Brillouin Frequency Shift Extraction Based on AdaBoost Algorithm

Huan Zheng *, Feng Xiao, Shijie Sun and Yali Qin *

Institute of Fiber-Optic Communication and Information Engineering, College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China; yhs95436007@163.com (F.X.); 2112003151@zjut.edu.cn (S.S.)
* Correspondence: zhenghuan@zjut.edu.cn (H.Z.), ylqin@zjut.edu.cn (Y.Q.)

Abstract: The Brillouin Optical Time-Domain Analyzer assisted by the AdaBoost Algorithm for Brillouin frequency shift (BFS) extraction is proposed and experimentally demonstrated. The Brillouin gain spectrum classification under different BFS is realized by iteratively updating the weak classifier in the form of a decision tree, forming several base classifiers and combining them into a strong classifier. Based on the pseudo-Voigt curve training set with noise, the performance of the AdaBoost Algorithm is studied, and the influence of different signal-to-noise ratio (SNR), frequency range, and frequency step is also studied. Results show that the performance of BFS extraction decreases with the decrease in SNR, the reduction in frequency range, and the increase in frequency step.

Keywords: distributed fiber sensing; Brillouin scattering; Brillouin frequency shift; AdaBoost algorithm

1. Introduction

The advantages of distributed optical fiber sensors include distributed sensing, high spatial resolution, large dynamic range, real-time monitoring, etc. The technique of distributed fiber sensing can be widely applied in various areas, such as industrial infrastructure health monitoring, long-haul vibration detection, and quick fault location [1–3]. The Brillouin Optical Time-Domain Analyzer (BOTDA), as one of many distributed optical fiber sensors, can be used for monitoring both temperature and strain in ultra-long sensing ranges [4]. In order to obtain temperature and strain information from BOTDA, Brillouin frequency shift (BFS) needs to be extracted from the measured Brillouin Gain Spectrum (BGS).

One method for BFS extraction is curve fitting [5–8]. The curves can be fitted by using Lorentzian curve, parabolic curve, pseudo-Voigt curve, etc. [5,6]. However, the curve fitting method is highly dependent on the initial conditions. As for the BGS with low signal-to-noise ratio (SNR), the initial parameters need to be adjusted carefully and often lead to fitting failures [7]. In addition, the curve fitting method requires a small frequency step, because a large frequency step could reduce the number of fitting points and affect the fitting performance [8].

In order to avoid the problem of initial parameter adjustment in the curve fitting method, the Cross-Correlation Method (XCM) is proposed to extract BFS [8,9]. XCM does not have the problem of initial parameter setting, but it requires data interpolation processing to up-sample the measured BGS, which could be time consuming. As a result, there is a trade-off between accuracy and processing speed [9].

The method of Artificial Neural Network (ANN) has also been introduced in the BOTDA system [10–12]. ANN is proved to be an effective method for BFS extraction; however, the number of intermediate layers and the number of neurons in layers need to be designed carefully [10]. Moreover, the training process is always complicated and time-consuming.
consumption. The method can be converted to the local optimal solution, which limits its applications [11,12].

In recent years, Support Vector Machine (SVM) has also been proposed to extract temperature from the whole BGS [13–15]. However, because SVM is essentially a binary classifier, it is often necessary to construct hundreds of support vectors to extract temperature information. The error of support vectors could also affect the accuracy of extraction.

Here, we propose a novel method for BFS extraction. The method is based on the AdaBoost Algorithm. The basic idea of AdaBoost is to train different weak classifiers with the same training sets and combine them to build a strong classifier. The advantages of the AdaBoost Algorithm include freedom of extraction failure and simplicity of parameter adjustments [16,17]. Generally, the decision tree or neural network can be chosen as the weak classifier [18,19]. In this study, the BFS extraction is treated as a supervised classification problem. The algorithm of the AdaBoost method is applied to extract BFS from BGS.

The article is organized as follows: In Section II the principle of the AdaBoost Algorithm is introduced, and the AdaBoost Algorithm for extracting BFS is illustrated. In Section III, the performance of BFS extraction by AdaBoost is numerically studied under different SNR, frequency range, and frequency step. In Section IV, the BFS extraction by the AdaBoost Algorithm is experimentally studied. The result of the experiment is compared with that of the simulation. Finally, the study is briefly concluded in Section V.

2. Theoretical Model

2.1. Principle of AdaBoost

AdaBoost is a supervised learning algorithm for solving binary classification problems. The principle of AdaBoost is shown in the details below.

The decision tree is chosen to be the weak classifier. The algorithm of Classification And Regression Tree (CART) is introduced to train the decision tree. For instance, \( X_{n\times m} \) stands for the properties of the training set, which can be expressed as follows [20]:

\[
X_{n\times m} = \begin{bmatrix}
X_{1,1} & X_{1,2} & \cdots & X_{1,m} \\
X_{2,1} & X_{2,2} & \cdots & X_{2,m} \\
\vdots & \vdots & \ddots & \vdots \\
X_{n,1} & X_{n,2} & \cdots & X_{n,m}
\end{bmatrix} = \begin{bmatrix}
\tilde{x}_1 \\
\tilde{x}_2 \\
\vdots \\
\tilde{x}_n
\end{bmatrix}
\]  

(1)

\[
Y_{n\times 1} = \begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_m
\end{bmatrix} = \begin{bmatrix}
Y_1 \\
Y_2 \\
\vdots \\
Y_m
\end{bmatrix}
\]  

(2)

\( Y_{n\times 1} \) represents the labels of one sample in the training set, where \( y_i = 0 \) or 1. After the training process, a tree structure is constructed, which can be denoted as a model of \( \tilde{p} \); then, the decision tree can be expressed as follows [21]:

\[
Y' = g(X, \tilde{p})
\]  

(3)

where \( Y' \) represents the output of the decision tree. The main process of AdaBoost is shown in Figure 1.
Figure 1. The principle of AdaBoost Algorithm, where \( X, W, X_i, e_i, g_i(x), \) and \( f(x) \) represent the labeled training samples, the weight set, the weighted sample set, the classification error, the weak classifier coefficient, the weak classifier, and the final strong classifier, respectively.

In the algorithm of AdaBoost, a weight factor \( w \) is introduced for each sample in the training set, where \( i = 1, 2, ..., n \), standing for the index of that sample. \( W_{x:1} \) is defined as the initial sample weight [20]:

\[
W_{x:1} = \begin{bmatrix} w_1 & w_2 & \cdots & w_n \end{bmatrix}^T
\]  

(4)

Here, an operation \( \odot \) is defined as follows [22]:

\[
W_{x:1} \odot X_{x:m} = \begin{bmatrix} w_1 x_{1,1} & w_1 x_{1,2} & \cdots & w_1 x_{1,m} \\
2 & 2 & 2 & \cdots \ \ 2 & 2 & 2 & \cdots \ \ 2 & 2 & 2 & \cdots \\
2 & 2 & 2 & \cdots \ \ 2 & 2 & 2 & \cdots \ \ 2 & 2 & 2 & \cdots \\
\end{bmatrix}
\]  

(5)

During iteration, the weak classifier is trained based on training samples, and then the weight of each sample is updated according to the error rate and classification error of this particular sample. The initial weight of the sample \( W_1 \) is set to [22]

\[
W_{x:1}^1 = \begin{bmatrix} 1/n & 1/n & \cdots & 1/n \end{bmatrix}^T
\]

(6)

The characteristics of training set \( X_1 \) used for the weak classifier are

\[
X_1 = W_{x:1}^1 \odot X_{x:m}
\]

(7)

Furthermore, the output \( Y_{x:1}^1 \) is

\[
Y_{x:1}^1 = g_1(X_1, \hat{p})
\]

(8)

The error rate \( e_1 \) can be obtained by

\[
e_1 = P(g_1(X_1, \hat{p}) \neq Y_{x:1})
\]

(9)

According to the calculated error rate, the weight of weak classifier \( a_1 \) is obtained by

\[
a_1 = \frac{1}{2} \log \frac{1-e_1}{e_1}
\]

(10)

After that, the weight of sample \( w_i^1 \) can be updated by

\[
w_i^1 = \frac{w_i^1}{Z_1} \exp(-a_1 y_i g_1(\vec{x}_i, \hat{p})), i = 1, 2, ..., n
\]

(11)

\[
Z_1 = \sum_{i=1}^{n} w_i^1 \exp(-a_1 y_i g_1(\vec{x}_i, \hat{p}))
\]

(12)
The classification error of the weak classifier generated by AdaBoost will be gradually decreased after iterations of $T$ steps.

AdaBoost combines all the weak classifiers to construct a strong classifier, in which the weak classifier with the lower classification error will obtain more weight.

It is worth noting that the resulting training error drops exponentially and rapidly to zero when the classification error of the weak classifier reaches only a little less than 50%. Therefore, the weak classifier needs to be slightly better than the random classifier.

The whole algorithm of AdaBoost is shown in detail (Algorithm 1), as follows [22].

**Algorithm 1: AdaBoost for supervised classification**

Give the number of iterations $T$ and input the labeled training sample set $F_k$, $k=0$

repeat

According to the weighted training samples, the weak classifier $g_k(X_k, \tilde{p})$ is trained.

Calculate error rate $e_i = P(g_i(X_i, \tilde{p}) \neq Y_i), i = 1, 2, ..., n$

Compute coefficients $a_k = \frac{1}{2} \log \frac{1 - e_k}{e_k}$

Compute $Z_k = \sum_{i=1}^{n} a_k^i \exp(-a_k y_i g_k(X_i, \tilde{p}))$

Update $w_i^{k+1} = \frac{w_i^k}{Z_k} \exp(-a_k y_i g_k(X_i, \tilde{p})), i = 1, 2, ..., n$

Until $k < T$, $k \leftarrow k + 1$

2.2. Principle of BFS Extraction by AdaBoost

In order to solve multiple classification problem of BFS extraction, for a given sample, firstly two different labels are chosen from the training set, and the AdaBoost Algorithm is applied for this exact binary classification problem. Then, this routine is repeated until all different label pairs in the training set are chosen, and finally the final label for the input sample is decided by voting.

In this study, the pseudo-Voigt curve is selected to make it applicable to conditions of different pump pulse widths, and $c = (a, p, \Delta v)$ is defined as a combination vector of Lorentz ratio $a$, curve gain parameter $p$ and curve line width $\Delta v$. Assume that the value range of center frequency $v_0$ is $v_0 = [v_{01}, v_{02}, ..., v_{0n}]$ and the value range of $c$ is $c = [c_1, c_2, ..., c_m]$. The scanning frequency range of curve is $v_1 = [v_{11}, v_{12}, ..., v_{1n}]$. The Brillouin gain set $F_{\text{meck}}$ can be expressed as follows:

$$F_{\text{meck}} = \begin{bmatrix}
S(v_{01}, c_1, v_{11}) & S(v_{01}, c_1, v_{12}) & \cdots & S(v_{01}, c_{1}, v_{1n}) \\
S(v_{01}, c_2, v_{11}) & S(v_{01}, c_2, v_{12}) & \cdots & S(v_{01}, c_{2}, v_{1n}) \\
\vdots & \vdots & \ddots & \vdots \\
S(v_{0m}, c_1, v_{11}) & S(v_{0m}, c_1, v_{12}) & \cdots & S(v_{0m}, c_{1}, v_{1n}) \\
S(v_{02}, c_2, v_{11}) & S(v_{02}, c_2, v_{12}) & \cdots & S(v_{02}, c_{2}, v_{1n}) \\
\vdots & \vdots & \ddots & \vdots \\
S(v_{0m}, c_m, v_{11}) & S(v_{0m}, c_m, v_{12}) & \cdots & S(v_{0m}, c_{m}, v_{1n})
\end{bmatrix}$$

(13)

Here, the Brillouin gain subsets $F_i$ and $F_j$ from $F_{\text{meck}}$ are chosen to be
of 13

\[ F_i = S_{mod}(v_{i0}, c_i, v_i) = \begin{bmatrix} \mathbf{R}(v_{i0}, c_1, v_i) \\
\mathbf{R}(v_{i0}, c_2, v_i) \\
\vdots \\
\mathbf{R}(v_{i0}, c_m, v_i) \end{bmatrix} = \begin{bmatrix} \mathbf{S}(v_{i0}, c_1, v_i) & \mathbf{S}(v_{i0}, c_1, v_i) & \cdots & \mathbf{S}(v_{i0}, c_1, v_i) \\
\mathbf{S}(v_{i0}, c_2, v_i) & \mathbf{S}(v_{i0}, c_2, v_i) & \cdots & \mathbf{S}(v_{i0}, c_2, v_i) \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{S}(v_{i0}, c_m, v_i) & \mathbf{S}(v_{i0}, c_m, v_i) & \cdots & \mathbf{S}(v_{i0}, c_m, v_i) \end{bmatrix} \]

(14)

\[ F_j = S_{mod}(v_{j0}, c_j, v_j) = \begin{bmatrix} \mathbf{R}(v_{j0}, c_1, v_j) \\
\mathbf{R}(v_{j0}, c_2, v_j) \\
\vdots \\
\mathbf{R}(v_{j0}, c_m, v_j) \end{bmatrix} = \begin{bmatrix} \mathbf{S}(v_{j0}, c_1, v_j) & \mathbf{S}(v_{j0}, c_1, v_j) & \cdots & \mathbf{S}(v_{j0}, c_1, v_j) \\
\mathbf{S}(v_{j0}, c_2, v_j) & \mathbf{S}(v_{j0}, c_2, v_j) & \cdots & \mathbf{S}(v_{j0}, c_2, v_j) \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{S}(v_{j0}, c_m, v_j) & \mathbf{S}(v_{j0}, c_m, v_j) & \cdots & \mathbf{S}(v_{j0}, c_m, v_j) \end{bmatrix} \]

(15)

\[ D_1 = (\omega_{11}, \omega_{12}, \ldots, \omega_{1m}); \omega_{1r} = \frac{1}{2m}; r = 1, 2, \ldots, 2m \] is initialized and the weighted sample set \( T_{1,i,j}^1 \) is constructed for the binary classification training of the AdaBoost Algorithm [23]:

\[ T_{1,i,j}^1 = \begin{bmatrix} \omega_{11} \mathbf{R}(f_{i,j}, c_1, f_i) \\
\omega_{12} \mathbf{R}(f_{i,j}, c_2, f_i) \\
\vdots \\
\omega_{1m} \mathbf{R}(f_{i,j}, c_m, f_i) \\
\omega_{j,1} \mathbf{R}(f_{i,j}, c_1, f_i) \\
\omega_{j,1} \mathbf{R}(f_{i,j}, c_2, f_i) \\
\vdots \\
\omega_{j,1} \mathbf{R}(f_{i,j}, c_m, f_i) \\
\omega_{j,2m} \mathbf{R}(f_{i,j}, c_m, f_i) \end{bmatrix} \]

(16)

For a sample set, the corresponding binary classifier is obtained by \( f_{i,j} = \text{sign}\left(\sum_{t=1}^{n} a_t G_t\right) \), as shown in Figure 2. Therefore, classifiers can be expressed as:

\[ f = \begin{bmatrix} f_{1,2} & f_{1,3} & \cdots & f_{1,n} & f_{2,3} & f_{2,4} & \cdots & f_{2,n} & \cdots & f_{n-1,n} \end{bmatrix}^T \]

(17)

For the given BGS \( S_p = S(f_0, c_0, f_0) \), the classifier set is constructed, the output results of the center frequency of each classifier are counted, and the maximum frequency value \( f_{\text{max}} \) is obtained, which should be the extracted BFS corresponding to the given BGS.

Figure 2. BFS extraction process based on AdaBoost, where \( F, F_{i,j}, f_i(x), R_i, \) and \( f_{\text{max}} \) are the classifier set, the binary label sample set, the binary classifier, the classification result, and the BFS of output, separately.

3. Simulation

The performance of AdaBoost extraction is studied by simulation. The AdaBoost Algorithm, using ideal training samples and training samples with noise, are trained separately.

Here, the ideal training sample can be expressed as [14]:

\[ f_v = p \frac{a}{1 + 4\left(\frac{v_{i1} - v_{j0}}{\Delta v}\right)^2} + p(1 - a) \exp[-4\ln 2\left(\frac{v_{i1} - v_{j0}}{\Delta v}\right)^2] \]

(18)
When the ideal samples are used for training, the Lorentz ratio \( \alpha \) varies from 0 to 1, with an interval of 0.2. The width of the curves \( \Delta v \) ranges from 10 MHz to 50 MHz, and the interval is 5 MHz. The curve gain parameter \( \gamma \) is fixed at 1. The center frequency \( v_0 \) ranges from 10,800 MHz to 10,980 MHz, which is the same as the frequency scanning range with an interval of 1 MHz. Therefore, \( 6 \times 9 \times 181 \) samples can be obtained for model training.

For the training samples with noise, a Gaussian white noise is introduced into the pseudo-Voigt curve. The width of the curves, the curve gain parameter, the center frequency, and the frequency scanning range are the same as that of the ideal training samples. The Gaussian white noise with mean value of 0 and different standard deviations (0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40) are added to the training samples, separately. The process with different standard deviations is repeated 5 times, so \( 6 \times 9 \times 61 \times 5 \times 8 \) samples are finally obtained for model training.

After model training, the model is applied to a test. During the test, the pseudo-Voigt curve with noise is used to simulate BGS, where the line width is fixed at 50 MHz, suggesting a pump signal with the pulse width of around 20 ns. The center frequency is fixed at 10,860 MHz, and the Lorentz ratio is fixed at 1. Here, the SNR is defined as the ratio between the mean amplitude of the Brillouin peak or trace and its standard deviation, which is proportional to the amplitude instead of the power [14].

### 3.1. BGS Extraction under Different SNR

The Root Mean Square Error (RMSE) and uncertainty under different SNR based on both training samples, with and without noise, is shown in Figure 3. With the increase in SNR, both the RMSE and uncertainty of the AdaBoost Algorithm decrease, and the values of RMSE and uncertainty are similar under different SNR. It is obvious that the algorithm model obtained by using training samples with noise is better than that without noise. This may be because the pseudo-Voigt curve without noise has only one data sample based on the same feature, while for the pseudo-Voigt curve with noise, there are 40 data samples used for classification training under the same feature.

![Figure 3. BFS extraction based on training samples with or without noise under different SNR.](image)

### 3.2. BGS Extraction under Different Cut-Off Frequency

Based on the frequency range of 10,800–10,980 MHz, the BGS extraction under different SNR is tested. Here, the initial frequency is fixed at 10,800 MHz and the cut-off frequency is changed. The result is shown in Figure 4. The SNR is fixed at 7 dB.

With the increase in cut-off frequency, both RMSE and uncertainty gradually decrease, and the extraction error tends to be converged when the cut-off frequency is larger than 10,880 MHz, as shown in Figure 4. With the decrease in cut-off frequency, the BGS data set of BOTDA decreases, which leads to the increase in the extracting error.
3.3. **BGS Extraction under Different Frequency Range**

The cut-off frequency is also fixed and the extraction performance under different initial frequencies is investigated. Results show that when the cut-off frequency is fixed at 10,980 MHz and the initial frequency is changed, the extraction error tends to be stable when the initial frequency is less than 10,835 MHz.

Therefore, further analysis is conducted for the frequency range between 10,825 MHz and 10,895 MHz. The results are shown in Figure 5. With the decrease in the frequency range, the BFS extraction accuracy decreases, and the feature number of BGS data increases with frequency range, so the BGS extraction accuracy could be stable when the frequency range is larger than 50 MHz. Overall, AdaBoost can retain extraction accuracy when the frequency range is small.

3.4. **BGS Extraction under Different Frequency Step**

The frequency step determines the number of features in each feature vector, i.e., the number of data points on each BGS. Here, the BGS extraction at different frequency step is studied. The results are shown in Figure 6. As the frequency step increases, the extraction performance of AdaBoost gradually deteriorates. Figure 6 shows that with a larger frequency step, the feature set becomes sparser, which could introduce more error into BFS extraction.
Figure 6. BFS extraction under different frequency step.

4. BOTDA Setup and Experiment Results

The BFS extraction by the AdaBoost Algorithm is studied experimentally. The configuration of the experiment is shown in Figure 7.

Figure 7. BOTDA experimental setup. ISO: Isolator, PRG: Pulse pattern generator. EOM: Electro-optic modulator. PS: Polarization scrambler, PD: Photodetector, MRG: Microwave generator, FBG: Fiber grating filter, FUT: Fiber under test, AD: Analog-digital converter, EDFA: Erbium-doped fiber amplifier.

In the experiment, a narrow-line width laser with the wavelength of 1550 nm is used as the light source. The signal from the laser source is divided into two branches by a 50:50 coupler after passing through the optical isolator. At the upper branch, the light is modulated by an electro-optic modulator to generate pump light pulse. The power of the pump light is amplified by an Erbium-doped fiber amplifier. A polarization scrambler is introduced to suppress the polarization fluctuations. After passing through the scrambler, the pump pulse goes into the optical circulator and enters the Fiber Under Test (FUT). At the lower branch, the microwave source generates a microwave signal with the frequency of 10.500–11.000 GHz, which is modulated to the light by an electro-optic modulator. The back-scattered signal is then detected by the photodetector and converted into an electrical signal. The data are received and quantized by a 250 MSa/s acquisition card. The FUT is a 1 km single-mode optical fiber, and the last section of 80 m is heated in a thermostatic
water tank, corresponding to 200 sampling points. The BGS distribution measured is shown in Figure 8. The width of the pump pulse is 25 ns, and the averaging time is 10,000. The frequency scanning step is 1 MHz.

![Figure 8](image)

Figure 8. (a) Measured BGS distribution along 1000 km FUT with last 1000 m section heated at 30 °C, and (b) BFS distribution.

The tolerance of AdaBoost to different SNR is also studied, as shown in Figure 9, where the width of pump pulse is 20 ns and the frequency step is 1 MHz. The averaging time is selected to be 2, 4, 8, 16, and 50, separately, corresponding to the measured SNRs of 5.52 dB, 6.78 dB, 7.96 dB, 8.86 dB, and 10.96 dB, respectively. The trend of the experiment is consistent with that of the simulation. With the increase in SNR, the extraction accuracy of BGS decreases gradually.

![Figure 9](image)

Figure 9. Experimental results under different SNR.

Moreover, the performance of the algorithm is also investigated under different cut-off frequency. The initial frequency is fixed at 10,800 MHz, the pump pulse is 20 ns frequency step at 1 MHz, and the average time is 4, corresponding to an SNR of 6.78 dB. The result is shown in Figure 10. The experimental result is consistent with the simulation result. The performance of the AdaBoost Algorithm becomes stable with the increase in cut-off frequency.
Similarly, in order to verify the relationship between AdaBoost performance and frequency range, tests are conducted for the frequency range of 10,825–10,895 MHz. The result is shown in Figure 11. As the frequency range decreases, the BFS extraction performance of AdaBoost gradually decreases, which is consistent with the numerical simulation.

Here, we compare extraction by both AdaBoost and SVM. The result is shown in a newly added figure (Figure 12). Figure 12 shows that the RMSE of the extracted BFS decreases rapidly with the increase in frequency range, due to the increase in the sampling points of BGS. RMSE of the BFS extracted by AdaBoost varies from 5.8 MHz down to 2.6 MHz, while for SVM, the RMSE varies from 6.3 MHz down to 1.5 MHz. When the frequency range is less than 35 MHz, the accuracy (RMSE) of AdaBoost is better than that of SVM. The reason may be that we use linear kernel function to train the SVM, which introduces error for small sampling sets. It is interesting to note that when the frequency range is larger than 35 MHz, the accuracy (RMSE) of SVM shows a better performance than that of AdaBoost, and the deviation between both methods is about 1 MHz. This suggests that the SVM with the linear kernel is a better choice for sampling sets which cover a large range of BGS.
Finally, the performance of the algorithm is also studied experimentally under different frequency step, as shown in Figure 13. With the increase in frequency step, the RMSE and uncertainty of the AdaBoost Algorithm gradually increase, which verifies the simulation conclusion that the extraction error of BFS by AdaBoost gradually increases when the sparsity of the feature set becomes larger and the number of features becomes smaller.

Figure 13. Experimental results under different frequency step.

The simulation and experimental results are compared. Although the trend of the experiment agrees with that of the simulation, there is a deviation between the result of the simulation and that of the experiment. The deviation is about 3 MHz. This may be because the type of noise source is different, e.g., in the simulation, we use Gaussian white noise as the noise source, while in the experiment, the actual noise may include the amplified spontaneous noise (ASE) from the Erbium-doped fiber amplifier, the relative intensity noise (RIN) and phase noise from the laser source, and the thermal noise from the photo-detector.

Distributed fiber sensors can be used to measure the temperature or strain distribution along the fiber. If the fiber is bound tightly with a large-scale structure, by using distributed fiber sensors, we can obtain the total strain distribution of the bridge.

In principle, when the pump and probe enter the fiber from two ends separately, Brillouin scattering may occur, and there will be a BGS at each position along the fiber. By extracting methods, we can obtain BFS from the peak of BGS.

Two different extracting methods are also compared. For the case when BGS has intense noise (shown in Figure 14), curve-fitting does not always lead to a good fitting result, due to possibly poor initial value settings. It is possible that the curve-fitting could result...
in a fitting failure, as shown in Figure 14. However, for our method of AdaBoost, we can still find BFS from a noisy BGS. Take Figure 14, for example; here, the BFS extraction by AdaBoost is about 10,860 MHz, and the extracting error is near 1 MHz. We believe this may show the advantage of the proposed AdaBoost extracting method.

![Figure 14. Curve fitting, (a) Error fitting, and (b) Inappropriate fitting.](image)

5. Conclusions

In this study, we propose a novel method of BFS extraction using the AdaBoost Algorithm. The extracted BFS are divided into different categories, and the weak classifiers constructed by decision trees are combined to form strong classifiers. The RMSE and uncertainty of the extracted BFS can be 1 MHz with a frequency scanning step of 1 MHz and an SNR of 11 dB. The BFS extraction performance under different SNR, cut-off frequency, frequency range, and frequency step is also studied by both simulation and experiment. Results show that the BFS extraction error increases with the decrease in SNR and frequency range, and with the increase in frequency step. The trend of the experiment agrees with that of the simulation. In addition, we compare the extracting performance of both AdaBoost and SVM. Results show that when the frequency range is less than 35 MHz, the accuracy (RMSE) of AdaBoost is better than that of SVM. This may be because we use the linear kernel function to train the SVM, which introduces error for small sampling sets. The comparison between AdaBoost and curve-fitting is also presented, and the result shows that our proposed BFS extracting method is more stable than curve-fitting, especially for the case when BGS has intense noise. It is believed that AdaBoost is a good candidate for BFS extraction from weak BGS. We would like to further investigate the performance of AdaBoost and other methods, such as ANN.

**Author Contributions:** Concept and structure of this study, F.X.; resources, H.Z. and Y.Q.; writing—original draft preparation, F.X.; writing—review and editing, F.X., H.Z., and S.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Natural Science Foundation of China (NSFC) (61675184, 61275124).

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

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

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

**References**

1. Hicke, K.; Hussels, M.T.; Eisermann, R. Condition monitoring of industrial infrastructures using distributed fibre optic acoustic sensors. In Proceedings of the 2017 25th Optical Fiber Sensors Conference (OFS), Jeju, Korea, 24–28 April 2017; pp. 1–4.
2. Mariñelarena, J.; Urricelqui, J.; Loayssa, A. Extension of the dynamic range in slope-assisted coherent BOTDA sensors. In Proceedings of the 25th Optical Fiber Sensors Conference (OFS), Jeju, Korea, 24–28 April 2017; pp. 1–4.
3. Cui, Q.; Pamukcu, S.; Xiao, W. Truly Distributed Fiber Vibration Sensor Using Pulse Base BOTDA With Wide Dynamic Range. *IEEE Photonics Technol. Lett.* **2011**, *23*, 1887–1889.
4. Bao, X.; Chen, L. Recent Progress in Brillouin Scattering Based Fiber Sensors. *Sensors* **2011**, *11*, 4152–4187.
5. Soto, M.A.; Thévenaz, L. Modeling and evaluating the performance of Brillouin distributed optical fiber sensors. *Opt. Express* **2013**, *21*, 31347–31366.
6. Wu, H.; Chang, Y.; Zhao, C. Distributed Brillouin frequency shift extraction via a convolutional neural network. *Photonics Res.* **2020**, *8*, 690–697.
7. Yaswanth, K.; Somepalli, B.; Khankhoje, U.K. Accurate Estimation of Brillouin Frequency Shift in Brillouin Optical Correlation Domain Analysis. *J. Lightwave Technol.* **2019**, *37*, 5875–5884.
8. Farahani, M.A.; Castillo-Guerra, E.; Colpitts, B.G. A Detailed Evaluation of the Correlation-Based Method Used for Estimation of the Brillouin Frequency Shift in BOTDA Sensors. *IEEE Sens. J.* **2013**, *13*, 4589–4598.
9. Farahani, M.A.; Castillo-Guerra, E.; Colpitts, B.G. Accurate estimation of Brillouin frequency shift in Brillouin optical time domain analysis sensors using cross correlation. *Opt. Lett.* **2011**, *36*, 4275–4277.
10. Azad, A.K.; Wang, L.; Guo, N. Signal processing using artificial neural network for BOTDA sensor system. *Opt. Express* **2016**, *24*, 6769–6783.
11. Ruiz-Lombera, R.; Fuentes, A.; Rodriguez-Cobo, L. Simultaneous Temperature and Strain Discrimination in a Conventional BOTDA via Artificial Neural Networks. *J. Lightwave Technol.* **2018**, *36*, 2114–2121.
12. Azad, A.K.; Wang, L.; Guo, N. Temperature sensing in BOTDA system by using artificial neural network. *Electron. Lett.* **2015**, *51*, 1578–1580.
13. Wu, H.; Wang, L.; Guo, N. Brillouin Optical Time-Domain Analyzer Assisted by Support Vector Machine for Ultrafast Temperature Extraction. *J. Lightwave Technol.* **2017**, *35*, 4159–4167.
14. Wu, H.; Wang, L.; Zhao, Z. Support Vector Machine based Differential Pulse-width Pair Brillouin Optical Time Domain Analyzer. *IEEE Photonics J.* **2018**, *10*, 1–11.
15. Zhu, H.; Yu, L.; Zhang, Y. Optimized Support Vector Machine Assisted BOTDA for Temperature Extraction With Accuracy Enhancement. *IEEE Photonics J.* **2019**, *12*, 1–15.
16. Schapire, R.E.; Freund, Y.; Bartlett, P. Boosting the margin: A new explanation for the effectiveness of voting methods. *Ann. Stat.* **1998**, *26*, 1651–1686.
17. Dietterich, T.G. An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. *Mach. Learn.* **2000**, *40*, 139–157.
18. Cherubini, D.; Fanni, A.; Montisci, A. Inversion of MLP neural networks for direct solution of inverse problems. *IEEE Trans. Magn.* **2005**, *41*, 1784–1787.
19. Schwenk, H.; Bengio, Y. Boosting Neural Networks. *Neural Comput.* **2000**, *12*, 1869–1887.
20. Harrington, P. *Machine Learning in Action*; Manning Publications, New York, U.S.A.: 2012; pp. 154–196.
21. Zheng, H. Extraction of Brillouin frequency shift in Brillouin distributed fiber sensors by neighbors-based machine learning. In *Proceedings of the Advanced Sensor Systems and Applications X*, Online, 10 October 2020.
22. Tan, P.N.; Steinback, M.; Kumar, V. *Introduction to Data Mining*; Pearson: London, UK, 2011; pp. 213–236.
23. Yang, J.; Ma, H.; Dou, J. Harmonic Characteristics Data-Driven THD Prediction Method for LEDs Using MEA-GRNN and Improved-AdaBoost Algorithm. *IEEE Access* **2021**, *9*, 31297–31308.