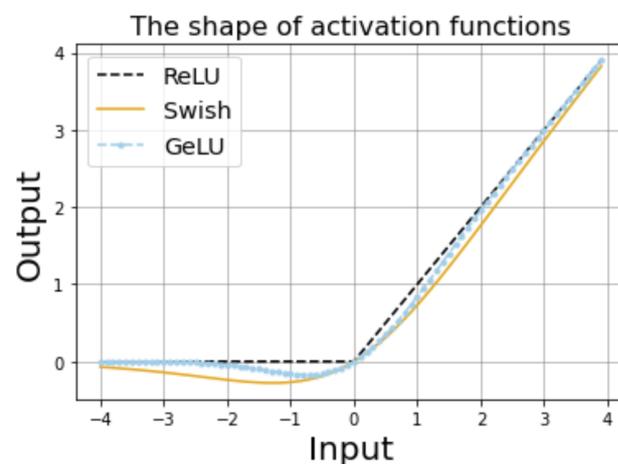
#### 4.3. Swish Activation Function

Traditionally, the ReLU activation function has been frequently used in deep learning models. However, in recent years, a number of studies are paying attention to the use of various activation functions, and in the BERT paper [39], the paper that this paper refers to as the core, compared to ReLU, GeLU has a smooth curve, which can reduce gradient vanishing. This is because GeLU can differentiate even for negative numbers, so it can transmit a small gradient.

This can also be seen in Figure 13. In Figure 13, the black line represents ReLU and the light blue line represents GeLU. As explained, GeLU has a smoother curve than ReLU. However, in this paper, swish was judged to be more advanced than GeLU, so swish was used. The formula for the swish activation function is shown in Formula (4).

$$f(x) = x \cdot \sigma(x), \quad \text{where } \sigma(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

Swish has a wider range of negative to  $x$  values than GeLU. This allows a wider range than GeLU allows for slightly negative values, allowing better representation of the gradients received from each node. Even if a small negative number is transmitted as a gradient, it can be transmitted to the previous layer, so learning is successful [42].



**Figure 13.** Comparison of GeLU, ReLU, and swish

Therefore, swish was able to establish the assumption that it would be a better type of activation function than GeLU while overcoming the limitations of the existing ReLU.

#### 4.4. Evaluation Metrics

The classification performance of the artificial intelligence model was measured using the CWRU bearing dataset. As a metric, accuracy was mainly examined (5).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

True Positive (TP) means that true is classified as true. True Negative (TN) means that true is classified as false. False Positive (FP) means that what is wrong is considered true. Finally, False Negative (FN) means the case where something is wrong is said to be wrong. The accuracy refers to the ratio of total number of samples vs. what the algorithm predicted correctly. For example, if my algorithm is 90% accurate, only 90 out of 100 samples are classified accurately.

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

The recall (6) is the proportion of the true class compared to what the model predicts as true. The parameters recall and precision have a trade-off.

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

The precision (7) describes the ratio of the true class to what the model classifies as true.

$$F1 - score = 2 \frac{Precision \times Recall}{Precision + Recall} \quad (8)$$

The f1-score (8) is called the harmonic mean, and if data labels are unbalanced, it can accurately assess the performance of the model. All experiments used the most important accuracy mentioned in the evaluation indicator, and the f1-score was used to compensate for the shortcomings in accuracy.

#### 4.5. Results

A total of three experiments were conducted. The first is to diagnose the fault of the general CWRU bearing dataset. The first experiment assumed that the SLTransformer can produce excellent performance when performing classification among comparative models. The second and third experiments evaluated the SSA algorithm. In the second experiment, data were classified without the SSA algorithm, and in the third experiment, through the SSA algorithm.

SVM, CNN, LSTM, and vanilla transformer were selected as artificial intelligence models that compare performance with Sl transformer. The artificial intelligence models listed are artificial intelligence models widely used in various domains from the past to present.

The results of the first experiment are shown in Table 2. The CWRU bearing dataset is classified. As expected, the experiment confirmed that the Sl transformer performed the best, as it recorded 95% accuracy and a 94% F1-score. Finally, it was confirmed that the Sl transformer performed better than the vanilla transformer. It has been confirmed that the accuracy of the artificial intelligence model has increased further through the method of replacing the MLP of the vanilla transformer with LSTM. Further, the reason for using swish activation was for a more stable convergence of artificial intelligence model learning. Looking at Figures 11 and 12, it was confirmed that the result accuracy and loss have a stable curve shape.

A total of 95% accuracy is a very good value for an artificial intelligence model. However, this is not realistic. The CWRU dataset has a relatively small amount of noise among bearing datasets. Therefore, it is difficult to reflect the actual factory environment, and was necessary to conduct another additional experiment.

**Table 2.** The table of accuracy of the SI transformer and compared model.

Model	SVM	CNN	LSTM	Transformer	SI Transformer
Accuracy	64.32	85.69	90.15	93.47	95.54
F1-score	57.44	84.44	89.72	92.31	94.47

The second and third experiments are about noise, which was inserted into the bearing data. The purpose of this experiment was to test how robust the methodology was to noise. First, the framework of this paper explains how to diagnose normal abnormalities in a noise-laden environment. The experiment was conducted by inserting Gaussian white noise into the CWRU dataset and then diagnosing the modified data. Experiments involving noise are close to actual conditions at industrial sites where the situation is not constant and various noise occurs. In this experiment, noise changes were measured using the signal to noise ratio (SNR) index [44]. The definition of SNR is as follows (8).

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \left( \frac{P_{\text{signal}}}{P_{\text{noise}}} \right) \quad (9)$$

Here,  $P_{\text{signal}}$  and  $P_{\text{noise}}$  are the power of signals and noise, respectively. The original signal is added to the inner race fault by additional white Gaussian noise. The white Gaussian noise ranged from  $-4$  dB to  $4$  dB.

The second experiment was on a non-applying SSA algorithm that shows how vulnerable artificial intelligence algorithms are to noise data. The second experiment can be confirmed through Tables 3–6. The experimental results shown in Tables 3–6 are shown in Figures 14 and 15, which are visualizations of data. In addition, Figure 16 is the data visualizing Table 3, and Figure 17 is the data visualizing Table 5. An experiment was conducted to compare accuracy and F1-score by dB in the preceding five comparison groups. As expected, the experiment showed a significant drop in accuracy. In experiments without SSA, most artificial intelligence models exhibit accuracy and F1-score performance below 70% at  $-4$  dB. The third experiment was about the SSA algorithm application, where it can be seen how robust SSA is to noise, and at the same time, the SI transformer performance can be verified. In the third experiment, interesting results were observed. Higher performance (96%) was observed in the presence of noise vs. 95% accuracy performance in a noise-free situation. This result made it possible to establish another hypothesis: that time series data were reconstructed more appropriately for artificial intelligence neural networks in the process of decomposing and reconstructing SSA algorithms. In conclusion, in this experiment, it was confirmed that the SI transformer recorded higher accuracy and F1-score overall compared to comparative models, and the SSA algorithm was robust to data mixed with noise.

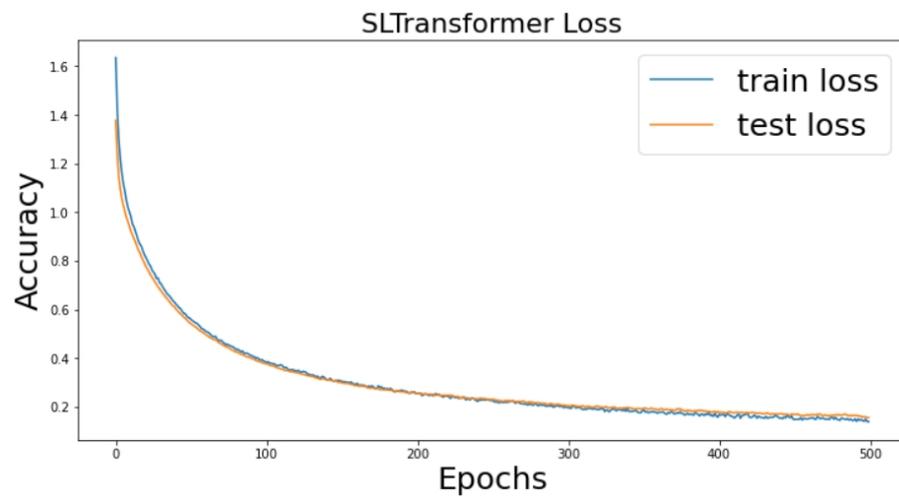


Figure 14. The loss graph of SI transformer. The x-axis describes epochs and the y-axis describes loss.

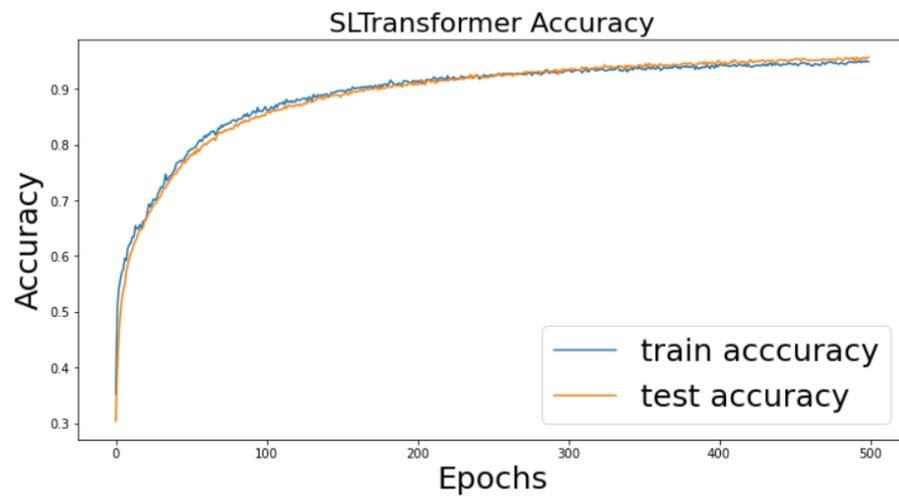


Figure 15. This accuracy graph of SI transformer. The x-axis describes epochs and the y-axis describes accuracy.

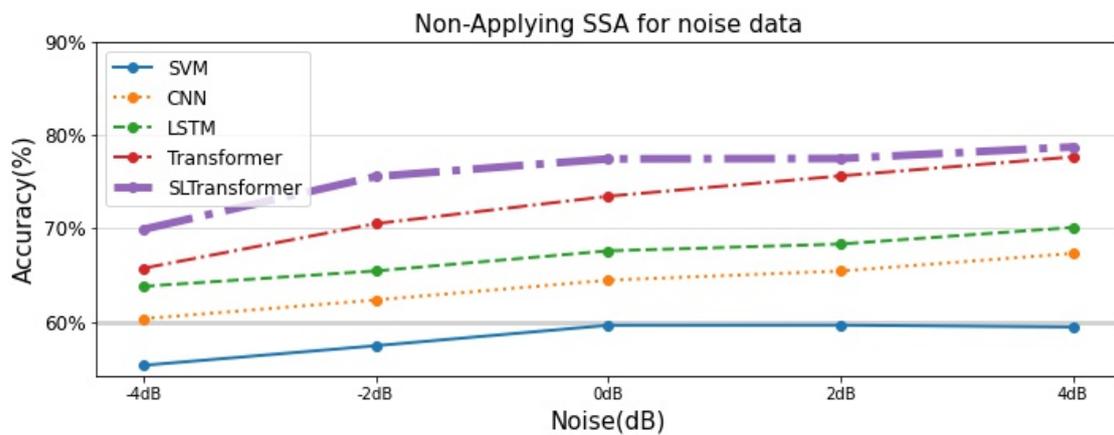


Figure 16. The accuracy graph of non-applying SSA algorithm for noise data.

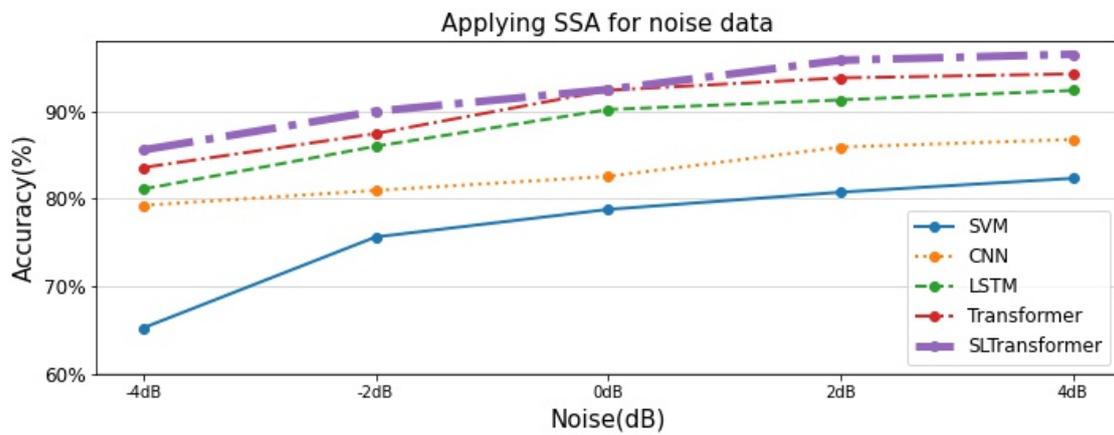


Figure 17. The accuracy graph of applying SSA algorithm for noise data.

Table 7 is composed of scenarios assuming situations in which the application of this experiment is required. The problem situations that usually occur in existing factories can be seen in before, and the situation that is solved through SSA-SI transformer can be seen in after.

Table 3. The accuracy table of non-applying SSA algorithm for noise data.

Model	−4 dB	−2 dB	0 dB	2 dB	4 dB
SVM	55.33	57.43	59.62	59.63	59.44
CNN	60.32	62.33	64.45	65.43	67.32
LSTM	63.79	65.44	67.59	68.32	70.10
Transformer	65.72	70.50	73.44	75.62	77.66
SI transformer	69.89	75.55	77.44	77.48	78.72

Table 4. The F1-score table of non-applying SSA algorithm for noise data.

Model	−4 dB	−2 dB	0 dB	2 dB	4 dB
SVM	54.79	55.44	58.41	58.01	58.88
CNN	59.44	61.15	61.90	63.57	64.23
LSTM	61.55	64.31	65.11	66.15	69.77
Transformer	64.16	68.51	72.45	74.10	76.09
SI transformer	68.13	74.10	76.36	75.66	77.77

Table 5. The accuracy table of applying SSA algorithm.

Model	−4 dB	−2 dB	0 dB	2 dB	4 dB
SVM	65.30	75.65	78.79	80.75	82.33
CNN	79.25	80.95	82.55	85.88	86.75
LSTM	81.10	85.97	90.15	91.25	92.33
Transformer	83.55	87.42	92.35	93.75	94.21
SI transformer	85.55	89.95	92.44	95.76	96.44

Table 6. The F1-score table of applying SSA algorithm.

Model	−4 dB	−2 dB	0 dB	2 dB	4 dB
SVM	64.24	74.10	76.42	77.10	79.17
CNN	78.15	79.95	81.34	83.11	85.15
LSTM	80.15	82.44	85.57	90.15	91.99
Transformer	82.11	85.17	88.44	92.28	93.12
SI transformer	84.00	86.37	92.15	94.35	95.04

**Table 7.** Before and after comparison with SSA-SI transformer introduction scenario.

Before	After
- Bearing data are retrieved from the sensor for bearing fault detection, but noise occurs in the data.	- Accept situations in which noise is generated rather than attempts to block the noise at the source.
- Remove the generated noise, and use denoise autoencoder as a representative method. As the neural network for classify and the neural network for denoise work together, real-time inspection of defects becomes difficult .	- By performing SSA preprocessing on the generated noise, the data mixed with noise are decomposed and reconstructed, so that a large number of noises are separated from the data.
- If noise is not removed, the performance of the artificial intelligence model will drop sharply due to noise generation, and deep learning projects will fail.	- Solves the classify problem by using a robust SI transformer model on time series data on noise-free data.

## 5. Conclusions

This paper proposes the SSA-SI transformer framework to solve the bearing fault diagnosis challenge. While conducting the bearing fault diagnosis research, it was possible to discover the limitations of existing studies and present the following hypothesis. Many existing studies did not assume the actual environment of the factory. The actual factory produces a variety of noise. In general, it is difficult to collect only clean data. Therefore, we assumed and implemented the actual factory environment as a situation in which noise is inserted into the bearing data. Therefore, the process of overcoming data mixed with noise is the main content of this paper. The proposed method in this paper is expected to be a re-examination of preprocessing techniques and a proposal of a new artificial intelligence model. SSA preprocessing techniques allow noise to be decomposed and reconstructed to create a robust framework for noise. An existing artificial intelligence model, a transformer, was newly approached. An LSTM with a recurrent attribute was applied inside the transformer, and a swish activation function suitable for time series data was used. This artificial intelligence model was an approach that did not exist before. We demonstrate the model performance of the SI transformer through various metrics. Three experiments were conducted to prove the hypothesis. The first experiment derived metrics scores using SI transformer for the CWRU dataset. The first experiment confirmed that the performance was better than that of the vanilla transformer, and it was confirmed that the classification was performed well. The second experiment added noise to the CWRU dataset. The degree of noise addition was confirmed by dividing it into dB. The second experiment confirmed that the artificial intelligence model was vulnerable to noise, as hypothesized. The third experiment added the SSA preprocessing process to the second experiment. When the noise was decomposed and reconstructed through SSA, it was confirmed that the performance of the artificial intelligence neural network was significantly improved, and in particular, the SSA-SI transformer showed a 96% performance.

Through this paper, follow-up researchers and engineers hope to obtain the following. Good performance can be expected by using the SSA methodology when processing data noise in an artificial intelligence framework, and we present an existing SI transformer artificial intelligence model. We look forward to further developing this model. The experimental conditions and parameters were presented in Sections 3 and 4 to enable this experiment to be implemented. Engineers can expect to be able to solve the noise generated when collecting data with sensors using the method of this paper.

However, the limitations of this paper were as follows: Section 4.4 also measured when noise (dB) was positive, and in the case of positive values, we found that the performance of artificial intelligence neural network models improved. The same phenomenon was also found in [22], but there was no mention of related content in the paper. Therefore, it was possible to hypothesize that a certain level of dB could create robust artificial intelligence, which was left as future work.

**Author Contributions:** Conceptualization, S.L. and J.J.; methodology, S.L.; software, S.L.; validation, S.L. and J.J.; formal analysis, S.L.; investigation, S.L.; resources, J.J.; data curation, S.L.; writing—original draft preparation, S.L.; writing—review and editing, J.J.; visualization, S.L.; supervision, J.J.; project administration, J.J.; funding acquisition, J.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ICT Creative Consilience Program (IITP-2022-2020-0-01821) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation), and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1F1A1060054).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data used in the paper can be found at <https://engineering.case.edu/bearingdatacenter/download-data-file>. This is the dataset of a bearing research conducted by Case Western Reserve University. Access date: 22 September 2021.

**Acknowledgments:** This research was supported by the SungKyunKwan University and the BK21 FOUR (Graduate School Innovation) funded by the Ministry of Education (MOE, Korea) and National Research Foundation of Korea (NRF).

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

## References

1. Kalsoom, T.; Ramzan, N.; Ahmed, S.; Ur-Rehman, M. Advances in sensor technologies in the era of smart factory and industry 4.0. *Sensors* **2020**, *20*, 6783. [[CrossRef](#)]
2. Pech, M.; Vrchota, J.; Bednář, J. Predictive maintenance and intelligent sensors in smart factory. *Sensors* **2021**, *21*, 1470. [[CrossRef](#)]
3. Sufian, A.T.; Abdullah, B.M.; Ateeq, M.; Wah, R.; Clements, D. Six-Gear Roadmap towards the Smart Factory. *Appl. Sci.* **2021**, *11*, 3568. [[CrossRef](#)]
4. Sinha, D.; Roy, R. Reviewing cyber-physical system as a part of smart factory in industry 4.0. *IEEE Eng. Manag. Rev.* **2020**, *48*, 103–117. [[CrossRef](#)]
5. Büchi, G.; Cugno, M.; Castagnoli, R. Smart factory performance and Industry 4.0. *Technol. Forecast. Soc. Chang.* **2020**, *150*, 119790. [[CrossRef](#)]
6. Toma, R.N.; Prosvirin, A.E.; Kim, J. Bearing fault diagnosis of induction motors using a genetic algorithm and machine learning classifiers. *Sensors* **2020**, *20*, 1884. [[CrossRef](#)]
7. Li, H.; Liu, T.; Wu, X.; Chen, Q. A bearing fault diagnosis method based on enhanced singular value decomposition. *IEEE Trans. Ind. Inform.* **2020**, *17*, 3220–3230. [[CrossRef](#)]
8. Kuncan, M. An intelligent approach for bearing fault diagnosis: Combination of 1D-LBP and GRA. *IEEE Access* **2020**, *8*, 137517–137529. [[CrossRef](#)]
9. Hoang, D.-T.; Kang, H.-J. A survey on deep learning based bearing fault diagnosis. *Neurocomputing* **2019**, *335*, 327–335. [[CrossRef](#)]
10. Yuan, L.; Lian, D.; Kang, X.; Chen, Y.; Zhai, K. Rolling bearing fault diagnosis based on convolutional neural network and support vector machine. *IEEE Access* **2020**, *8*, 137395–137406. [[CrossRef](#)]
11. Han, T.; Zhang, L.; Yin, Z.; Tan, A.C. Rolling bearing fault diagnosis with combined convolutional neural networks and support vector machine. *Measurement* **2021**, *177*, 109022. [[CrossRef](#)]
12. Zhang, X.; Han, P.; Xu, L.; Zhang, F.; Wang, Y.; Gao, L. Research on bearing fault diagnosis of wind turbine gearbox based on 1DCNN-PSO-SVM. *IEEE Access* **2020**, *8*, 192248–192258. [[CrossRef](#)]
13. Zhang, R.; Li, B.; Jiao, B. Application of XGboost algorithm in bearing fault diagnosis. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2019; Volume 490.
14. Neupane, D.; Seok, J. Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: A review. *IEEE Access* **2020**, *8*, 93155–93178. [[CrossRef](#)]
15. Zhu, J.; Jiang, Q.; Shen, Y.; Qian, C.; Xu, F.; Zhu, Q. Application of recurrent neural network to mechanical fault diagnosis: A review. *J. Mech. Sci. Technol.* **2022**, *36*, 527–542. [[CrossRef](#)]
16. Liu, H.; Zhou, J.; Zheng, Y.; Jiang, W.; Zhang, Y. Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders. *ISA Trans.* **2018**, *77*, 167–178. [[CrossRef](#)]
17. Eren, L.; Ince, T.; Kiranyaz, S. A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier. *J. Signal Process. Syst.* **2019**, *91*, 179–189. [[CrossRef](#)]
18. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, arXiv:1706.03762.
19. Han, K.; Xiao, A.; Wu, E.; Guo, J.; Xu, C.; Wang, Y. Transformer in transformer. *arXiv* **2021**, arXiv:2103.00112.

20. Ha, S.; Marchetto, D.J.; Dharur, S.; Asensio, O.I. Topic classification of electric vehicle consumer experiences with transformer-based deep learning. *Patterns* **2021**, *2*, 100195. [[CrossRef](#)]
21. Li, G.; Deng, C.; Wu, J.; Chen, Z.; Xu, X. Rolling bearing fault diagnosis based on wavelet packet transform and convolutional neural network. *Appl. Sci.* **2020**, *10*, 770. [[CrossRef](#)]
22. Zhang, W.; Li, C.; Peng, G.; Chen, Y.; Zhang, Z. A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. *Mech. Syst. Signal Process.* **2018**, *100*, 439–453. [[CrossRef](#)]
23. Oh, S.; Han, S.; Jeong, J. Multi-scale convolutional recurrent neural network for bearing fault detection in noisy manufacturing environments. *Appl. Sci.* **2021**, *11*, 3963. [[CrossRef](#)]
24. Hassani, H. Singular spectrum analysis: Methodology and comparison. *J. Data Sci.* **2007**, *5*, 239–257. [[CrossRef](#)]
25. Golyandina, N.; Korobeynikov, A.; Zhigljavsky, A. *Singular Spectrum Analysis with R*; Springer: Berlin/Heidelberg, Germany, 2018.
26. Yan, X.; Xu, Y.; She, D.; Zhang, W. Reliable fault diagnosis of bearings using an optimized stacked variational denoising auto-encoder. *Entropy* **2021**, *24*, 36. [[CrossRef](#)] [[PubMed](#)]
27. Chen, Z.; Li, W. Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 1693–1702. [[CrossRef](#)]
28. Liu, H.; Li, L.; Ma, J. Rolling bearing fault diagnosis based on STFT-deep learning and sound signals. *Shock Vib.* **2016**, *2016*, 6127479. [[CrossRef](#)]
29. Islam, M.M.; Kim, J.M. Automated bearing fault diagnosis scheme using 2D representation of wavelet packet transform and deep convolutional neural network. *Comput. Ind.* **2019**, *106*, 142–153. [[CrossRef](#)]
30. Abdelkader, R.; Kaddour, A.; Derouiche, Z. Enhancement of rolling bearing fault diagnosis based on improvement of empirical mode decomposition denoising method. *Int. J. Adv. Manuf. Technol.* **2018**, *97*, 3099–3117. [[CrossRef](#)]
31. Chen, B.; Zhang, W.; Song, D.; Cheng, Y. Blind deconvolution assisted with periodicity detection techniques and its application to bearing fault feature enhancement. *Measurement* **2020**, *159*, 107804. [[CrossRef](#)]
32. Qiao, Z.; Elhattab, A. A second-order stochastic resonance method enhanced by fractional-order derivative for mechanical fault detection. *Nonlinear Dyn.* **2021**, *106*, 707–723. [[CrossRef](#)]
33. Qiao, Z.; Pan, Z. SVD principle analysis and fault diagnosis for bearings based on the correlation coefficient. *Meas. Sci. Technol.* **2015**, *26*, 085014. [[CrossRef](#)]
34. Rodrigues, P.C.; Pimentel, J.; Messala, P.; Kazemi, M. The decomposition and forecasting of mutual investment funds using singular spectrum analysis. *Entropy* **2020**, *22*, 83. [[CrossRef](#)] [[PubMed](#)]
35. Hassani, H.; Yeganegi, M.R.; Khan, A.; Silva, E.S. The effect of data transformation on singular spectrum analysis for forecasting. *Signals* **2020**, *1*, 4–25. [[CrossRef](#)]
36. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 2222–2232. [[CrossRef](#)] [[PubMed](#)]
37. Karim, F.; Majumdar, S.; Darabi, H.; Chen, S. LSTM fully convolutional networks for time series classification. *IEEE Access* **2017**, *6*, 1662–1669. [[CrossRef](#)]
38. Karim, F.; Majumdar, S.; Darabi, H.; Harford, S. Multivariate LSTM-FCNs for time series classification. *Neural Netw.* **2019**, *116*, 237–245. [[CrossRef](#)]
39. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
40. Li, Y.; Yuan, Y. Convergence analysis of two-layer neural networks with relu activation. *Adv. Neural Inf. Process. Syst.* **2017**, arXiv:1705.09886.
41. Nguyen, A.; Pham, K.; Ngo, D.; Ngo, T.; Pham, L. An Analysis of State-of-the-art Activation Functions For Supervised Deep Neural Network. In Proceedings of the 2021 International Conference on System Science and Engineering (ICSSE), Nha Trang, Vietnam, 26–28 August 2021.
42. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for activation functions. *arXiv* **2017**, arXiv:1710.05941.
43. Garbin, C.; Zhu, X.; Marques, O. Dropout vs. batch normalization: An empirical study of their impact to deep learning. *Multimed. Tools Appl.* **2020**, *79*, 12777–12815. [[CrossRef](#)]
44. Chang, C.-I. Hyperspectral Target Detection: Hypothesis Testing, Signal-to-Noise Ratio, and Spectral Angle Theories. *IEEE Trans. Geosci. Remote. Sens.* **2021**, *60*, 1–23. [[CrossRef](#)]