

Article

Comparative Performance Analysis of RNN Techniques for Predicting Concatenated Normal and Abnormal Vibrations

Ju-Hyung Lee ¹  and Jun-Ki Hong ^{2,*}

¹ Department of Computer Engineering, Pai Chai University, Daejeon 35345, Republic of Korea; 2237012@pcu.ac.kr

² Department of Smart Information Technology Engineering, Kongju National University, Cheonan 31080, Republic of Korea

* Correspondence: jkhong@kongju.ac.kr

Abstract: We analyze the comparative performance of predicting the transition from normal to abnormal vibration states, simulating the motor's condition before a drone crash, by proposing a concatenated vibration prediction model (CVPM) based on recurrent neural network (RNN) techniques. Subsequently, using the proposed CVPM, the prediction performances of six RNN techniques: long short-term memory (LSTM), attention-LSTM (Attn.-LSTM), bidirectional-LSTM (Bi-LSTM), gate recurrent unit (GRU), attention-GRU (Attn.-GRU), and bidirectional-GRU (Bi-GRU), are analyzed comparatively. In order to assess the prediction accuracy of these RNN techniques in predicting concatenated vibrations, both normal and abnormal vibration data are collected from the motors connected to the drone's propellers. Consequently, a concatenated vibration dataset is generated by combining 50% of normal vibration data with 50% of abnormal vibration data. This dataset is then used to compare and analyze vibration prediction performance and simulation runtime across the six RNN techniques. The goal of this analysis is to comparatively analyze the performances of the six RNN techniques for vibration prediction. According to the simulation results, it is observed that Attn.-LSTM and Attn.-GRU, incorporating the attention mechanism technique to focus on information highly relevant to the prediction target through unidirectional learning, demonstrate the most promising predictive performance among the six RNN techniques. This implies that employing the attention mechanism enhances the concentration of relevant information, resulting in superior predictive accuracy compared to the other RNN techniques.

Keywords: recurrent neural network (RNN); time series concatenated vibration; vibration prediction; long short-term memory (LSTM); gated recurrent unit (GRU); attention mechanism; bidirectional RNN



Citation: Lee, J.-H.; Hong, J.-K. Comparative Performance Analysis of RNN Techniques for Predicting Concatenated Normal and Abnormal Vibrations. *Electronics* **2023**, *12*, 4778. <https://doi.org/10.3390/electronics12234778>

Academic Editors: Hao Xue and Du Huynh

Received: 23 October 2023
Revised: 17 November 2023
Accepted: 23 November 2023
Published: 25 November 2023



Copyright: © 2023 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

Unmanned drones have recently gained prominence as a core technology of the fourth industrial revolution. Drones find extensive applications in diverse fields, including transportation [1–4], search and rescue operations [5–11], and agriculture [12–17]. Moreover, research is ongoing to effectively control drones in the applications mentioned above [18,19]. According to the ‘Drone Market Analysis’ by Drone Industry Insights, released in 2022, the size of the drone market in 2022 is approximately \$30.6 billion, and it is expected to grow to about \$55.8 billion by 2030 with an average annual growth rate of 7.8% [20].

However, drone crashes can lead to significant incidents, such as fires caused by battery explosions or even accidents resulting in human casualties. According to a drone safety survey released by the Korea Consumer Agency, the trend of safety incidents involving drones is on the rise as the purchase of drones for leisure increases, with drone crashes accounting for approximately 20% of the reported incidents [21]. Therefore, research has been conducted to detect anomalies by training and predicting time series vibration data

using recurrent neural network (RNN). Table 1 below illustrates previous studies where RNN techniques are employed to train and predict time series vibration data.

Table 1. Previous research on RNN-based predictions of time series vibration data.

Purpose	Studies	Models	Evaluation Criteria
Prediction of spindle rotation error	Liang et al. [22]	Bi-LSTM	Accuracy
Prediction of bearing life	Han et al. [23]	LSTM	MSE, RMSE, and MAE
Prediction of turbine condition	Adlen et al. [24]	LSTM, GRU, etc.	RMSE, MAE, and MAPE
Fault detection of gear	Li et al. [25]	GRU, CNN	Accuracy, loss, and training time
Fault detection of quadcopter	Zhang et al. [26]	LSTM	Accuracy
Fault detection of bearings and gearboxes	Ruslan et al. [27]	RNN and CNN	Accuracy
Fault detection of motor	Xiao et al. [28]	LSTM, RNN, etc.	Accuracy
	Huang et al. [29]	LSTM, GRU, etc.	Accuracy and loss
	Wang et al. [30]	LSTM, GRU, etc.	RMSE, MAPE, and MSE
	Wang et al. [31]	LSTM	Accuracy
Prediction of vibration	ElSaid et al. [32]	LSTM	MSE, MAE
	Zhu et al. [33]	GRU and ANN	RMSE
	Yang et al. [34]	RNN, LSTM, etc.	MAE, MSE, RMSE, etc.
	Hong [35]	LSTM, GRU, etc.	Accuracy and sim. runtime
	Lee et al. [36]	LSTM, GRU, etc.	Accuracy and sim. runtime

As shown in Table 1, research has been conducted in various fields using RNN techniques to train and predict vibrations. Furthermore, a previous study [36] forecasted both normal and abnormal drone vibrations individually using six RNN techniques: long short-term memory (LSTM), attention-LSTM (Attn.-LSTM), bidirectional-LSTM (Bi-LSTM), gate recurrent unit (GRU), attention-GRU (Attn.-GRU), and bidirectional-GRU (Bi-GRU). However, the previous studies had limitations, as they predicted normal and abnormal vibrations separately without considering the transition from normal to abnormal vibrations in real-world scenarios when the motor of a drone is damaged.

Therefore, in this study, an RNN-based concatenated vibration prediction model (CVPM) is proposed to comparatively analyze the performance of predicting concatenated vibrations transitioning from normal to abnormal states. Furthermore, six RNN models—LSTM, Bi-LSTM, Attn.-LSTM, GRU, Attn.-GRU, and Bi-GRU—are applied to assess the comparative prediction accuracy and simulation runtime of the concatenated vibrations using identical simulation parameters, including the optimizer and the number of hidden units. Subsequently, the vibration prediction accuracy of the six RNN techniques is analyzed in relation to simulation runtime using the proposed CVPM. The major contributions of this study are summarized as follows:

1. Since previous studies focused on researching normal and abnormal vibrations separately, the CVPM is proposed to analyze the predictive performance of the six RNN techniques for concatenated vibrations. These concatenated vibrations represent the transition from normal to abnormal states reflecting the sudden damage to the motor.
2. This study involves a comparative analysis of the prediction efficiency of the six RNN techniques, examining not only the prediction accuracy but also the simulation runtime. The initiative that we took to examine the simulation runtime for each of the six RNN techniques utilized in the CVPM, Attn.-LSTM, bi-LSTM, Attn.-GRU, and Bi-GRU techniques, is one of the major contributions of our study.
3. Using the proposed CVPM, we conducted an analysis of the vibration prediction performance by progressively increasing the training segment of the concatenated vibrations from 40% to 90% in increments of 10%. These simulations allowed us to identify the training data segments where the prediction accuracy values of the six RNN techniques converged.

The remainder of this paper is organized as follows: Section 2 reviews previous research related to the prediction of time-series vibrations using various RNN techniques. In Section 3, the proposed CVPM is described, along with the description of the synthesized concatenated vibration dataset utilized to analyze the predictive performance of the six RNN techniques. Section 4 provides an insight into the techniques employed to predict concatenated vibrations, including LSTM, GRU, attention mechanism, and bidirectional techniques. Section 5 conducts a comprehensive analysis, comparing vibration prediction accuracy, simulation runtime, and the relationship between simulation runtime and predictive accuracy for the six RNN techniques. Finally, Section 6 summarizes the conclusions and describes future research.

2. Related Work

In recent years, machine learning algorithms have become popular due to the rapid development of software and hardware. As a result, research on predicting vibrations using machine learning and deep learning has been actively conducted. This section introduces studies related to fault detection and vibration prediction, highlighting their importance in Industry 4.0.

Wang et al. [31] introduce a fault diagnosis method for miniature vibration motors. This method combines wavelet packet decomposition with an improved three-layer LSTM network to enhance accuracy. ElSaid et al. [32] conducted a study on predicting excess vibration events in aircraft engines, a crucial aspect of the aviation industry. They utilized LSTM for accurate predictions, which yielded promising results with low error rates. Wang et al. [30] introduce a model for predicting and analyzing vibration severity in steam turbine rotor systems. This model combines sequence prediction with GRU and GRU-Seq2Seq to address the gradient disappearance problem, outperforming back-propagation (BP) and LSTM-Seq2Seq models. Hong [35] applied LSTM and GRU models to forecast vibrations based on time series motor data, comparing their accuracy and simulation runtime efficiency. This research indicates that GRU forecasts vibrations faster than LSTM.

Additionally, research has been conducted on predicting vibrations using LSTM, support vector machines (SVMs), artificial neural network (ANN), and RNN techniques. Xiao et al. [28] introduce a fault diagnosis method for three-phase asynchronous motors using RNNs. Experimental tests on six motors under different fault conditions demonstrate that this approach outperforms other methods, including logistic regression (LR), SVM, multi-layer perceptron (MLP), and basic RNN, in terms of fault diagnosis accuracy. Zhu et al. [33] applied the variational mode decomposition (VMD) technique to predict vibrations using SVM, ANN, and GRU techniques and conducted a comparative analysis of their performance. The results show that among these methods leveraging VMD, VMD-GRU demonstrated the most accurate predictive performance. Furthermore, Zhang et al. [26] introduced an LSTM-based fault detection and identification (FDI) method for quadcopter blades based on airframe vibration signals. This method outperforms the Back Propagation (BP) neural network-based FDI model, especially when dealing with larger volumes of vibration data.

Furthermore, studies have been conducted to predict performance by combining RNN and convolutional neural network (CNN) technologies. Li et al. [25] proposed a method that combines a CNN trained with acoustic emission signals and a GRU network trained with vibration signals, resulting in highly accurate gear pitting fault diagnosis with over 98% accuracy. The combination of CNN and GRU effectively leverages the advantages of both networks, demonstrating superior performance compared to using CNN or GRU individually.

Research has been conducted using Bidirectional RNN technology, which simultaneously processes input data in both forward and backward directions, considering information from both directions, to predict vibrations. Liang et al. [22] proposed a method to predict spindle rotation error that involves three key steps: data preprocessing, training a Bi-LSTM classification network, and predicting spindle rotation error. This method

demonstrates the effectiveness of this predictive approach. Adlen et al. [24] conducted research using LSTM, Bi-LSTM, and GRU to predict the condition of wind turbine operations based on vibration time series data. In this study, Bayesian optimization was used to fine-tune the training parameters. The results demonstrate that these models achieved more accurate predictions of wind turbine conditions compared to models trained with conventional parameters.

Additionally, studies predicting vibrations were conducted using Autoencoder and RNN techniques. Han et al. [23] used LSTM techniques to predict the remaining useful life (RUL) of bearings through an approach that involves two models: degradation state model and RUL prediction model. This approach utilizes a stacked autoencoder (SAE) to extract health indications (HIs) from selected features in the degradation state model and employs an LSTM for RUL prediction with standard deviation input and HI training labels. Huang et al. [29] introduced a two-stage machine learning architecture for accurate motor fault prediction based solely on motor vibration time-domain signals, avoiding complex preprocessing. In the first stage, they use the RNN-based Variational Encoder (VAE) method to reduce dimension and improve prediction accuracy. The second stage employs Principal Components Analysis (PCA) and Linear Discriminant Analysis (LDA) for further dimension reduction, enabling clear visualization and detection of different fault modes. This approach simplifies fault detection, reduces computational costs, and enhances classification accuracy.

A study predicting vibrations was conducted using RNN and Attention techniques together. Yang et al. [34] introduces Informer into time series forecasting of motor bearing vibration. Through random search for model parameter optimization, Informer successfully minimizes error accumulation, improving forecasting accuracy. Lee and Hong [36] analyzed that RNN models with attention mechanisms are the most suitable for predicting time series normal and abnormal vibration data, addressing that future research will focus on predicting vibrations in near real-time with coexisting normal and abnormal vibrations.

To sum up, most existing studies utilize RNN techniques to predict time series normal and abnormal vibrations separately. Therefore, the previous studies lack consideration of the transition of the vibration states from normal to abnormal when the actual motor is damaged. Hence, research predicting vibrations that reflect the transition from normal to abnormal vibration states is highly required.

3. Methodology

3.1. Proposed Concatenated Vibration Prediction Model

In this subsection, the proposed CVPM designed for forecasting concatenated vibrations using six distinct RNN techniques is described. The goal of the proposed CVPM is to comparatively analyze the prediction accuracy and simulation runtime of the six RNN techniques.

Figure 1 shows the flowchart of the proposed CVPM. According to Figure 1, it starts by collecting both normal and abnormal vibration data from the motors connected to the drone's propellers. Then, it generates concatenated vibration data by merging 50% of the normal vibration data with 50% of the abnormal vibration data. This concatenated vibration data represents the transition from the normal to abnormal vibration. Then, the concatenated vibration data is divided into training and testing datasets, with percentages ranging from 40% to 90%. The training data consists of 40%, 50%, 60%, 70%, 80%, and 90% of the concatenated vibration data, while the remaining portions, 60%, 50%, 40%, 30%, 20%, and 10%, form the testing dataset.

Subsequently, the proposed CVPM is employed to train on the segmented concatenated vibration data from 40%, 50%, 60%, 70%, 80%, and 90% segments using six different RNN techniques and to predict respectively the remaining 60%, 50%, 40%, 30%, 20%, and 10% segments. Finally, to assess predictive accuracy, the evaluation involves comparing and analyzing the coefficient of determination, R^2 values, among the predictions of the six RNN techniques and the corresponding testing dataset. This assessment is conducted

alongside an examination of the simulation runtimes for the six RNN techniques. By comparing R^2 values and simulation runtimes, the performance of each RNN technique in predicting concatenated vibrations is comprehensively analyzed. Subsequently, the most suitable RNN technique for concatenated vibration prediction is determined through this comparative analysis.

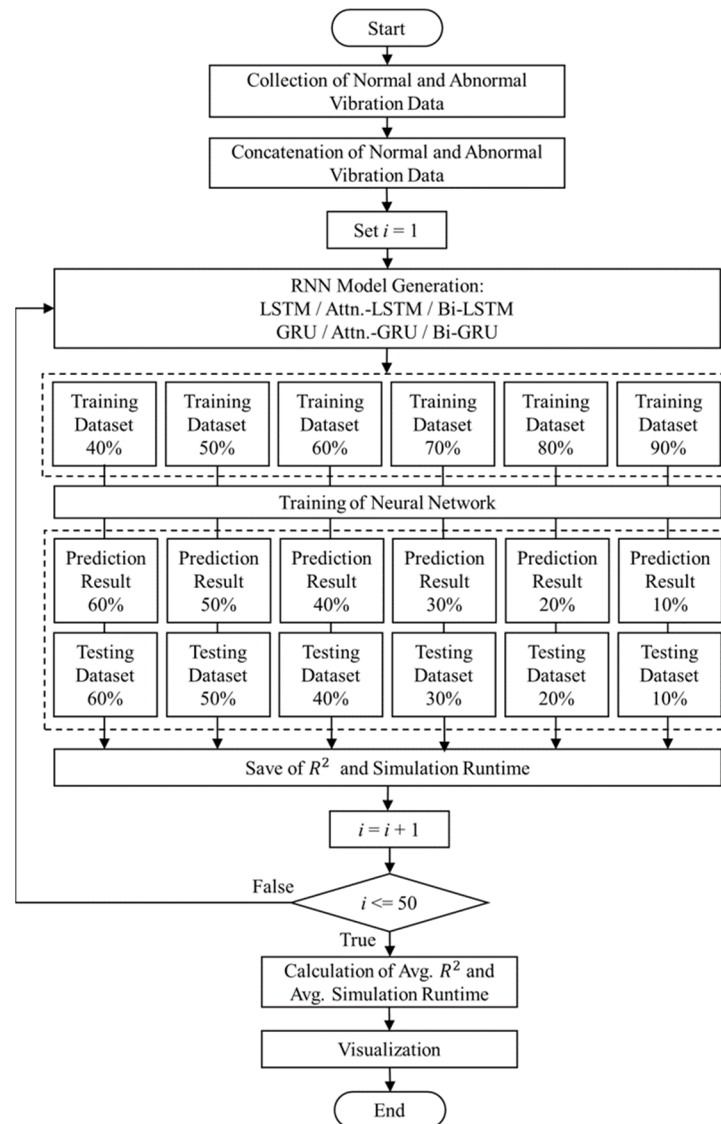


Figure 1. Flowchart of the proposed CVPM.

3.2. Generation of Concatenated Vibration Data

This section describes the methodology for collecting both normal and abnormal vibration data from the motors connected to an actual drone's propellers. Subsequently, the process of generating concatenated vibration data is explained, entailing the combination of 50% of the normal vibration data with 50% of the abnormal vibration data. This generation of the concatenated vibration dataset effectively models the transition from normal to abnormal vibrations.

Figure 2 shows the configuration for collecting time series vibration data using an accelerometer attached to a motor. Subsequently, vibration data within the 1 kHz frequency band is acquired over a duration of 100 milliseconds through an acceleration sensor affixed to the drone's motor.

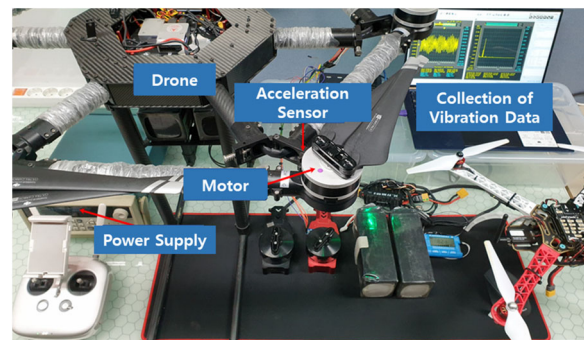


Figure 2. Configuration of collecting time series vibration data.

Figure 3a,b show the normal motor and the abnormal motor with damaged rotors, respectively. The motor in Figure 3 has a rated speed of 180 Kv, a maximum output power of 1474.6 W, and a maximum torque of 1.992 N·m. In order to collect vibration data from the normal motor and damaged motors, the rotation rate of the motors is set at 1200 revolutions per minute (RPM). Furthermore, the abnormal vibration data collected from the damaged motor were obtained after the vibrations had progressed to the advanced stage.

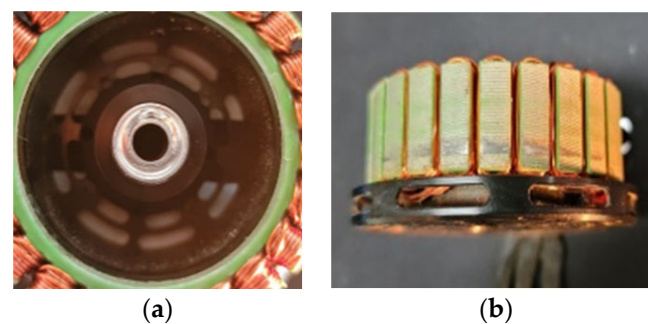


Figure 3. (a) Normal motor; (b) motor with a damaged rotor.

The collected normal vibration data and abnormal vibration data undergo min—max normalization, as represented by Equation (1), resulting in values between 0 and 1.

$$x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

where x and x_{norm} represent an individual vibration value from the collected vibration data and the normalized vibration value, respectively. Figure 4 (below) illustrates the waveforms of the normalized normal and abnormal vibration data.

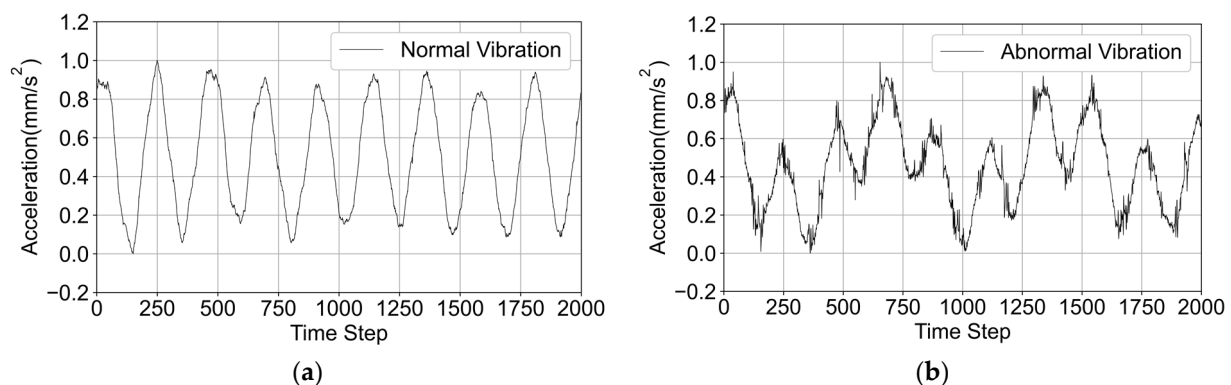


Figure 4. Waveform of the motor's vibrations: (a) normal motor; (b) abnormal motor.

According to Figure 4, the vibration waveform of the normal motor shows consistent wavelengths and amplitudes, while the waveform of the abnormal vibration data appears irregular with varying wavelengths and amplitudes compared to the normal vibration waveform. The abnormal vibration also contains more residual vibrations. Furthermore, since the abrupt transition from normal vibrations to abnormal vibrations due to the sudden damage to the motor is assumed, the vibration used for prediction is generated by concatenating normal and abnormal vibrations.

Figure 5 illustrates the waveform of the concatenated vibrations, created by concatenating 50% of normal vibration data with 50% of abnormal vibration data. By using the 50:50 ratio of normal and abnormal vibrations, as illustrated in Figure 5, the proposed CVPM can simulate predictions covering various vibration scenarios. This includes cases with normal vibrations only and the ones with a combination of normal and abnormal vibrations. The proposed CVPM achieves these scenarios by employing segments comprising 40%, 50%, 60%, 70%, 80%, and 90% of the vibration data for the learning segment.

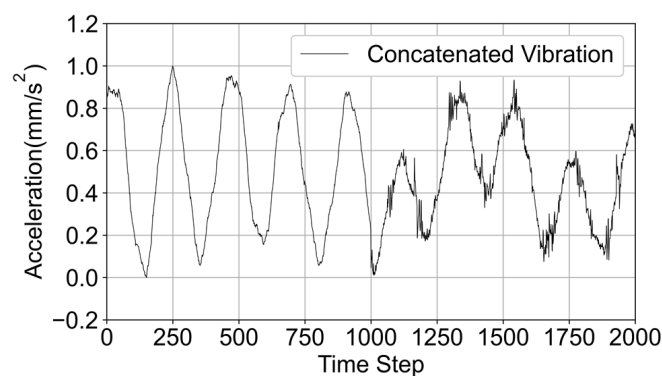


Figure 5. Waveform of a concatenated vibration from the normal and abnormal vibrations.

4. RNN

This section describes the usage of LSTM and GRU to predict the vibration, the attention mechanism applied to Attn.-LSTM and Attn.-GRU, and the bidirectional method applied to Bi-LSTM and Bi-GRU.

4.1. LSTM

RNN is a neural network model with a recurrent structure, capable of time series prediction by leveraging the characteristic where the output of the previous time step influences the output of the current time step. Nevertheless, conventional RNNs encounter the challenge of long-term dependency, resulting in information loss as the input sequence length grows. For example, when information from earlier time steps holds significant importance, the issue of long-term dependencies can have severe repercussions.

LSTM is a specialized type of RNN architecture designed to address the vanishing gradient problem and to capture long-term dependencies in sequential data [37]. LSTM networks are equipped with memory cells that can store information for extended periods, enabling them to effectively retain and utilize relevant information from previous time steps in a sequence. The distinct design of LSTM allows it to alleviate the issues of short-term memory limitations that traditional RNNs face. By allowing the network to train when to store, read, or erase information, LSTM can capture intricate dependencies and patterns in various types of sequential data, such as time series, natural language, and speech.

In the architecture of an LSTM cell, there are three essential gates: forget gate, input gate, and output gate. These gates play a crucial role in managing the flow of information through the cell, which is essential for capturing long-term dependencies in sequential data. Figure 6 below shows the architecture of LSTM.

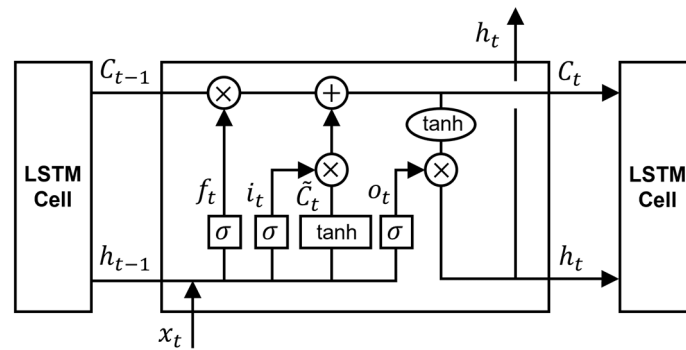


Figure 6. LSTM architecture.

In Figure 6, C , $\tilde{C}$, h , t , and x refer to the cell state, the candidate values to be added to the cell state, the output value, the time step, and the input value, respectively. Moreover, f , i , and o represent the forget gate, the input gate, and the output gate, correspondingly. Equations (2)–(7) below represent the expressions for the variables that constitute the LSTM.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \quad (2)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad (3)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \quad (4)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \quad (5)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \quad (6)$$

$$h_t = o_t \odot \tanh(C_t) \quad (7)$$

Equation (2), Equation (3), and Equation (4) represent the formulations of the forget gate, the input gate, and the output gate, respectively. Equations (5)–(7) elucidate the LSTM cell's process of determining the cell state and hidden state. In Equation (2), Equation (3), Equation (4), and Equation (5), the symbols W and b represent the trainable weight matrix and the bias vector, respectively.

Equation (2) represents the output of the forget gate, which determines the information to be forgotten in the cell state. This gate plays a crucial role in deciding what information should be retained or discarded within the cell state. The forget gate receives two inputs: the input for the hidden state from the previous time step, h_{t-1} , and the input for the current time step, x_t .

Equation (3) represents the output of the current time step's input gate, which determines the information to be newly added to the cell state at the current time step. The value of the input gate at the current time step is calculated through a neural network operation involving the previous time step's hidden state, h_{t-1} , and the current time step's input, x_t , as inputs.

Equation (4) represents the output gate's output, which influences the calculation of the current time step's hidden state, h_t . The current time step's output gate value is determined by the previous time step's hidden state, h_{t-1} , and the current time step's input, x_t , as inputs to the neural network operation.

Equation (5), $\tilde{C}_t$, represents the candidate cell state at the current time step, t . This candidate cell state is calculated based on the input value at the current time step, x_t , and the previous time step's hidden state, h_{t-1} . W_C denotes the trainable weight matrix, which determines how to combine the previous time step's hidden state, h_{t-1} , and the current time step's input, x_t . $[h_{t-1}, x_t]$ represents the vector obtained by concatenating the previous

time step's hidden state and the current time step's input. LSTM employs this mechanism to selectively add and update information, enabling it to capture long-term dependencies in the sequential data effectively.

Equation (6), C_t , represents the cell state at the current time step, storing information learned by the network and retaining valuable information for the current time step. f_t denotes the forget gate at the current time step, t . The forget gate determines how much of the previous cell state, C_{t-1} , to retain or to forget. i_t represents the input gate at the current time step, t . The input gate determines how much of the new information, $\tilde{C}_t$, to add to the current cell state. $\tilde{C}_t$ represents the candidate cell state at the current time step, calculated based on the current input and the previous hidden state, representing new information that can be added to the current cell state. Therefore, the current cell state, C_t , is updated by combining what should be forgotten from the previous cell state, denoted as $f_t \odot C_{t-1}$, with the new information to be included, represented as $i_t \odot \tilde{C}_t$.

Equation (7) represents the process of calculating the hidden state, h_t , at the current time step, t , in LSTM. This equation determines which part of the cell state, C_t , at the current time step, t , as decided by the output gate, o_t , will be used and compresses this information using the hyperbolic tangent function to calculate the hidden state, h_t , at the current time step. Through this process, LSTM updates the hidden state at the current time step, combining the previous information with the new information to capture long-term dependencies in the sequential data.

4.2. GRU

GRU is a technique designed to simplify the structure of LSTM in order to reduce computational time [38]. GRU offers a streamlined architecture compared to traditional LSTM while still retaining the ability to capture long-term dependencies in sequential data. Figure 7 below illustrates the architecture of GRU.

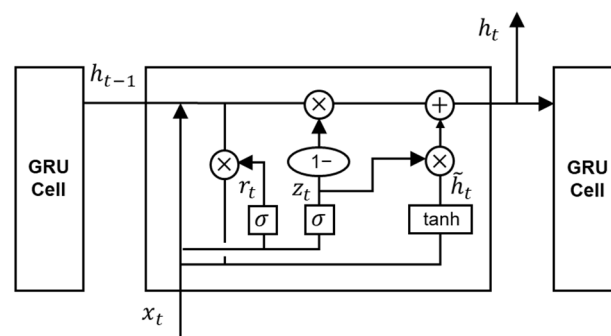


Figure 7. GRU architecture.

Figure 7 presents the architecture of GRU, equipped with gating mechanisms that regulate the flow of information through the network. It comprises two gates: reset gate and update gate. The reset gate determines the extent to which the previous hidden state is combined with the new input, allowing the model to decide what information to discard and what to keep, while the update gate controls how much of the previous hidden state should be retained and how much of the new information should be added to the current hidden state.

In Figure 7, $\tilde{h}$, h , t , and x denote the candidate output value, the output value, the time step, and the input value, respectively. The variables r and z correspond to the reset gate and the update gate, respectively. Furthermore, Equations (8) and (9) below represent the formulas for the reset gate and the update gate, while Equations (10) and (11) describe how the output value of the GRU cell is determined.

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]), \quad (8)$$

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]), \quad (9)$$

$$\tilde{h}_t = \tanh(W \cdot [r_t \odot h_{t-1}, x_t]) \quad (10)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (11)$$

Equation (8) represents the output of the current time step's reset gate, which determines the strength of incorporating the previous time step's hidden state, h_{t-1} , into the current time step's output with a trainable weight matrix, W_r . The current time step's reset gate value is calculated based on the previous time step's hidden state and the current time step's input, x_t , through a neural network operation. Since the reset gate's activation function is a sigmoid function, the reset gate has values between 0 and 1.

Equation (9) depicts the process of computing the update gate in GRU. z_t represents the value of the update gate at the current time step, t , and the update gate is determined by the previous time step, h_{t-1} , and the current time step's input, x_t . σ denotes the sigmoid activation function, which compresses the input values into the range between 0 and 1. Therefore, Equation (9) represents a convolution operation processed by the sigmoid function. W_z represents a trainable weight matrix. This matrix plays a role in determining how to combine the input values, $[h_{t-1}, x_t]$.

Equation (10) represents the candidate value to be added from the previous time step to the current time step. It uses the hyperbolic tangent function as its activation function and involves element-wise multiplication between the previous time step's hidden state, h_{t-1} , and the current time step's reset gate value, r_t , along with the neural network operation's result, based on the current time step's input, x_t , with a trainable weight matrix, W .

Equation (11) represents the hidden state for the current time step. According to Equation (11), element-wise multiplication is performed between the value obtained by subtracting the value of the current time step's update gate, z_t , from 1, and the previous time step's hidden state value, h_{t-1} . Additionally, element-wise multiplication is carried out with the value to be added for the current time step, $\tilde{h}_t$, and the update gate's value. As a result, the ratio between the information to be forgotten from the previous time step and the information to be added from the current time step is determined, thus influencing the calculation of the hidden state for the current time step.

4.3. Attention Mechanism

The attention mechanism is a pivotal component in neural network architectures, devised to enhance the processing of sequential data by selectively focusing on relevant information. The attention mechanism enables the network to dynamically allocate varying degrees of attention to different parts of the input sequence, effectively adapting its processing based on the context and importance of each element [39]. Figure 8 below shows the architecture of attention mechanism.

In Figure 8, c , e_i , h , M , $\tilde{s}$, x , and $\hat{y}$ represent the attention value, the set of attention scores, the hidden state, the length of the input sequence, the input of the output layer, the input, and the predicted value, respectively.

$$e_i = s^T W_c h_i \quad (12)$$

$$\alpha_i = \text{softmax}(e_i) \quad (13)$$

$$c = \sum_{i=1}^M \alpha_i h_i \quad (14)$$

$$\tilde{s} = \tanh(W_c [c; s]) \quad (15)$$

$$\hat{y} = \text{softmax}(W_y \tilde{s} + b_y) \quad (16)$$

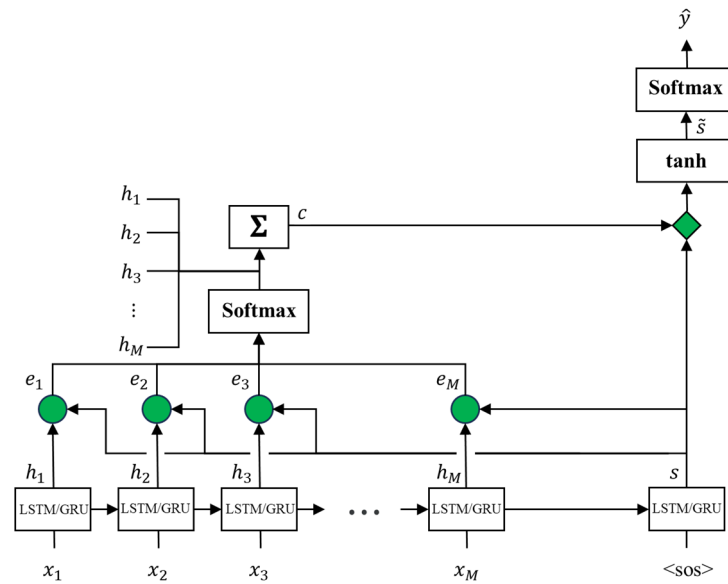


Figure 8. Architecture of LSTM and GRU with the attention mechanism.

Equation (12), Equation (15), and Equation (16) represent the i th attention score, the i th attention distribution, the attention value, and the input and prediction values of the output layer, respectively.

Equation (12) calculates the i th attention score by multiplying the transpose of the decoder's hidden state, s^T , with a trainable weight matrix, W_c , and the i th hidden state of the encoder, h_i .

Equation (13) calculates the i th element that makes up the attention distribution. The Softmax function is applied to the attention scores in Equation (13) to calculate an attention distribution, which is a probability distribution with a total sum of 1. Each element of the attention distribution signifies the importance of each respective encoder's hidden state.

Equation (14) calculates the attention value by summing the products of the i th attention distribution and the corresponding hidden state. Therefore, the attention value signifies the importance of the encoder's hidden states.

Equation (15) updates the hidden state, denoted as $\tilde{s}$, representing the updated hidden state candidate, indicating how new information is incorporated into the current hidden state.

Equation (16) represents the final step in the attention mechanism, where the predicted output, $\hat{y}$, is computed. W_y denotes the trainable weight matrix, which is applied to the updated hidden state candidate, $\tilde{s}$, determining its contribution to generating predictions. $\tilde{s}$ represents the updated hidden state candidate, computed in Equation (15), encapsulating the model's current state, including the information processed up to this point. b_y stands for the bias vector added to the product of W_y and $\tilde{s}$.

Therefore, the attention mechanism enhances prediction performance by assigning varying degrees of importance to the constituent values of the input sequence, thus elevating the weights of crucial information while attenuating the weights of less significant details when calculating prediction values.

4.4. Bidirectional RNN Techniques

In a standard RNN, information flows only from the past to the present, which means that the current time step's prediction depends solely on past time steps. However, the bidirectional RNN overcomes this limitation by introducing two separate hidden

layers in both directions: one capturing information from past time steps in the forward direction, and the other capturing information from future time steps in the backward direction [40]. In this study, the Bi-LSTM and Bi-GRU models, which incorporate the bidirectional technique, are employed and applied to the LSTM and GRU architectures. Figure 9 below illustrates the process through which Bi-LSTM and Bi-GRU generate the predicted values.

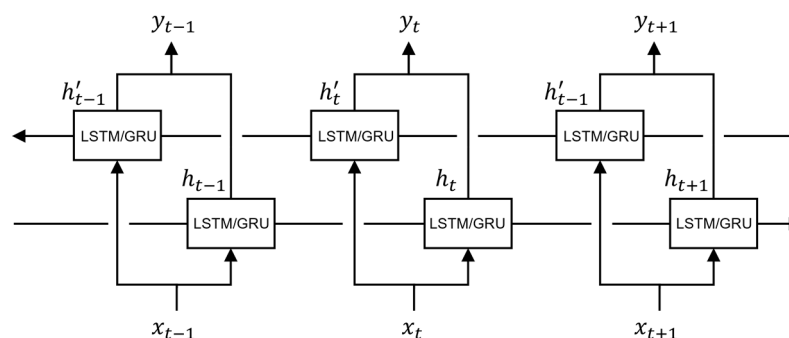


Figure 9. Architecture of Bi-LSTM and Bi-GRU.

In Figure 9, h , h' , t , y , and x represent the forward hidden state, backward hidden state, time step, output, and input respectively. As illustrated in Figure 9, the application of the bidirectional technique to LSTM and GRU introduces two RNN layers, each dedicated to learning information from the forward and backward contexts. This stands in contrast to the single LSTM and GRU models, which consider only unidirectional information. Consequently, when forecasting time series data, bidirectional models leverage information from both directions. Therefore, due to the bidirectional training process in Bi-LSTM and Bi-GRU, both models take a longer training time compared to the conventional LSTM or GRU models.

5. Simulation Results and Discussion

In this section, a comparative analysis of predictive accuracy and simulation runtime among the six RNN techniques is conducted using the proposed CVPM.

5.1. Simulation Environment and Parameters

This section describes the simulation environment and parameters used to predict concatenated vibrations using the six RNN techniques. Table 2 below describes the simulation environment.

Table 2. Simulation environment.

Parameter	Specification
CPU	Intel Xeon CPU @ 2.00 GHz
GPU	Nvidia Tesla T4
Memory	26 GB
Simulation tools	Google Colaboratory Pro, Python 3.10.12 and Tensorflow 2.12.0

In addition, Table 3 below provides an explanation of the parameters configured for the simulation.

In Table 3, the number of hidden units is set to 32, the initial learning rate is set to 0.0002, and the number of epochs is set to 10. These settings are chosen for a clear observation of the performance variations when predicting concatenated vibrations using the six RNN techniques. Additionally, the batch size is configured as 32, aligning with the recommended range of 32 to 128 [41]. Furthermore, the adaptive moment estimation (Adam) algorithm is used as an optimization algorithm to minimize training errors and avoid local minima problems in the simulations [42].

Table 3. Simulation parameters.

Parameter	Specification
No. of hidden units	32
Initial learning rate	0.0002
No. of epochs	10
Minimum batch size	32
Optimizer	Adam [42]
Iterations	50

5.2. Waveform of Predicted Vibrations

This section explains the predicted vibration waveforms when using the six RNN techniques to train on 40%, 50%, 60%, 70%, 80%, and 90% of concatenated vibration data and to predict the remaining 60%, 50%, 40%, 30%, 20%, and 10%, respectively. The coefficient of determination, R^2 , for comparing the predictive accuracy of the six RNN techniques in the proposed CVPM is calculated as follows:

$$R^2 = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (17)$$

where n , y , $\hat{y}$, and $\bar{y}$ denote the total number of vibration data values, the actual vibration values, the predicted vibration values, and the average vibration values, respectively.

Figure 10a–f represents the waveforms of the vibrations predicted using the six RNN techniques, where each technique is used to train on 40%, 50%, 60%, 70%, 80%, and 90% segments of concatenated vibration data and to predict the remaining 60%, 50%, 40%, 30%, 20%, and 10% segments.

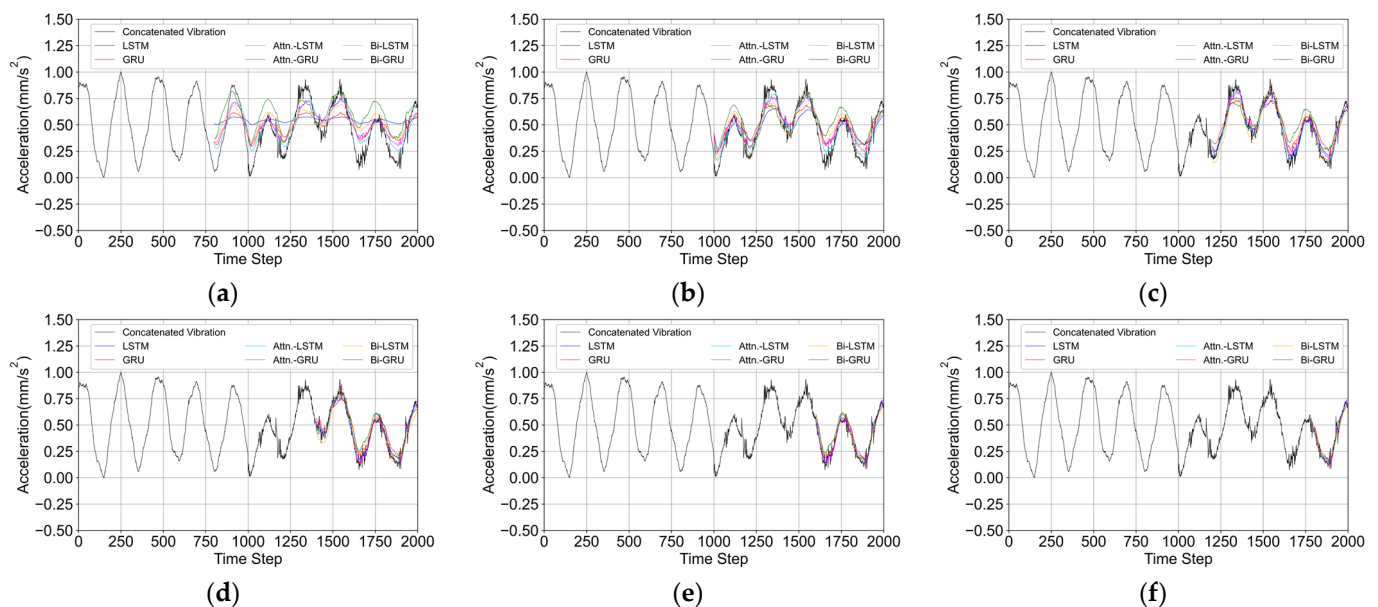


Figure 10. Waveforms of the vibrations predicted from concatenated vibrations: (a) 40%, (b) 50% (c) 60%, (d) 70%, (e) 80%, and (f) 90% segments of training data.

Figure 10a shows the predicted waveform of the vibrations after learning on 40% training data using the six RNN techniques. Since the training segment contains only normal vibration data in the 40% training segment, it is impossible to train abnormal vibration. Therefore, the average R^2 value of the predicted vibrations using the six RNN techniques is 0.49, which is very low.

According to Figure 10b, similar to the 40% training segment, the 50% training segment cannot learn abnormal vibration data either. However, since more information about the amplitudes and periods of vibrations is learned compared to the 40% training segment, the average R^2 value of the six RNN techniques is 0.75. This indicates an increase in prediction accuracy of approximately 53.06% compared to the average R^2 value of the 40% training segment.

Figure 10c shows the predicted vibration waveform after learning 60% training segment, where 50% of normal vibration data and 10% of abnormal vibration data are trained. Therefore, since both normal and abnormal vibrations are trained, the average R^2 value of the six RNN techniques is 0.88, representing an increase of approximately 79.59% compared to the average R^2 value in the 40% training segment. Furthermore, as observed in Figure 10d, since 20% of abnormal vibration data is trained in the 70% training segment, the average R^2 value of the prediction accuracy of the six RNN techniques is 0.92. This R^2 value represents an approximately 87.76% increase compared to the R^2 value observed in the 40% training segment.

Also, as shown in Figure 10d, since 20% of abnormal vibration data is trained in the 70% training segment, the average R^2 value of prediction accuracy of the six RNN techniques is 0.92. This R^2 value is an increase of approximately 87.76% compared to the R^2 value in the 40% segment. Furthermore, according to the simulation results in Figure 10e,f, where the abnormal vibration data is trained at 30% and 40%, respectively, it is observed that the R^2 values increased by approximately 85.71% and 93.88%, respectively, compared to the R^2 value of the 40% training segment.

In short, according to Figure 10, all of the six RNN techniques demonstrate incremental improvement in prediction accuracy as the training segment size increases. In the next section, the predicted and actual vibration values from the six RNN techniques will be comparatively analyzed using scatter plots.

5.3. Comparison of Scatter Plots

In this section, the vibration prediction accuracy of the six RNN techniques is comparatively analyzed based on changes in the training segment using scatter plots.

Figure 11 shows the scatter plot of the predicted vibration data using the six RNN techniques and the actual vibration data. According to Figure 11a,b, after the six RNN techniques are trained on the 40% and 50% training segments, the predicted vibration values notably differ from the actual vibration values. Specifically, the predicted vibration accuracy using LSTM is significantly lower compared to the predicted vibration accuracy from the other five RNN techniques. According to Figure 11c, the vibration results predicted by the six RNN techniques gradually converge toward the actual concatenated vibration value as the abnormal vibration data is trained from the 60% segment.

Additionally, as shown in Figure 11d–f, the predicted vibration values match closely with the actual vibration values when predicting vibrations from the 70% training segment. These simulation results are attributed to the inclusion of more abnormal vibration data during training when the learning segments exceed 70%.

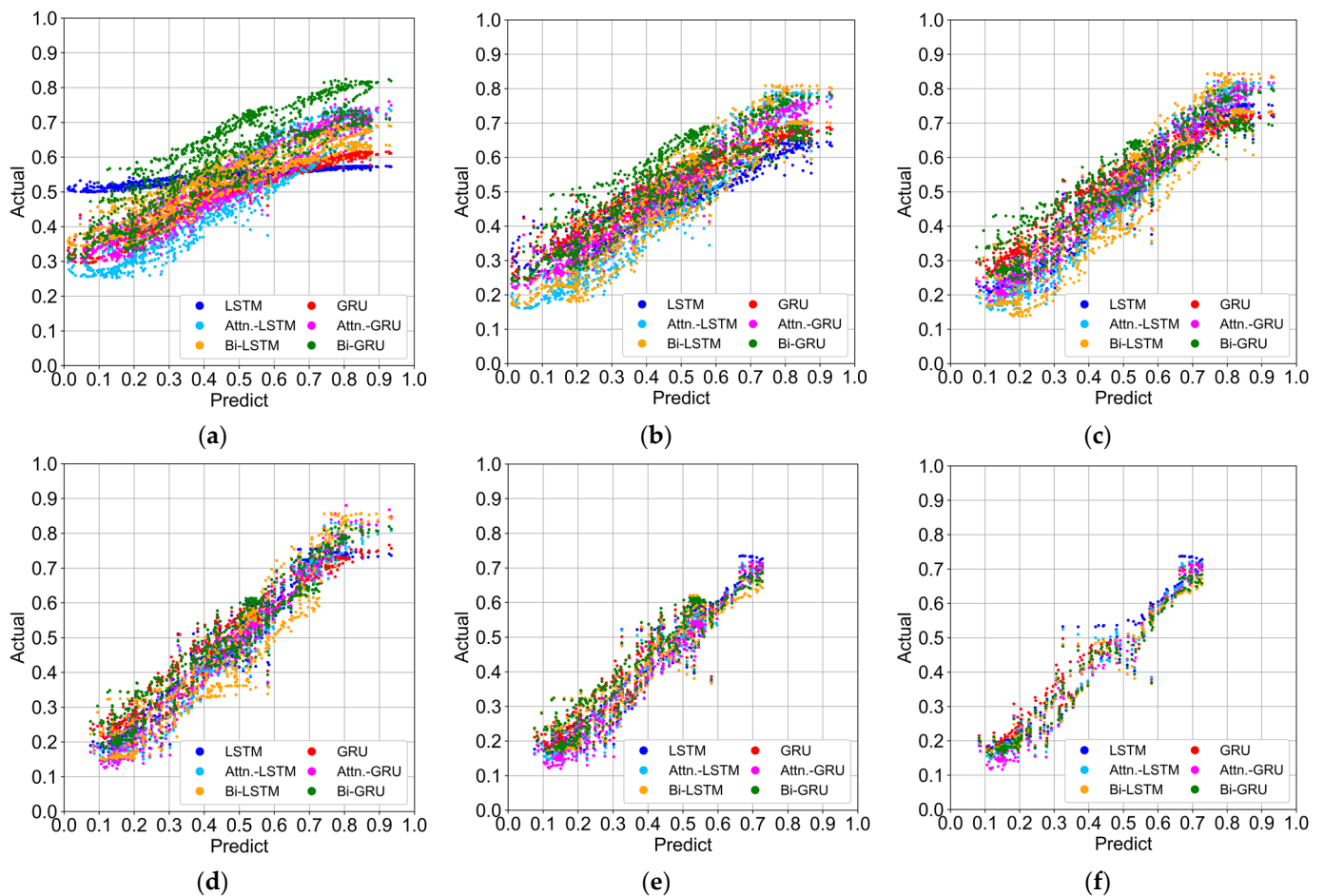


Figure 11. Scatter plots with respect to training segment changes: (a) 40%, (b) 50% (c) 60%, (d) 70%, (e) 80%, and (f) 90% segments of training data.

5.4. Comparative Analysis of Concatenated Vibration Prediction Accuracy

In this section, the changes in R^2 values of the six RNN techniques are analyzed as the training data segment increases from 40% to 90%. Figure 12 below illustrates the R^2 values of the six RNN in relation to the changes in the training segment.

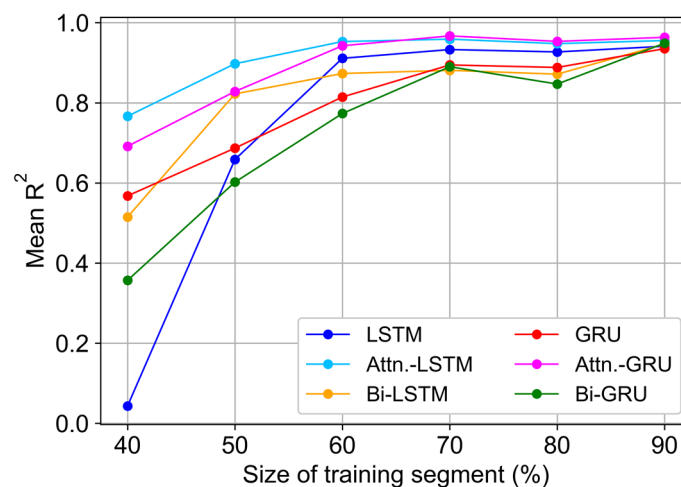


Figure 12. Average value of the coefficient of determination according to the size of the training segment.

According to Figure 12, it is observed that the R^2 values predicted by the six RNN techniques increase as the training segment increases from 40% to 90%. Furthermore,

it is noted that the accuracy of the vibrations predicted by Attn.-LSTM and Attn.-GRU, both utilizing the attention mechanism, is the highest among all the training segments, surpassing the respective groups of LSTM and GRU.

Furthermore, in the 40% training data segment, the vibration prediction results obtained using LSTM and GRU are similar to or lower than those obtained using Bi-LSTM and Bi-GRU. However, as the training segment increases, the R^2 values of LSTM and GRU show higher accuracy compared to those of Bi-LSTM and Bi-GRU. Table 4 below presents the R^2 values for each training segment of the six RNN techniques.

Table 4. R^2 values of six RNN techniques as the training segment changes.

Model	Size of Training Segment						Avg.
	40%	50%	60%	70%	80%	90%	
LSTM	0.04	0.66	0.91	0.93	0.93	0.94	0.74
Attn.-LSTM	0.77	0.9	0.95	0.96	0.95	0.96	0.92
Bi-LSTM	0.52	0.82	0.87	0.88	0.87	0.95	0.82
GRU	0.57	0.69	0.81	0.89	0.89	0.94	0.80
Attn.-GRU	0.69	0.83	0.94	0.97	0.95	0.96	0.89
Bi-GRU	0.36	0.60	0.77	0.89	0.85	0.95	0.74
Avg.	0.49	0.75	0.88	0.92	0.91	0.95	-
Change rate (%)	-	+53.06	+17.33	+4.55	−1.09	+4.40	-

According to Table 4, Attn.-LSTM and Attn.-GRU achieved the highest prediction accuracy in all the training segments, while Bi-GRU indicated lower accuracy compared to GRU. Moreover, in the 40% training segment, the prediction accuracy of LSTM and GRU is similar to or lower than that of Bi-LSTM and Bi-GRU, but from the 60% training segment, the prediction accuracy of LSTM and GRU is higher than that of Bi-LSTM and Bi-GRU. These simulation results indicate that when the training segment is 40% and 50%, LSTM, GRU, Bi-LSTM, and Bi-GRU face the common challenge of having a small training data size and are unable to effectively learn from abnormal vibrations. However, within the same training segment, Bi-LSTM and Bi-GRU learn more information about the vibration's amplitude and frequency compared to LSTM and GRU.

Nevertheless, starting from the 60% training segment, LSTM and GRU undergo sufficient learning on amplitude and frequency information as well, similar to Bi-LSTM and Bi-GRU techniques. Meanwhile, Bi-LSTM and Bi-GRU overlearn normal vibrations excessively, which can act as interfering factors when predicting abnormal vibrations. As a result, the prediction accuracy of Bi-LSTM and Bi-GRU becomes lower than that of LSTM and GRU.

In the next section, we will conduct a detailed comparative analysis of the simulation runtime required by the six RNN techniques in order to predict vibrations.

5.5. Comparison of Simulation Runtimes

In this section, the simulation runtimes required for the six RNN techniques to predict concatenated vibrations are analyzed comparatively. Table 5 below presents the average simulation runtimes for each training segment.

According to Table 5, the simulation runtimes of all six RNN techniques increase with the training segment size. Furthermore, LSTM and GRU techniques indicate faster simulation speeds compared to Attn.-LSTM, Bi-LSTM, Attn.-GRU, and Bi-GRU techniques across all the training segments.

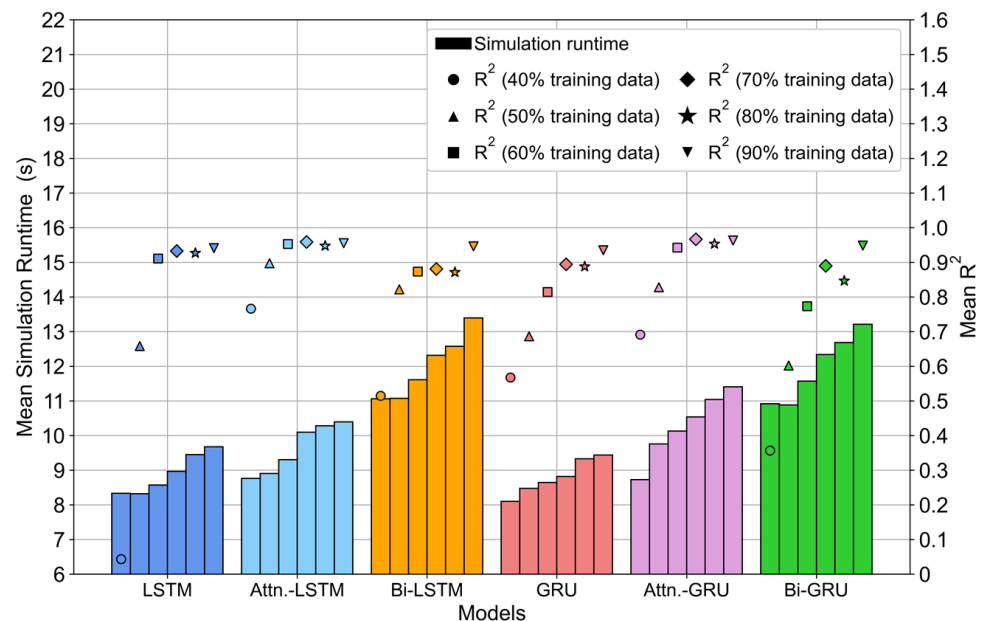
However, as evident from Figure 12 and Table 4, in the 40% training segment, LSTM exhibits very fast simulation runtime but the lowest predictive accuracy among the six RNN techniques. Therefore, it is important to compare simulation runtime and prediction accuracy comprehensively. Therefore, in Section 5.6, a comparison and analysis of the simulation runtime and predictive accuracy of the six RNN techniques are conducted to analyze their prediction performance comprehensively.

Table 5. Average simulation runtime (s) by training segment.

Model	Size of Training Segment						Avg.
	40%	50%	60%	70%	80%	90%	
LSTM	8.34	8.32	8.57	8.97	9.45	9.67	8.89
Attn.-LSTM	8.76	8.90	9.30	10.10	10.28	10.40	9.62
Bi-LSTM	11.06	11.08	11.61	12.32	12.58	13.4	12.01
GRU	8.10	8.48	8.65	8.82	9.33	9.44	8.80
Attn.-GRU	8.73	9.76	10.13	10.54	11.04	11.41	10.27
Bi-GRU	10.92	10.88	11.57	12.34	12.69	13.21	11.94

5.6. Comparative Analysis of Accuracy Efficiency

In this section, a comparative analysis is conducted by simultaneously evaluating the simulation runtimes and prediction accuracy of the six RNN techniques. This assessment aims to determine the RNN techniques that demonstrate the best performance in predicting concatenated vibrations. Figure 13 below shows a comparison of concatenated vibration prediction accuracy and simulation runtime of the six RNN techniques.

**Figure 13.** Comparison of the average value of the coefficient of determination for each RNN model according to the average simulation runtime.

In Figure 13, the bars represent the simulation runtime, and the circle, triangle, square, diamond, star, and inverted triangle represent the R^2 values of the 40%, 50%, 60%, 70%, 80%, and 90% training segment, respectively.

According to Figure 13, all six RNN techniques show an increase in simulation runtime as the training segment extends from 40% to 90%. Furthermore, the R^2 values increase with the expansion of the training segment, converging at approximately 70%. Additionally, among the converged R^2 values, Attn.-LSTM and Attn.-GRU show the highest R^2 values, while Bi-LSTM and Bi-GRU show the lowest R^2 values. In terms of simulation runtimes, LSTM and GRU are the shortest while Bi-LSTM and Bi-GRU are the longest.

Table 6 below represents an analysis of the prediction accuracy improvement rates and the simulation runtime changes of the remaining five RNN techniques relative to LSTM, which presents the lowest average prediction accuracy.

Table 6. Change rates in simulation runtime (s) and prediction accuracy (%) in comparison to LSTM.

Model	Size of Training Segment											
	40%		50%		60%		70%		80%		90%	
	Sim. r.t.	R^2	Sim. r.t.	R^2	Sim. r.t.	R^2	Sim. r.t.	R^2	Sim. r.t.	R^2	Sim. r.t.	R^2
Attn.-LSTM	+5.13	+1681.40	+6.99	+36.47	+8.55	+4.61	+12.63	+2.79	+8.83	+2.27	+7.45	+1.49
Bi-LSTM	+32.70	+1097.67	+33.09	+24.92	+35.48	−4.17	+37.40	−5.57	+33.10	−6.04	+38.40	+0.64
GRU	−2.81	+1220.93	+1.86	+4.41	+0.90	−10.65	−1.61	−4.18	−1.26	−4.21	−2.44	−0.64
Attn.-GRU	+4.71	+1506.98	+17.24	+25.84	+18.16	+3.40	+17.56	+3.64	+16.80	+2.81	+17.93	+2.34
Bi-GRU	+31.01	+730.23	+30.77	−8.51	+35.01	−15.15	+37.64	−4.61	+34.25	−8.63	+36.59	+0.85

According to Table 6, within the 40% training segment, the R^2 values for Attn.-LSTM, Bi-LSTM, GRU, Attn.-GRU, and Bi-GRU exhibit a minimum accuracy increase of approximately 730% compared to LSTM. Furthermore, Attn.-LSTM, Attn.-GRU, Bi-LSTM, and Bi-GRU exhibit longer simulation runtimes compared to LSTM and GRU across all the training segments due to the incorporation of additional attention mechanisms and bidirectional methods applied to the LSTM and GRU models. However, as shown in Table 6, both Attn.-LSTM and Attn.-GRU exhibit an increase in vibration prediction accuracy compared to LSTM across all the training segments. In contrast, it is observed that the vibration prediction accuracy of Bi-LSTM and Bi-GRU is slightly lower than that of LSTM in the 60%, 70%, and 80% training segments. Whereas LSTM and GRU are unidirectional, because Bi-LSTM and Bi-GRU are bidirectional, they tend to learn abnormal vibrations more, which act as distractors while predicting vibrations in both directions.

However, the average R^2 values for the entire training segments of Bi-LSTM and Bi-GRU increased compared to the R^2 values of LSTM. Therefore, in Table 7, the prediction accuracy and simulation runtime of LSTM are compared with the other five RNN techniques.

Table 7. Change rates in average simulation runtime (s) and prediction accuracy (R^2) compared to LSTM.

Model	Avg. Sim. Runtime	Avg. R^2
Attn.-LSTM	+8.26	+288.17
Bi-LSTM	+35.03	+184.58
GRU	−0.89	+200.94
Attn.-GRU	+15.40	+257.50
Bi-GRU	+34.21	+115.70

Table 7 presents the change rates in average simulation runtimes and average R^2 values for the five RNN techniques compared to those of LSTM. According to the findings in Table 7, Attn.-LSTM shows an 8.26% increase in simulation runtime compared to LSTM, along with a 288.17% accuracy improvement. Similarly, Attn.-GRU exhibits a 15.40% average increase in simulation runtime compared to LSTM, with an accompanying 257.50% accuracy enhancement.

Furthermore, it is observed that Attn.-LSTM and Attn.-GRU show smaller increases in simulation runtimes compared to Bi-LSTM and Bi-GRU, and yet they achieved a more substantial increase in accuracy. Hence, Attn.-LSTM and Attn.-GRU exhibit high prediction accuracy in comparison to their shorter simulation runtimes. The lower prediction accuracy of Bi-LSTM and Bi-GRU in longer simulation runtimes can be attributed to the following reasons.

Attn.-LSTM and Attn.-GRU, with their attention mechanisms designed to focus on information highly correlated with the prediction target in the training data, achieve a higher prediction accuracy when there is a low proportion of abnormal vibrations in the prediction segment. This is due to their ability to concentrate on training information relevant to abnormal vibrations, resulting in high prediction accuracy despite shorter simulation runtimes.

However, Bi-LSTM and Bi-GRU, which employ bidirectional learning based on LSTM and GRU, not only transmit information from the previous time steps to the following ones, but also learn from the backward time steps. This bidirectional learning leads to an excessive focus on learning from normal vibrations that act as distractors when predicting abnormal vibrations within the prediction segment. Consequently, while attempting to predict abnormal vibrations, Bi-LSTM and Bi-GRU may experience significant increases in simulation runtime and subsequently demonstrate lower prediction accuracy.

Consequently, based on the simulation results, it is established that among the six RNN techniques employed for concatenated vibration data prediction, Attn.-LSTM and Attn.-GRU are the most suitable RNN techniques, while Bi-LSTM and Bi-GRU exhibit the least suitability.

6. Conclusions

In this study, a CVPM is proposed for predicting concatenated vibrations, and a comparative analysis of vibration prediction accuracy and simulation runtime is conducted among the six RNN techniques: LSTM, Attn.-LSTM, Bi-LSTM, GRU, Attn.-GRU, and Bi-GRU. The concatenated vibration data used in this study are collected from the motors connected to the drone's propellers, including both normal and abnormal vibration data. Then, the concatenated vibration dataset is created by merging 50% of normal vibration data with 50% of abnormal vibration data. Subsequently, this vibration dataset is utilized to comprehensively analyze the prediction accuracy and simulation runtime of the six different RNN techniques using the proposed CVPM.

According to the simulation results, it is observed that as the training segment increase, the predictive accuracies and simulation runtimes also increase for all the RNN techniques, including LSTM, Attn.-LSTM, Bi-LSTM, GRU, Attn.-GRU, and Bi-GRU. Additionally, when predicting concatenated vibrations, Attn.-LSTM and Attn.-GRU, which utilize the attention mechanism, exhibit the highest predictive accuracy relative to simulation runtime. On the other hand, the Bi-LSTM and Bi-GRU models, which employ the bidirectional technique, exhibit the lowest predictive accuracy relative to simulation runtime. Therefore, it is determined that Attn.-LSTM and Attn.-GRU are the most suitable RNN models for predicting concatenated vibrations.

In future work, we plan to predict vibrations transitioning smoothly from normal to abnormal states by incorporating the early and advanced stages during the transition from normal to abnormal vibrations, applied to an actual vibration prediction system.

Author Contributions: Conceptualization, J.-K.H.; methodology, J.-K.H.; software, J.-H.L. and J.-K.H.; validation, J.-H.L. and J.-K.H.; formal analysis, J.-H.L. and J.-K.H.; writing—original draft preparation, J.-H.L. and J.-K.H.; writing—review and editing, J.-H.L. and J.-K.H.; visualization, J.-H.L.; supervision, J.-K.H.; funding acquisition, J.-K.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the research grant of Kongju National University in 2023.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

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

References

1. Kong, F.; Li, J.; Jiang, B.; Wang, H.; Song, H. Trajectory Optimization for Drone Logistics Delivery via Attention-Based Pointer Network. *IEEE Tran. Intell. Transp. Syst.* **2023**, *24*, 4519–4531. [[CrossRef](#)]
2. Agha-mohammadi, A.-A.; Ure, N.K.; How, J.P.; Vian, J. Health Aware Stochastic Planning for Persistent Package Delivery Missions Using Quadrotors. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 3389–3396.
3. Luo, L.; Zhang, L.; Wang, S.; Zeng, Y. Research on Intelligent Delivery based on UAVs. In Proceedings of the 2021 IEEE 2nd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), Chongqing, China, 17–19 December 2021; pp. 139–144.

4. Khan, M.M.; Ali, Q. Simulation and Development of an Autonomous Drone for Delivery of Medicines During COVID-19. In Proceedings of the 2022 19th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, Pakistan, 16–20 August 2022; pp. 153–160.
5. Sanjana, P.; Prathilothamai, M. Drone Design for First Aid Kit Delivery in Emergency Situation. In Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 March 2020; pp. 215–220.
6. Tinetti, F.G.; Riveros, O.C.V. Combining Technologies for Aiding Search Missions with Drones. In Proceedings of the 2021 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 15–17 December 2021; pp. 1617–1622.
7. Yao, P.; Xie, Z.; Ren, P. Optimal UAV Route Planning for Coverage Search of Stationary Target in River. *IEEE Trans. Control Syst. Technol.* **2019**, *27*, 822–829. [\[CrossRef\]](#)
8. Gupta, L.; Jain, R.; Vaszkun, G. Survey of Important Issues in UAV Communication Networks. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1123–1152. [\[CrossRef\]](#)
9. Ajgaonkar, K.; Khanolkar, S.; Rodrigues, J.; Shilker, E.; Borkar, P.; Braz, E. Development of a Lifeguard Assist Drone for Coastal Search and Rescue. In Proceedings of the Global Oceans 2020: Singapore—U.S. Gulf Coast, Biloxi, MS, USA, 5–30 October 2020; pp. 1–10.
10. Yavuz, D.; Akbiyik, H.; Bostanci, E. Intelligent Drone Navigation for Search and Rescue Operations. In Proceedings of the 2016 24th Signal Processing and Communication Application Conference (SIU), Zonguldak, Turkey, 16–19 May 2016; pp. 565–568.
11. Sambolek, S.; Ivasic-Kos, M. Person Detection in Drone Imagery. In Proceedings of the 2020 5th International Conference on Smart and Sustainable Technologies (SpliTech), Split, Croatia, 23–26 September 2020; pp. 1–6.
12. Mazzia, V.; Comba, L.; Khaliq, A.; Chiaberge, M.; Gay, P. UAV and Machine Learning-Based Refinement of a Satellite-Driven Vegetation Index for Precision Agriculture. *Sensors* **2020**, *20*, 2530. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Guo, Y.; Guo, J.; Liu, C.; Xiong, H.; Chai, L.; He, D. Precision Landing Test and Simulation of the Agricultural UAV on Apron. *Sensors* **2020**, *20*, 3369. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Thongnim, P.; Yuvanatemiya, V.; Charoenwanit, E.; Srinil, P. Design and Testing of Spraying Drones on Durian Farms. In Proceedings of the 2023 International Technical Conference on Circuits/Systems, Computers, and Communications (ITC-CSCC), Jeju, Republic of Korea, 25–28 June 2023; pp. 1–6.
15. Junying, S. Modeling Design of Intelligent Agricultural Plant Protection UAV. In Proceedings of the 2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballar, India, 29–30 April 2023; pp. 1–7.
16. Al-Shareeda, M.A.; Manickam, S.; Saare, M.A. Intelligent Drone-Based IoT Technology for Smart Agriculture System. In Proceedings of the 2022 International Conference on Data Science and Intelligent Computing (ICDSIC), Karbala, Iraq, 1–2 November 2022; pp. 41–45.
17. Raj, K.M.; Balaji, N.; Vairavel, K.S.; Sharmila, P.; Azhagumurugan, R.; Jayakushal, S. IoT-Based Agricultural Drones for Pest Control. In Proceedings of the 2022 1st International Conference on Computational Science and Technology (ICCST), Chennai, India, 9–10 November 2022; pp. 134–137.
18. Zhang, S.; Wu, D.; Jiang, L.; Jin, X.; Cen, S. Research on UAV Access Deployment Algorithm Based on Improved Virtual Force Model. *KSII Trans. Internet Inf. Syst.* **2022**, *16*, 2606–2626.
19. Cai, H.; Wu, Z.; Chen, M. Design of STM32-Based Quadrotor UAV Control System. *KSII Trans. Internet Inf. Syst.* **2023**, *17*, 353–368.
20. Alvarado, E. *Drone Market Analysis 2022–2030*; Drone Industry Insights UG: Hamburg, Germany, 2022.
21. Product Safety Team. *Drone Safety Status Survey Results*; Korean Consumer Agency: Eumseong-gun, Republic of Korea, 2017.
22. Liang, J.; Wang, L.; Wu, J.; Liu, Z.; Yu, G. Prediction of Spindle Rotation Error through Vibration Signal based on Bi-LSTM Classification Network. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1043*, 042033. [\[CrossRef\]](#)
23. Han, T.; Pang, J.; Tan, A.C. Remaining Useful Life Prediction of Bearing Based on Stacked Autoencoder and Recurrent Neural Network. *J. Manuf. Syst.* **2021**, *61*, 576–591. [\[CrossRef\]](#)
24. Adlen, K.; Ridha, K. Recurrent Neural Network Optimization for Wind Turbine Condition Prognosis. *Diagnostyka* **2022**, *23*, 1–11. [\[CrossRef\]](#)
25. Li, X.; Li, J.; Qu, Y.; He, D. Gear Pitting Fault Diagnosis Using Integrated CNN and GRU Network with Both Vibration and Acoustic Emission Signals. *Appl. Sci.* **2019**, *9*, 768. [\[CrossRef\]](#)
26. Zhang, X.; Zhao, Z.; Wan, Z.; Wang, X. Fault Detection and Identification Method for Quadcopter Based on Airframe Vibration Signals. *Sensors* **2021**, *21*, 581. [\[CrossRef\]](#) [\[PubMed\]](#)
27. Ruslan, M.M.; Hassan, M.F. Unbalance Failure Recognition Using Recurrent Neural Network. *Int. J. Automot. Mech. Eng.* **2022**, *19*, 9668–9680. [\[CrossRef\]](#)
28. Xiao, D.; Huang, Y.; Zhang, X.; Shi, H.; Liu, C.; Li, Y. Fault Diagnosis of Asynchronous Motors Based on LSTM Neural Network. In Proceedings of the 2018 Prognostics and System Health Management Conference, Chongqing, China, 26–28 October 2018; pp. 540–545.
29. Huang, Y.; Chen, C.-H.; Huang, C.-J. Motor Fault Detection and Feature Extraction Using RNN-Based Variational Autoencoder. *IEEE Access* **2019**, *7*, 139086–139096. [\[CrossRef\]](#)
30. Wang, Z.; Qian, H.; Zhang, D.; Wei, Y. Prediction Model Design for Vibration Severity of Rotating Machine Based on Sequence-to-Sequence Neural Network. *Math. Probl. Eng.* **2019**, *2019*, 4670982. [\[CrossRef\]](#)

31. Wang, R.; Feng, Z.; Huang, S.; Fang, X.; Wang, J. Research on Voltage Waveform Fault Detection of Miniature Vibration Motor Based on Improved WP-LSTM. *Micromachines* **2020**, *11*, 753. [[CrossRef](#)] [[PubMed](#)]
32. ElSaid, A.; Wild, B.; Higgins, J.; Desell, T. Using LSTM Recurrent Neural Networks to Predict Excess Vibration Events in Aircraft Engines. In Proceedings of the 2016 IEEE 12th International Conference on e-Science (e-Science), Baltimore, MD, USA, 23–27 October 2016; pp. 260–269.
33. Zhu, J.; Ma, H.; Ji, L.; Zhuang, J.; Wang, J.; Liu, B. Vibration Trend Prediction of Pumped Storage Units based on VMD and GRU. In Proceedings of the 2020 5th International Conference on Mechanical, Control and Computer Engineering 2020, Harbin, China, 25–27 December 2020; pp. 180–183.
34. Yang, Z.; Liu, L.; Li, N.; Tian, J. Time Series Forecasting of Motor Bearing Vibration Based on Informer. *Sensors* **2022**, *22*, 5858. [[CrossRef](#)] [[PubMed](#)]
35. Hong, J.-K. Vibration Prediction of Flying IoT Based on LSTM and GRU. *Electronics* **2022**, *11*, 1052. [[CrossRef](#)]
36. Lee, J.-H.; Hong, J.-K. Comparative Performance Analysis of Vibration Prediction Using RNN Techniques. *Electronics* **2022**, *11*, 3619. [[CrossRef](#)]
37. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
38. Chung, J.; Gülçehre, Ç.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:1412.3555.
39. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015; pp. 1–15. [[CrossRef](#)]
40. Schuster, M.; Paliwal, K. Bidirectional Recurrent Neural Networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [[CrossRef](#)]
41. Wu, Y.; Johnson, J. Rethinking “Batch” in BatchNorm. *arXiv* **2021**, arXiv:2105.07576.
42. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference for Learning Representations, San Diego, CA, USA, 7–9 May 2015; pp. 1–15. Available online: <https://arxiv.org/abs/1412.6980> (accessed on 22 November 2023).

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