

Article

# Fast Measurement of Brillouin Frequency Shift in Optical Fiber Based on a Novel Feedforward Neural Network

Fen Xiao, Mingxing Lv  and Xinwan Li \* 

State Key Laboratory of Advanced Optical Communication Systems and Networks, Shanghai Jiao Tong University, Shanghai 200240, China; fxiao\_ch@sjtu.edu.cn (F.X.); l120034910186@sjtu.edu.cn (M.L.)

\* Correspondence: lixinwan@sjtu.edu.cn

**Abstract:** Brillouin scattering-based distributed optical fiber sensors have been successfully employed in various applications in recent decades, because of benefits such as small size, light weight, electromagnetic immunity, and continuous monitoring of temperature and strain. However, the data processing requirements for the Brillouin Gain Spectrum (BGS) restrict further improvement of monitoring performance and limit the application of real-time measurements. Studies using Feedforward Neural Network (FNN) to measure Brillouin Frequency Shift (BFS) have been performed in recent years to validate the possibility of improving measurement performance. In this work, a novel FNN that is 3 times faster than previous FNNs is proposed to improve BFS measurement performance. More specifically, after the original Brillouin Gain Spectrum (BGS) is preprocessed by Principal Component Analysis (PCA), the data are fed into the Feedforward Neural Network (FNN) to predict BFS.

**Keywords:** neural networks; Principal Components Analysis; distributed fiber sensing



**Citation:** Xiao, F.; Lv, M.; Li, X. Fast Measurement of Brillouin Frequency Shift in Optical Fiber Based on a Novel Feedforward Neural Network. *Photonics* **2021**, *8*, 474. <https://doi.org/10.3390/photonics8110474>

Received: 31 August 2021  
Accepted: 19 October 2021  
Published: 25 October 2021

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



**Copyright:** © 2021 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

Smart manufacturing based on the Internet of Things (IoT) is growing rapidly, driving the rapid development of IoT technology [1,2]. IoT is a network system that connects the Internet, global positioning systems, and various sensors, and is widely used in a variety of industries [3]. The key technology of IoT is sensing technology, in which distributed fiber optic sensors offer the advantages of small sensing size, light weight, electromagnetic immunity and continuous monitoring for temperature and strain. In the Brillouin optical time-domain analyzer (BOTDA), which is a typical system of distributed fiber optic sensors, the pump wave is pulsed into fiber, transferring energy into the probe wave. Scanning the required range of pump–probe frequency difference is used to construct BGS. BFS, the peak of BGS, is a function of local temperature and strain in the fiber [4,5]. Hence, one of the key technologies for the sensors is technology of estimating BFS from BGS, which can be realized using the traditional method, Lorentzian Curve Fitting (LCF). LCF is very time consuming and constitutes an obstacle to real-time measurement. Many techniques focused on hardware have been proposed to solve this problem, such as the use of balanced detection instead of increasing the number of measurements to eliminate polarization, and the use of a digital signal generator to switch quickly among scanning frequencies [6,7]. Additionally, sweep-free BOTDA has been proposed in order to boost the measurement speed, at the expense of complicated hardware and high cost [8–11]. Hence, some software approaches have been developed in recent years, wherein the training of FNN using BGSs with different full-width at half maximum (FWHM) was reported in 2019 [12,13], and the use of Principal Component Analysis (PCA) based on pattern recognition to avoid the measurement of BFS and directly measure the sensing information was reported in 2017 [14]. More specifically, the study conducted in [12] showed that FNN using ideal BGSs as the training set and temperature as the output of the network has higher accuracy and is faster than LCF. However, the FNN needs to be retrained when the frequency

scanning step of the inputted spectra changes, which is a time-consuming task. The use of noisy BGSs for FNN training has been reported, and the training method of FNN has been well described [13]. The study conducted in [14] constructed a reference database comprising BGS compressed by PCA and temperature, and the predicted temperature was determined by finding the best match in the reference database. The measurement range for this algorithm is correlated with the size of the reference database, meaning that the required storage of the algorithm increases as the measurement range increases. This is a space-consuming task. Additionally, machine learning algorithms have been investigated to study the feasibility of using them for temperature measurement, skipping the BFS measurement process [15]. The Random Forest algorithm is usually outperformed by the FNN algorithm; when attempting to improve the efficiency of the Gradient Boosted Trees algorithm, its processing time increases; the Support Vector Machine algorithm requires complex classifiers when the number of classes is greater than two, leading to increased processing time.

In this work, a novel FNN is proposed to improve BFS measurement and achieve a faster measurement speed. Simulation results show that the proposed FNN has the ability to measure BFS from BGS with a shorter measurement time and adaptability to different frequency scanning steps. In this proof-of-concept experiment, the experimental results show that the novel FNN is able to accurately measure BFS from experimental BGS.

## 2. Theory

### 2.1. PCA

PCA is a powerful tool for data analysis, enabling us to identify patterns in data, especially the patterns in data of high dimension, and to express the data in such a way as to reduce their dimension [16–18]. Consequently, the other advantage of PCA is to compress the data without losing much information. In the mathematical computation of PCA,  $M$  samples are needed, where the  $i^{\text{th}}$  sample is represented as  $s_i$ , with a length of  $N$ . These samples are represented as

$$S = [s_1^T, s_2^T, \dots, s_M^T]^T, \quad (1)$$

where  $T$  represents the transposing matrix. For PCA to work properly, the mean matrix  $\bar{S}$  is obtained using the following equations:

$$\bar{s} = \frac{\sum_{i=1}^M s_i}{M}, \quad (2)$$

$$\bar{S} = [\bar{s}_1^T, \bar{s}_2^T, \dots, \bar{s}_M^T]^T, \quad (3)$$

$$\bar{s}_i = s_i - \bar{s}. \quad (4)$$

The next step is to calculate the covariance matrix  $Cov$  using the following equation:

$$Cov = \frac{1}{M} \sum_{i=1}^M \bar{s}_i^T \bar{s}_i = \bar{S}^T \bar{S}. \quad (5)$$

Then, the eigenvectors  $\mu_i$  and eigenvalues  $\lambda_i$  of the covariance matrix  $Cov$  are calculated, where  $i = 1, 2, \dots, N$ . These eigenvalues are different, and the eigenvectors are ordered by the corresponding eigenvalues from highest to lowest. Then, the first  $p$  eigenvectors are selected, and the others are ignored, whereby  $p$  is decided by

$$\frac{\sum_{i=1}^p \lambda_i}{\sum_{i=1}^N \lambda_i} > \delta. \quad (6)$$

Most applications of PCA usually employ  $\delta$  of more than 0.9. The first  $p$  eigenvectors sorted from highest to lowest can be expressed together as a matrix  $PC = [\mu_1, \mu_2, \dots, \mu_p]$ . Therefore, the  $i^{\text{th}}$  sample  $s_i$  is transformed to

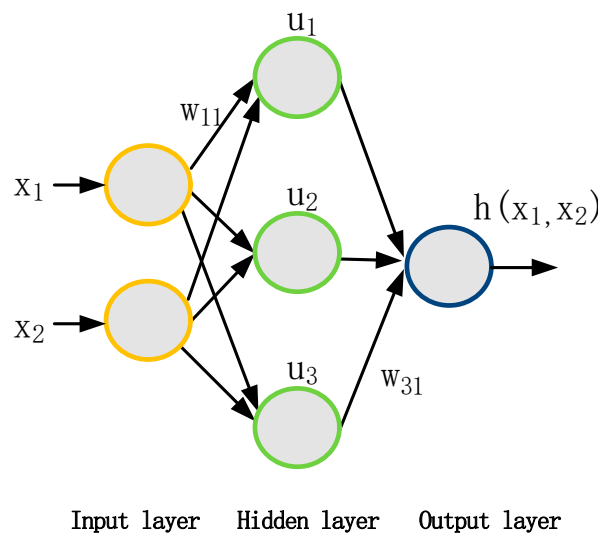
$$\hat{s}_i = \bar{s}_i \times PC = (s_i - \bar{s}) \times PC, \tag{7}$$

where  $\hat{s}_i$  is a row vector, a length of  $p$ , and  $i = 1, 2, \dots, M$ . Because  $p < N$ , all samples can be reduced from high dimension to a comparatively small dimension.

### 2.2. FNN

FNN is the simplest type of Artificial Neural Network, and is shown in Figure 1 [19]. All units of the previous layer are connected to all units of the next layer; however, units in the same layer cannot be connected to each other. In this network, information flows from the input layer, through the hidden layer, to the output layer, in only one direction. To understand FNN, taking FNN of 3 layers as example, some common markups as shown in Figure 1 are defined:

- The input vector is  $[x_1, x_2]$ ;
- The input and output of unit  $u_i$  in the hidden layer is  $m_i$  and  $n_i$ , respectively;
- The weight from  $j^{\text{th}}$  unit in the previous layer to  $i^{\text{th}}$  in the next layer is  $w_{j,i}$ ;
- The output value is  $h(x_1, x_2)$ .



**Figure 1.** Typical FNN with 3 layers, which are the input layer, the hidden layer and the output layer, respectively. There are 3 units in the hidden layer.

When information moves from the input layer to the output layer, the input of the unit  $u_i$  in the hidden layer is

$$m_i = \sum_{j=1}^2 x_j w_{j,i} \tag{8}$$

and the output is

$$n_i = f(m_i), \tag{9}$$

where  $f(\cdot)$  is the activation function and  $i = 1, 2, 3$ . Consequently, the output of the unit in the output layer is

$$h(x_1, x_2) = \sum_{j=1}^3 n_j w_{j,1} = \sum_{j=1}^3 f(m_j) w_{j,1}. \tag{10}$$

The expected result of the input  $[x_1, x_2]$  is  $y$ , and the error between the expected result and the output is

$$e = y - h(x_1, x_2). \tag{11}$$

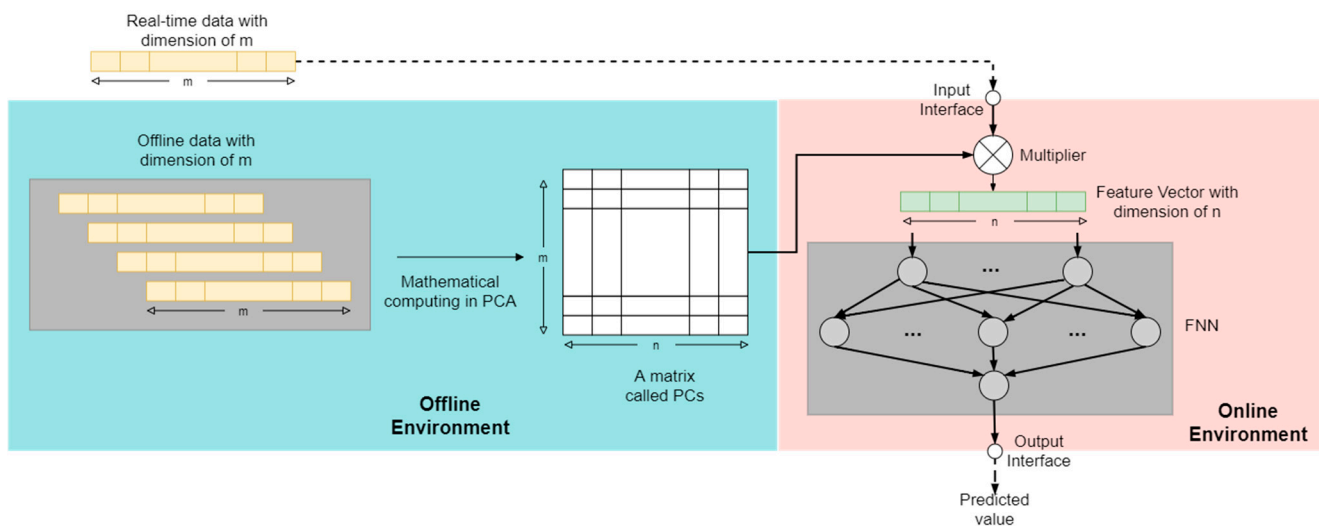
To use the algorithm of backpropagation (BP), the cost function

$$J(w) = e^2 = (y - h(x_1, x_2))^2. \tag{12}$$

can be minimized by adjusting the weights  $w_{j,i}$ . FNN uses two datasets: the training set and the testing set. The training set is used to adjust the weights of FNN and the testing set used to determine their effectivity.

### 2.3. FNN with PCA

Making use of the advantages of PCA and FNN, a novel FNN is proposed in this work. The theoretical framework of this technique is shown in Figure 2. In an offline environment, a matrix is obtained through the mathematical computation of PCA and is not changed when real-time data are fed into the system. In an online environment, real-time data are fed into the system through the Input Interface, and the Output Interface outputs the predicted value.



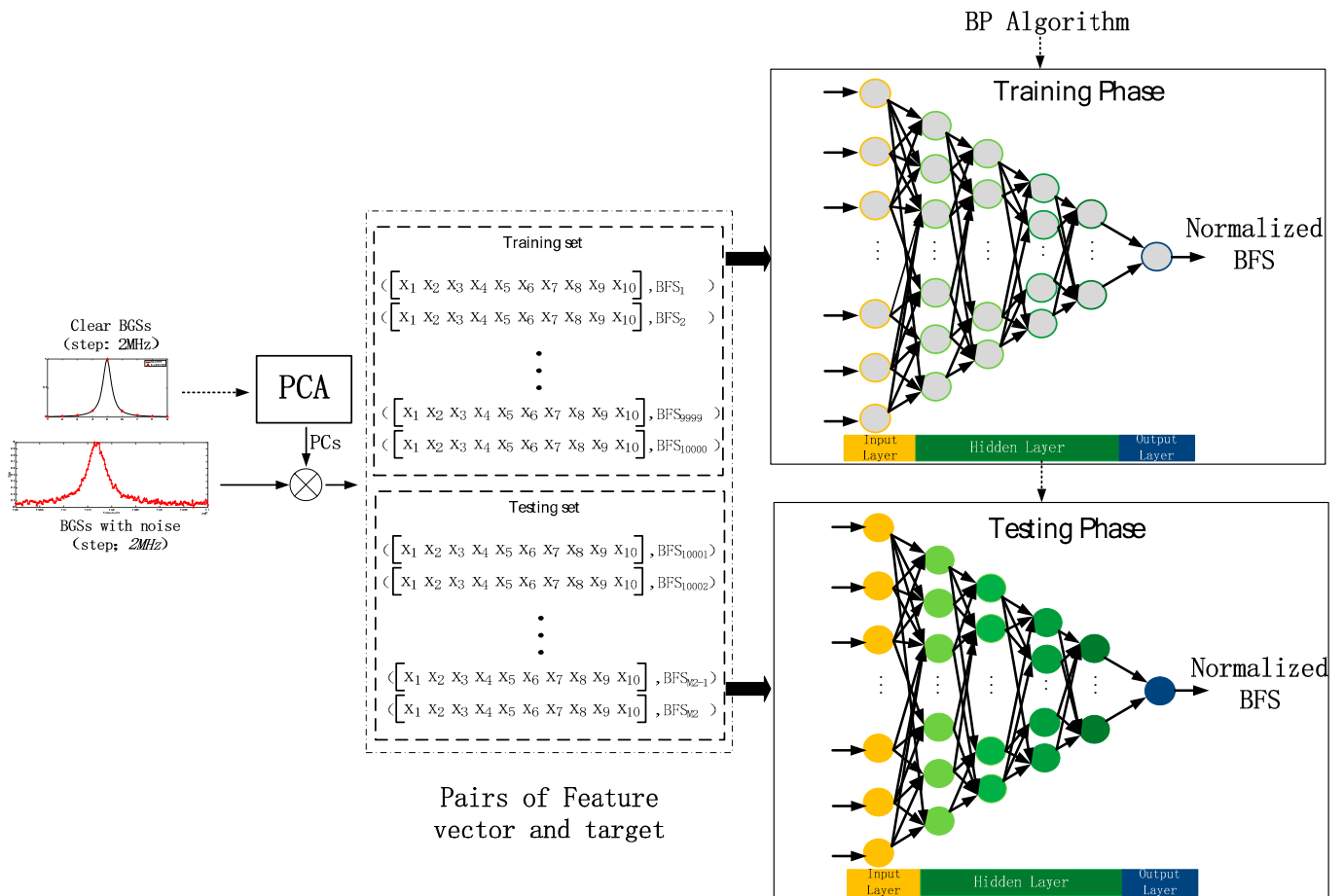
**Figure 2.** Theoretical framework of FNN with PCA. PCs is a matrix, which is the result of the mathematical computation and maps real-time data to the corresponding Feature Vector.

In an offline environment, a set of data are needed having a length of  $m$ . The aim of PCA is to learn a mapping relationship from the group of data. A matrix, referred to as PCs, represents this relationship. PCs has dimensions of  $m \times n$ , and  $n$  determines the number of units in the input layer of FNN. Additionally, note that there is a one-to-one correspondence between PCs and the length of data, and  $n$  is much lower than  $m$ .

In an online environment, real-time data can be fed into the system and a predicted result should be measured through FNN. In this process, real-time data are transformed into low-dimensional data, referred to as a Feature Vector. In fact, FNN receives the transformed data and not the raw data. This is unique and novel. Consequently, the novel FNN can achieve fast measurement of accurate results without sacrificing accuracy.

### 3. Proposed Method

To improve the measurement speed of BFS, the novel FNN was applied to estimate BFS, as shown in Figure 3, wherein the layout of the FNN was set as 10-50-15-1, where the numbers represent the number of neurons in the input layer, the two hidden layers and the output layer, respectively.



**Figure 3.** Detailed architecture for the measurement of BFS using the proposed method. The matrix, PCs, is obtained from a group of clear BGS, and another group of BGS, as input data, is divided into two parts: a training set to train the FNN, and a testing set to determine the generalization ability of the FNN. The layout of the FNN is set to 10-50-15-1. The output of the FNN is normalized BFS.

In the training phase, the training set of BGS is multiplied by the selected PCs to obtain 10-dimensional Feature Vectors to train the FNN. The Feature Vectors and the corresponding BFSs are used to construct (feature vector, BFS) pairs, wherein feature vector is the input of the FNN and BFS is the peak of the BGS. Then, the weights of FNN are randomly initialized and updated using the training set until Equation (12) is minimized by the algorithm of BP. In addition, theoretical BGS is used to train the FNN, not experimental BGS. An example of theoretical BGS is shown in Figure 4, presenting a Lorentzian spectral profile:

$$g(\nu) = \frac{g_0}{4\left(\frac{\nu - \nu_B}{\Delta\nu_B}\right)^2 + 1}, \quad (13)$$

where  $g_0$  is the max of BGS,  $\Delta\nu_B$  is the FWHM and BGS peak at the frequency of  $\nu_B$ . According to Equation (13), simulated BGS can be produced. More specifically, a group of discrete data points, the red triangle in Figure 4, represent BGS, and the sampling interval is the frequency scanning step. Obviously, the frequency scanning step affects the length of BGS, i.e.,  $m$  in Figure 2. This means that different frequency scanning steps require different PCs. Noisy BGS is used to train the FNN. The starting frequency of BGS is 10.6 GHz, the ending frequency is 11.00 GHz, and frequency scanning step is 2 MHz; more details about noisy BGS are shown in Table 1. Meanwhile, BGS without noise is used to obtain PCs, which is almost same as noisy BGS, but has no noise; more details shown in Table 2.

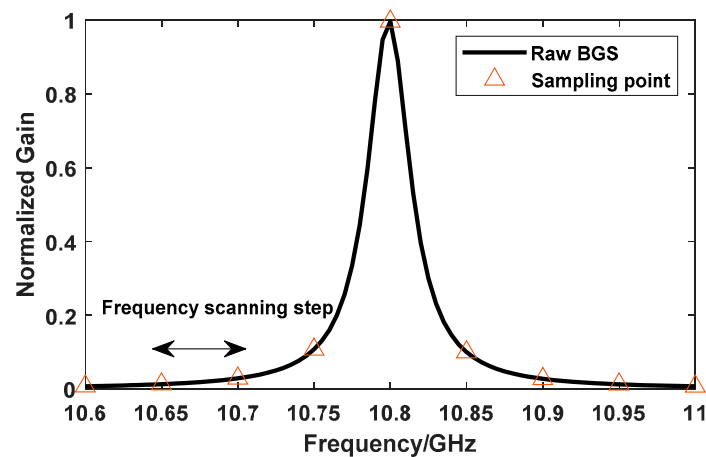


Figure 4. A BGS sample is obtained according to Equation (13).

Table 1. Details of the noisy BGSs used to train and test the proposed method. The starting frequency, the ending frequency, and the frequency scanning step of BGS were 10.6 GHz, 11.00 GHz and 2 MHz, respectively.

| Parameter           | Range           | Interval | Number of Repetitions |
|---------------------|-----------------|----------|-----------------------|
| BFS                 | 10.62–10.98 GHz | 1 MHz    | 1                     |
| FWHM                | 10–60 MHz       | 1 MHz    | 1                     |
| SNR                 | 10–40 dB        | 1 dB     | 3                     |
| BFS <sub>norm</sub> | 5–95%           | 1/400    | 51                    |

Table 2. Details of the clear BGSs used to obtain PCs. The starting frequency, the ending frequency, and the frequency scanning step of BGS were 10.6 GHz, 11.00 GHz and 2 MHz, respectively.

| Parameter           | Range           | Interval | Number of Repetitions |
|---------------------|-----------------|----------|-----------------------|
| BFS                 | 10.62–10.98 GHz | 1 MHz    | 1                     |
| FWHM                | 10–60 MHz       | 1 MHz    | 1                     |
| SNR                 | -               | -        | -                     |
| BFS <sub>norm</sub> | 5–95%           | 1/400    | 51                    |

In the testing phase, the testing environment is simulated using the testing set. The Feature Vector is fed into the FNN with optimized weights and measured value of BFS is outputted. The performance of the FNN is estimated by calculating the Mean Absolute Error

$$MAE = \frac{\sum_{i=1}^n |x^i - x_{target}^i|}{n}, \tag{14}$$

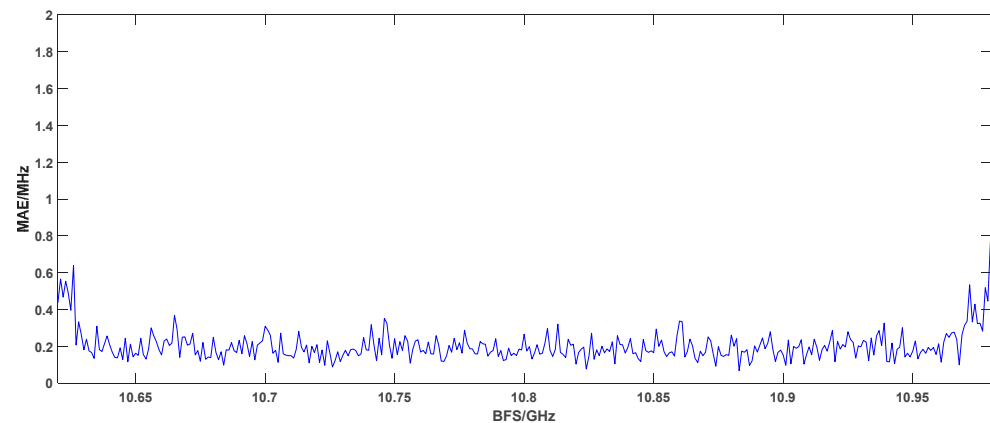
where  $x^i$  is the  $i^{th}$  measured BFS and  $x_{target}^i$  is the corresponding target value.

## 4. Results

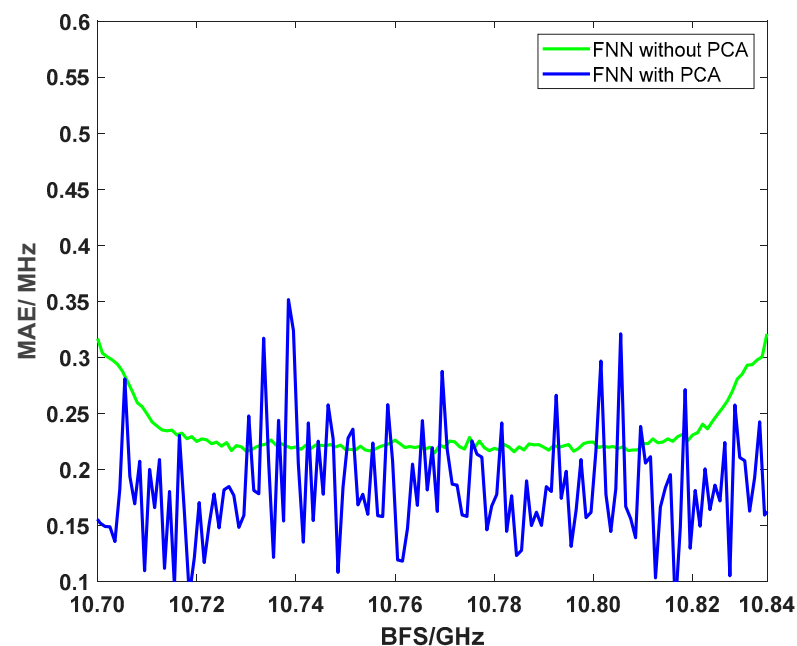
### 4.1. Simulation Results

According to the proposed method, a well-trained FNN was constructed. In this section, simulated BGSs that were not used in the training phase are fed into the FNN. The MAEs of the group of simulated BGSs at different targeted BFS are shown in Figure 5. It can be seen that the MAEs were less than 0.4 MHz between 10.65 GHz and 10.95 GHz. When BFSs were less than 10.65 GHz or more than 10.95 GHz, MAEs became larger, but were not more than 0.8 MHz. It can be concluded that the proposed FNN is capable of measuring BFS from BGS. In Figure 6, a comparison is presented between the proposed FNN, marked as FNN with PCA, and other kinds of FNN [13], marked as FNN without

PCA. The details of the comparison are shown in Table 3. The efficiencies of the two methods were taken into account in the comparison. The processing time is the total time required, from feeding 1,712,223 samples into the neural network until all measurement results had been outputted. It can be seen that the efficiency of the FNN with PCA is 3 times that of the second FNN. Additionally, FNN with PCA has almost the same accuracy as FNN without PCA.



**Figure 5.** MAEs of the proposed method at different BFS. MAEs are less than 0.4 MHz between 10.65 GHz and 10.95 GHz.



**Figure 6.** Comparison between FNN with PCA and FNN without PCA. FNN with PCA is the proposed method and other FNN is taken from [13].

FNN with PCA is not only efficient, but it can also be adapted to different frequency scanning steps. For the same frequency scanning range, different frequency scanning steps means different numbers of sampling points. In other words, when the raw sampling data of BGS is the input data of the FNN, the number of inputs is equal to the number of sampling points. In this way, BGS with different frequency scanning steps cannot use the same FNN, necessitating the retraining of the FNNs. However, the FNN with PCA solves this problem, since the length of the Feature Vector does not change as the frequency scanning step changes. Consequently, the FNN with PCA has the ability to use BGS with different frequency scanning steps as input. This ability is referred to as adaptability to different frequency scanning steps.

**Table 3.** Details of the comparison between FNN with PCA and FNN without PCA.

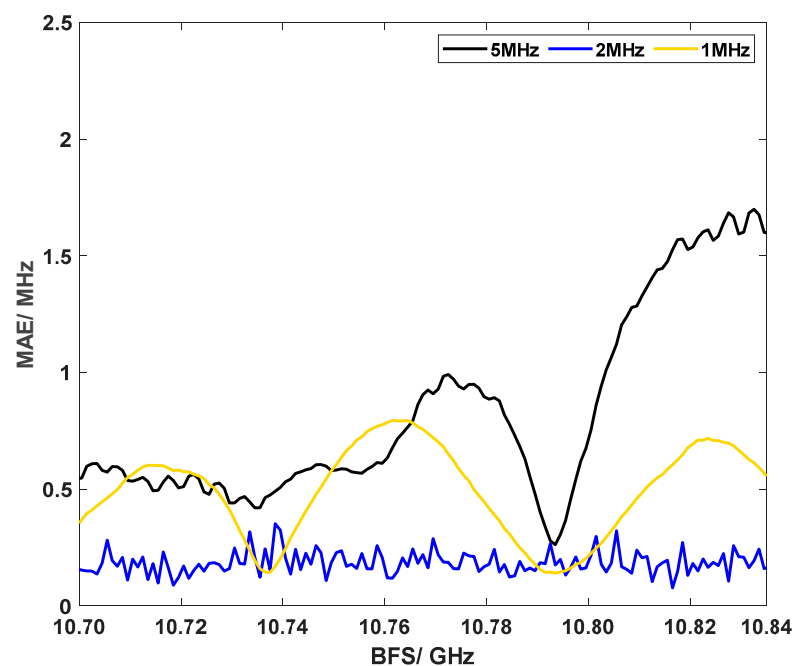
| FNN Type        | MAE        | Processing Time <sup>1</sup> |
|-----------------|------------|------------------------------|
| FNN with PCA    | 0.2027 MHz | 1.4224 s                     |
| FNN without PCA | 0.2329 MHz | 4.5273 s                     |
| LCF             | 0.1816 MHz | 886 s                        |

<sup>1</sup> Processing time of the two FNN is the total time from the 1,712,223 samples being fed into the neural network until the moment when all measurement results are outputted. The processing time of LCF is the total processing time required for 1,712,223 samples.

Three frequency scanning steps, 1 MHz, 2 MHz and 5 MHz, are taken into consideration. The MAEs of the three steps are shown in Figure 7. It can be clearly seen that 5 MHz is more unstable than both 2 MHz and 1 MHz. As shown in Table 4, the MAE for 5 MHz is 1.1968 MHz, 2 MHz is 0.2027 MHz and 1 MHz is 0.4888 MHz. Hence, it can be concluded that the frequency scanning step of the inputted BGS can be either 1 MHz or 2 MHz. The MAE of 2 MHz is lower than that of 1 MHz, because the training set consists of 2 MHz BGSs. The FNN only learns the pattern between BFS and the Feature Vector of 2 MHz from the training set, and the 2 MHz pattern is different from the 1 MHz pattern. However, the patterns between BFS for the three steps, namely 1 MHz, 2 MHz and 5 MHz, are similar. Similarity is calculated using the following equation:

$$S(i, j) = \frac{\sum_{k=1}^n (x_k^i - x_k^j)}{n}, \tag{15}$$

where  $(x_1^i \ x_2^i \ \dots \ x_n^i)$  is the Feature Vector of the  $i^{\text{th}}$  step and  $(x_1^j \ x_2^j \ \dots \ x_n^j)$  is the Feature Vector of the  $j^{\text{th}}$  step; the similarity between them can be calculated. Because the FNN with PCA is trained using a 2 MHz step, the similarity between 2 MHz and the other two steps should be examined. As shown in Table 5, the similarity between 2 MHz and 1 MHz is the lowest. Comparing the similarity between the Feature Vectors with different steps, it can be seen that the Feature Vector with 2 MHz is more similar to the Feature Vector with 1 MHz than to the Feature Vector with 5 MHz, as can be seen in Figure 8. Consequently, the MAE of 1 MHz is smaller than the MAE of 5 MHz.



**Figure 7.** Adaptability of different steps for the FNN with PCA.

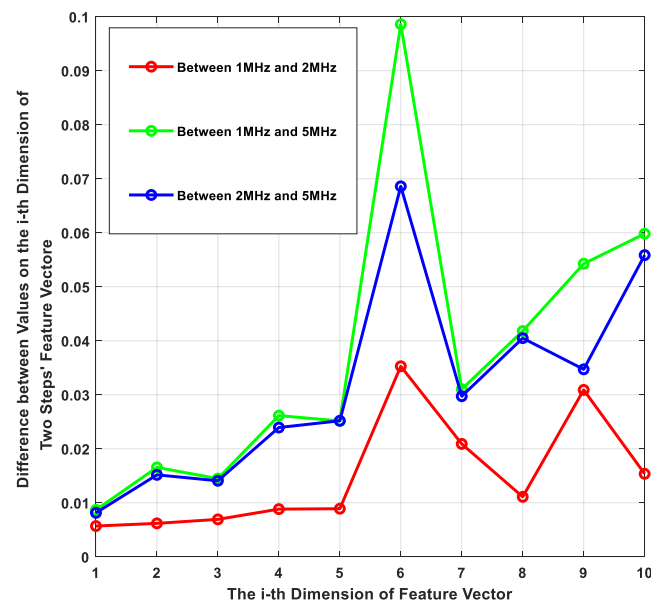


**Table 4.** MAEs of different frequency scanning steps.

| Frequency Scanning Step | MAE        |
|-------------------------|------------|
| 5 MHz                   | 1.1968 MHz |
| 2 MHz                   | 0.2027 MHz |
| 1 MHz                   | 0.4888 MHz |

**Table 5.** There is similarity between different frequency scanning steps.  $S(i, j)$  is calculated using Equation (15), and there is similarity between the  $i^{\text{th}}$  frequency scanning step and the  $j^{\text{th}}$  frequency scanning step.

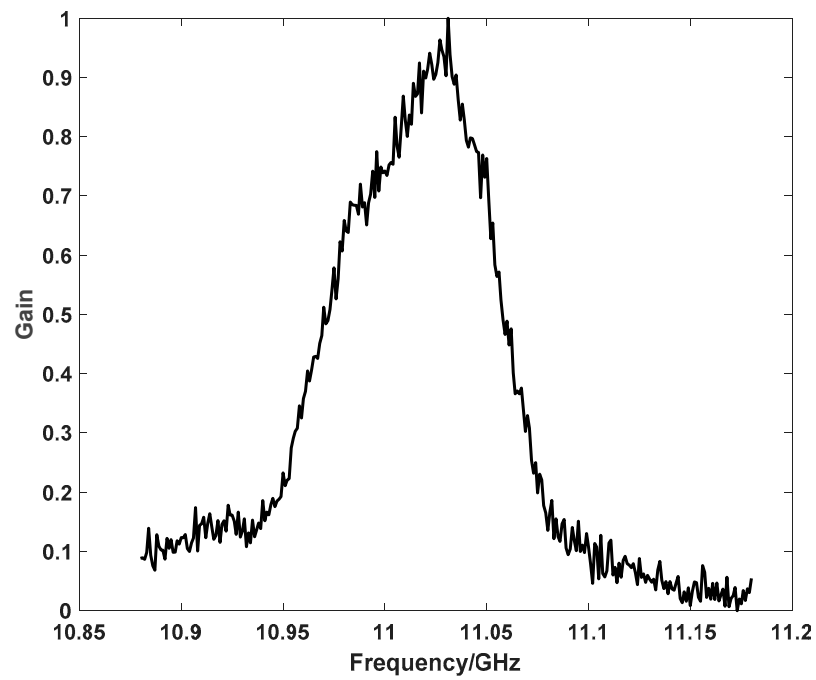
| The $i^{\text{th}}$ Frequency Scanning Step | The $j^{\text{th}}$ Frequency Scanning Step | $S(i, j)$ |
|---|---|-----------|
| 1 MHz                                       | 2 MHz                                       | 0.0150    |
| 1 MHz                                       | 5 MHz                                       | 0.0377    |
| 2 MHz                                       | 5 MHz                                       | 0.0316    |



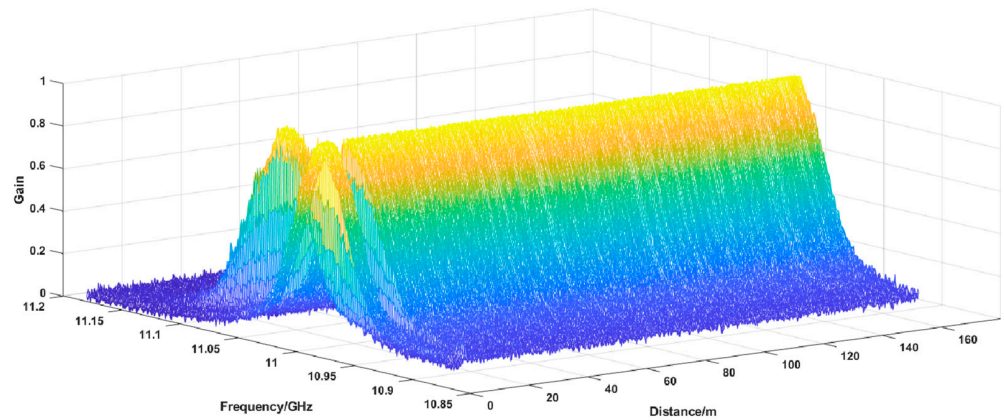
**Figure 8.** Results of vector subtraction between 10-dimensional Feature Vectors of different steps are visually shown. The smaller the magnitude of vector subtraction, the more similar the two feature vectors of different steps.

4.2. Experimental Results

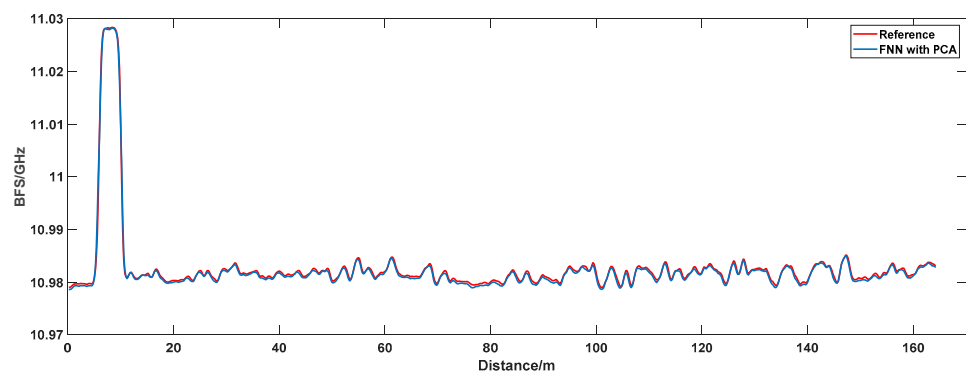
Experimental BGSs were obtained from [20], as shown in Figure 9. The starting frequency is 10.88 GHz, the ending frequency is 11.19 GHz, and the step is 1 MHz. In the 80-km BOTDA system, the 80-km sensing fiber is in room of 26 °C and the 5-m hotspot of 76 °C is placed at the beginning of the fiber. A three-dimensional BGS spectrum along the fiber in the BOTDA system is shown in Figure 10. Measurement of BFS is performed in an online environment. The experimental BGSs has step of 1 MHz, so PCs taken from clear BGSs of 1 MHz is used to compress experimental data and obtain the corresponding feature vector. Now we feed the experimental BGSs into the well-trained FNN to measure BFS. In this way, BFS can be estimated from BGS, as shown in Figure 11. The measurement error of the technique is shown in Figure 12. It is shown that the error is huge, when BFS has sudden change, because the shape of BGS is distorted in the measurement process of the BOTDA system, such as when the shape changes from one-peak to two-peak. In addition, MAE is used for the quantitative interpretation of BFS error value. The MAE excluding the data near temperature variations is 0.3289 MHz, as calculated using Equation (14), where the target value is obtained by LCF, indicating that FNN with PCA can accurately measure BFS from experimental BGS.



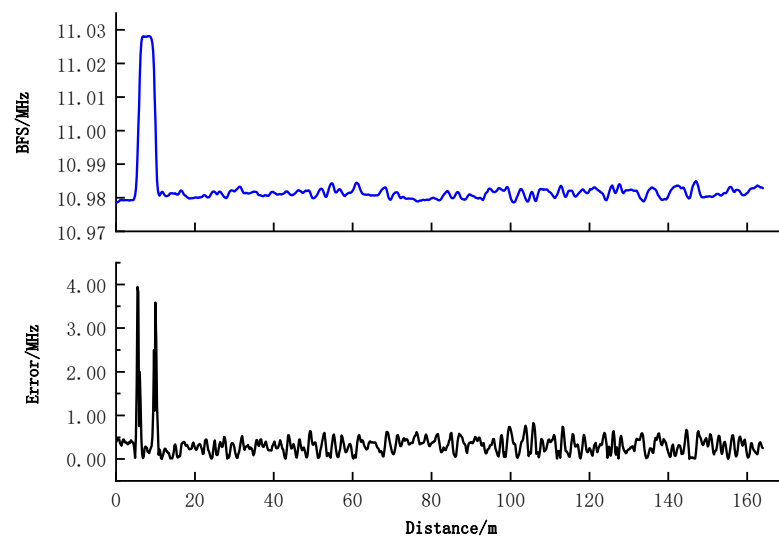
**Figure 9.** Experimental BGS obtained through BOTDA. The starting frequency of BGS is 10.88 GHz, the ending frequency is 11.19 GHz, and the step is 1 MHz.



**Figure 10.** Three-dimensional BGS spectrum along the fiber in the BOTDA system.



**Figure 11.** There are estimated BFSs at the beginning of the sensing fiber, where the 5-m hotspot of 76 °C is placed. The red line represents the results of LCF measuring BFS, as a reference for real BFS.



**Figure 12.** Errors between FNN with PCA and LCF along the fiber under test are shown, in the form of the black line. The blue line represents the BFS estimated by FNN with PCA along the fiber.

## 5. Conclusions

An accurate and ultra-fast method for estimating BFS from BGS is was and systematically compared with FNN with PCA and FNN without PCA. By comparison, our approach is 3 times faster than other FNN without sacrifice of accuracy. Meanwhile, experimental results show that the novel FNN can accurately measure BFS from experimental BGS. This will contribute to distributed optical fiber sensors, especially when the sensing distance is long. More importantly, measuring BFS using FNN with PCA is adaptable to different sensing applications, such as temperature and strain. Additionally, FNN based on PCA can be reused for different frequency scanning step, saving training cost and improving adaptability of the FNN.

**Author Contributions:** Conceptualization, F.X.; Data curation, F.X. and M.L.; Formal analysis, F.X. and M.L.; Funding acquisition, X.L.; Investigation, F.X.; Methodology, F.X.; Project administration, X.L.; Resources, F.X.; Software, F.X. and M.L.; Supervision, X.L.; Validation, F.X.; Visualization, F.X.; Writing—original draft, F.X.; Writing—review & editing, X.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by NSFC (Grant Number 12042507).

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

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Restrictions apply to the availability of these experimental data. Data was obtained from the research team of Jian Wu and are available at <https://data.mendeley.com/datasets/s7yx6xb7ys/1> with the permission of the research team of Jian Wu.

**Acknowledgments:** The authors thank the research team of Jian Wu for experimental data.

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

## References

1. Thoben, K.-D.; Wiesner, S.; Wuest, T. “Industrie 4.0” and Smart Manufacturing—A Review of Research Issues and Application Examples. *Int. J. Autom. Technol.* **2017**, *11*, 4–16. [CrossRef]
2. Wang, J.; Ma, Y.; Zhang, L.; Gao, R.X.; Wu, D. Deep learning for smart manufacturing: Methods and applications. *J. Manuf. Syst.* **2018**, *48*, 144–156. [CrossRef]
3. Bandyopadhyay, D.; Sen, J. Internet of Things: Applications and Challenges in Technology and Standardization. *Wirel. Pers. Commun.* **2011**, *58*, 49–69. [CrossRef]
4. Feng, C.; Schneider, T. Benefits of Spectral Property Engineering in Distributed Brillouin Fiber Sensing. *Sensors* **2021**, *21*, 1881. [CrossRef] [PubMed]

5. Horiguchi, T.; Tateda, M. Optical-fiber-attenuation investigation using stimulated Brillouin scattering between a pulse and a continuous wave. *Opt. Lett.* **1989**, *14*, 408–410. [[CrossRef](#)] [[PubMed](#)]
6. Lopez-Gil, A.; Dominguez-Lopez, A.; Martin-Lopez, S.; Gonzalez-Herraez, M. Simple Method for the Elimination of Polarization Noise in BOTDA Using Balanced Detection and Orthogonal Probe Sidebands. *J. Light. Technol.* **2014**, *33*, 2605–2610. [[CrossRef](#)]
7. Peled, Y.; Motil, A.; Tur, M. Fast Brillouin optical time domain analysis for dynamic sensing. *Opt. Express* **2012**, *20*, 8584–8591. [[CrossRef](#)] [[PubMed](#)]
8. Voskoboinik, A.; Willner, A.E.; Tur, M. Extending the dynamic range of sweep-free Brillouin optical time-domain analyzer. *J. Light. Technol.* **2015**, *33*, 1. [[CrossRef](#)]
9. Voskoboinik, A.; Wang, J.; Shamee, B.; Nuccio, S.R.; Zhang, L.; Chitgarha, M.; Willner, A.E.; Tur, M. SBS-Based Fiber Optical Sensing Using Frequency-Domain Simultaneous Tone Interrogation. *J. Light. Technol.* **2011**, *29*, 1729–1735. [[CrossRef](#)]
10. Jin, C.; Guo, N.; Feng, Y.; Wang, L.; Liang, W.; Li, J.; Li, Z.; Yu, C.; Lu, C. Scanning-free BOTDA based on ultra-fine digital optical frequency comb. *Opt. Express* **2015**, *23*, 5277–5284. [[CrossRef](#)] [[PubMed](#)]
11. Voskoboinik, A.; Yilmaz, O.F.; Willner, A.W.; Tur, M. Sweep-free distributed Brillouin time-domain analyzer (SF-BOTDA). *Opt. Express* **2011**, *19*, B842–B847. [[CrossRef](#)] [[PubMed](#)]
12. Azad, A.K.; Wang, L.; Guo, N.; Tam, H.Y.; Lu, C. Signal processing using artificial neural network for BOTDA sensor system. *Opt. Express* **2016**, *24*, 6769–6782. [[CrossRef](#)] [[PubMed](#)]
13. Liang, Y.; Jiang, J.; Chen, Y.; Zhu, R.; Lu, C.; Wang, Z. Optimized Feedforward Neural Network Training for Efficient Brillouin Frequency Shift Retrieval in Fiber. *IEEE Access* **2019**, *7*, 68034–68042. [[CrossRef](#)]
14. Azad, A.K.; Khan, F.N.; Al-Arashi, W.; Guo, N.; Lau, A.P.T.; Lu, C. Temperature extraction in Brillouin optical time-domain analysis sensors using principal component analysis based pattern recognition. *Opt. Express* **2017**, *25*, 16534–16549. [[CrossRef](#)] [[PubMed](#)]
15. Nordin, N.D.; Zan, M.S.D.; Abdullah, F. Comparative Analysis on the Deployment of Machine Learning Algorithms in the Distributed Brillouin Optical Time Domain Analysis (BOTDA) Fiber Sensor. *Photonics* **2020**, *7*, 79. [[CrossRef](#)]
16. Aguilera, A.M.; Gutiérrez, R.; Ocaña, F.A.; Valderrama, M.J. Computational approaches to estimation in the principal component analysis of a stochastic process. *Appl. Stoch. Model. Data Anal.* **1995**, *11*, 279–299. [[CrossRef](#)]
17. Jolliffe, I.T. *Principal Component Analysis*, 2nd ed.; Springer: New York, NY, USA, 2002; pp. 150–166. [[CrossRef](#)]
18. Alvin, C.; Rencher, W.F.C. *Methods of Multivariate Analysis*, 3rd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2021; pp. 169–257. [[CrossRef](#)]
19. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)] [[PubMed](#)]
20. Hong, X.; Wu, J.; Li, Y.; Wang, S.; Qiu, J.; Sun, X.; Zhang, X. Data for: A fast method for Brillouin frequency shift estimation. *Sens. Actuators A Phys.* **2018**, *284*, 6–11. [[CrossRef](#)]