Recurrent Neural Detection of Time–Frequency Overlapped Interference Signals

Qianqian Wu, Zhuo Sun(✉), and Xue Zhou

Wireless Signal Processing and Network Laboratory, Beijing University of Posts and Telecommunications, Beijing 100876, China
{wuqq, zhuosun, 277210680}@bupt.edu.cn

Abstract. For interfering signals overlap with normal signals in both time and frequency domain, it is difficult to detect them. Therefore, this paper proposes a novel bidirectional recurrent neural network-based interference detection method. By utilizing the ability of recurrent neural network of extracting the nonlinear features of the time series context, the model can get a prediction of following signal samples and calculate the difference between prediction signal and original signal to do interference detection. The proposed method can achieve a better sensitivity and determine the exact location of the complete interfering signal. In the experiment part, we demonstrate the efficacy of this method in multiple typical scenarios of time–frequency overlapped wireless signals.

1 Introduction

The development of wireless communication makes it be widely used in various fields, but the electromagnetic environment is complex and changeable, and the reliability of the communication system is still threatened by interference. Therefore, the anti-interference technology is necessary to ensure the communication reliability, and the interference detection technology is the basis and key of the communication anti-interference technology.

The purpose of interference detection is to determine whether there are interfering signals in received signals and then feed back to the transmitter or command center to take effective anti-interference measures. Traditional wireless interference detection approaches include time domain and transform domain-based energy detection algorithm (e.g., consecutive mean excision (CME) and forward consecutive mean excision (FCME)). Some approaches analyze the received signal strength indicator (RSSI) samples in the frequency and time domain [1,2] or to perform a cyclostationary signal analysis and blind signal detection and other spectrum sensing techniques [3]. These methods can well detect many kinds of interfering signals. However, there are many interfering signals whose power is small and frequency is the same as original signals in the

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 67–75, 2020.
https://doi.org/10.1007/978-981-15-0187-6_8

actual communication environment. When these interfering signals are superimposed on the original signal, the time and the frequency domain features do not change significantly. We call these interfering signals as time–frequency overlapped signals. Obviously, the time–frequency overlapped signals are hard to detect by above methods.

The research found that the neural network can better extract the time and frequency domain feature of signals. The feature can be used as signal fingerprint to classify signals [4,5]; therefore, many interference detection methods based on deep learning emerged in recent years. A sequential autoencoder framework is introduced to distinguish normal and interfering signal in [6]. They calculate the difference between original signals and reconstructed signals by autoencoder framework. In [7,8], predicting future signals from known signals by utilizing the prediction function of recurrent neural network and then taking the difference between original and predictive signals as a feature to do interference detection. However, these methods have not good detection performance under low signal-to-interference-plus-noise ratio (SINR) conditions, especially for time–frequency overlapped interfering signals.

In this paper, we present a bidirectional recurrent neural network (BI-RNN)-based interference detection structure, which can utilize the correlation between the front and back sampling points to predict the data of the intermediate position. The structure can do bidirectional training and has the better prediction performance; hence, the difference between predictive and original signals at the interference-containing part is larger. And then in prediction process, we adopt special training labels for noise reduction; therefore, the detection accuracy is improved under low SINR condition. Finally, in the detection process, we use the feature correlation classification instead of simply taking a decision threshold, which reduces the contingency of the decision process and can completely identify the complete interfering signal.

The rest of this paper is organized as follows. In Sect. 2, the interfering signal and the principle of interference detection are described. Then, the neural network model and the special training method are explained in Sect. 3. Section 4 shows the performance of interference detection and analysis of influencing factors. Finally, Sect. 5 concludes the paper and suggests future work.

2 Problem Formulation

When there is no interference, the received signal can be represented by Eqs. 1, and 2 and the received signal contains interference.

$$r(t) = s(t) + n(t), \tag{1}$$

$$r_j(t) = s(t) + n(t) + j(t). \tag{2}$$

where $s(t)$ is a carrier signal transmitted by transmitter, $n(t)$ represents noise, and $j(t)$ is a interfering signal(In this paper, we consider $j(t)$ as the time–frequency overlapped interfering signal.) The aim of interference detection is to find the exact position of $j(t)$.

As shown in Fig. 1, a predictor f is trained by normal signal r and we can get the predictive signal $\widehat{r}$ from f. Then, calculate the difference e between r and $\widehat{r}$ by Eq. 3 and e can be considered as a feature to train a classifier for interference detection. Similarly, when the interference-containing signal r_j uses f to predict, we can get the predictive signal $\widehat{r_j}$ and the difference e_j is calculated by Eq. 4. It is obvious that e_j is greater than e, because the presence of interference destroys the correlation of the original signal, which makes f unable to accurately predict the interference-containing signal. Finally, the classifier can find interfering signal based on the difference between e and e_j.

$$e = |r - \widehat{r}|\,, \tag{3}$$

$$e_j = |r_j - \widehat{r_j}|\,. \tag{4}$$

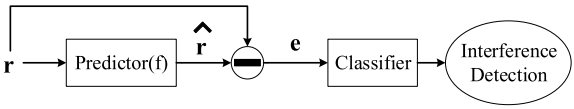

Fig. 1. General process of interference detection

3 BI-RNN-Based Prediction of Time Series Signal

It can be seen from Sect. 2 that the core of the interference detection method proposed in this paper is the construction of prediction model. Research shows that RNN can extract the nonlinear features of time series [9], so choose RNN predictor here. As a special RNN, bidirectional Long Short-Term Memory (BI-LSTM) adds a set of weight parameters for backward calculation; therefore, it can also utilize the data information after the sample points to be predicted for prediction. For that reason, we built the BI-LSTM-based model for signal prediction.

3.1 Prediction Model

At the very beginning, we should preprocess the signal into a form suitable for BI-LSTM. Supposing the time series of received signal is $Y = \left\{y_1, y_2, y_3, \ldots, y_t, y_{(t+1)}, \ldots, y_{(M)}\right\}$, choose $Y_{\text{train}} = \left\{y_1, y_2, y_3, \ldots, y_t, y_{(t+C+1)}, \ldots, y_{(M)}\right\}$ as training set (C and M are fixed values) and $Y_{\text{label}} = \left\{y_{(t+1)}, y_{(t+2)}, \ldots, y_{(t+C)}\right\}$ as label to predict $Y_{\text{pre}} = \left\{\widehat{(y_{(t+1)})}, \widehat{(y_{(t+2)})}, \ldots, \widehat{(y_{(t+C)})}\right\}$. Through a lot of iterations calculation, the network will constantly adjust the weights to make Y_{label} and Y_{pre} closer and closer. Then the computed Y_{error} is used to determine if it is interference-containing signal. Y_{error} is defined as $Y_{\text{error}} = |Y_{\text{label}} - Y_{\text{pre}}|$.

The specific process of signal prediction is shown as Fig. 2. Divide N time-banks of M time steps shifting by C time steps between adjacent chunks. If the total signal sampling points is T, we can calculate the number of time-banks by $N = \frac{T-M}{C} + 1$.

For each time-bank, remove p consecutive sample points inside, and set the remaining sample points as one input. So the output from each input is the predictive p samples, and the output from the N inputs is a continuous predictive signal. In theory, the larger the value of M, the better the prediction performance. In Fig. 2, we set $M = 110$ and $p = 1$.

The reason why we use BI-LSTM is that BI-LSTM adds a delay between the input and the target to give the network some time to add future context information for prediction [10,11]. The hidden layers of the Bi-LSTM store two sets of parameters, one for forward calculation and the other for backward calculation, and the final outputs depend on both parameters. But experiments have shown that when the sequence after the predicted value is too long, the network will pay more attention on the behind part and the predictive performance will worsen. Therefore, should choose the appropriate value (here we set 10). In order to verify the validity of the proposed method, we compared it with the predictive model of LSTM mentioned in [7]. The BI-LSTM predictive model is described as follows.

- The stacked BI-LSTM sequence predictor model is implemented with a 3-layer BI-LSTM followed by a fully connected layer culminating in a linear activation for output. The dropout between each layer is 0.2 and chooses mean squared error loss function (MSE) as the loss function. Besides, the number of hidden layer is 128, and batch size is 64.

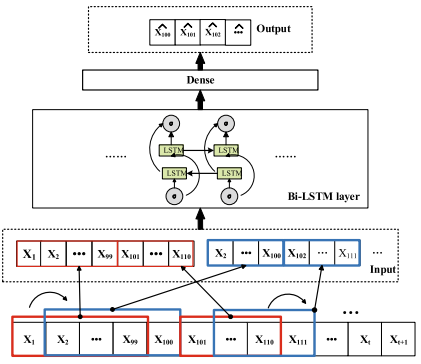

Fig. 2. Stacked BI-LSTM prediction model for signal prediction

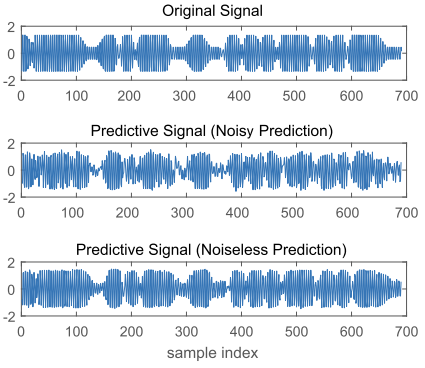

Fig. 3. Predictive signal by using BI-LSTM

3.2 Model Training for Noiseless Prediction

During the research, it was found that when the SINR is relatively small, the network extract the signal features hardly; therefore, the predictive signal and the original signal are very different. The implementation of speech enhancement uses LSTM by learning the correlation between noisy signals and noiseless signals in [12]; hence, we are inspired by speech enhancement to propose a novel training method which can improve the prediction performance under low SINR conditions. We call this prediction process as noiseless prediction.

In noiseless prediction, the training data is the received signal with noise, but the input label during training is the corresponding noiseless signal sampling point. This method makes the network focus only on the correlation of the signal itself, regardless of the irregularity of noise; therefore, the output is closer to the noiseless signal. Figure 3 depicts the digital modulation signal predicted by BI-LSTM, when the SINR = 6 db. The first picture is the original signal without noise for comparison, and then the second is the predictive signal when use noise signal as label, and the third is the predictive signal when use noiseless signal as label. From the figure, we can see that noiseless prediction model has better predictive performance than the noisy prediction model.

3.3 Prediction Performance

To measure the effectiveness of different neural network, the mean absolute errors (MAE) and mean absolute percentage errors (MAPE) are computed in Eqs. 5 and 6:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |x_i - \widehat{x_i}|\,, \tag{5}$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n} \left|\frac{x_i - \widehat{x_i}}{x_i}\right|. \tag{6}$$

where x_i is the actual signal sampling point at i_{th}, $\widehat{x_i}$ is the predictive signal sampling point, and n is the total number of test signal sampling points.

All the predictive result of different models in this section trained and tested multiple times to eliminate outliers. We compare the prediction performance of LSTM, BI-LSTM and the case of noiseless prediction in Table 1.

We can see from Table 1 that the prediction result is getting better with the increase of SINR, but the influence becomes less obvious when the SINR reaches a certain level. This proves that Gaussian white noise does interfere with the learning ability of the network. In addition, the performance of BI-LSTM model is better than that of LSTM, although not obvious, which indicates that the sequence correlation after a certain signal sampling point will affect the prediction result of model. Finally, we can also see that noiseless prediction obtains the best results, especially under low SINR condition. This proves that noiseless prediction does have good denoise ability.

Table 1. Prediction performance of three models

	Performance					
Model	SINR	0 db	4 db	8 db	12 db	16 db
LSTM	MAE	0.437	0.422	0.277	0.203	0.118
	MAPE	0.796	0.950	0.623	0.456	0.278
BI-LSTM (noisy prediction)	MAE	0.565	0.413	0.267	0.168	0.109
	MAPE	1.099	0.945	0.620	0.454	0.275
BI-LSTM (noiseless prediction)	MAE	0.257	0.205	0.163	0.103	0.07
	MAPE	0.592	0.548	0.361	0.248	0.146

Figure 4a shows the comparison of original signal and predictive signal based on digital modulation when the SINR = 12 db. Similarly, Fig. 4b demonstrates the case of FM modulation. We can find from the pictures that the difference between the predictive signal and the original noiseless signal in the normal part is much smaller than in the interference-containing part. So the difference of predictive signal and original signal can be used for interference detection.

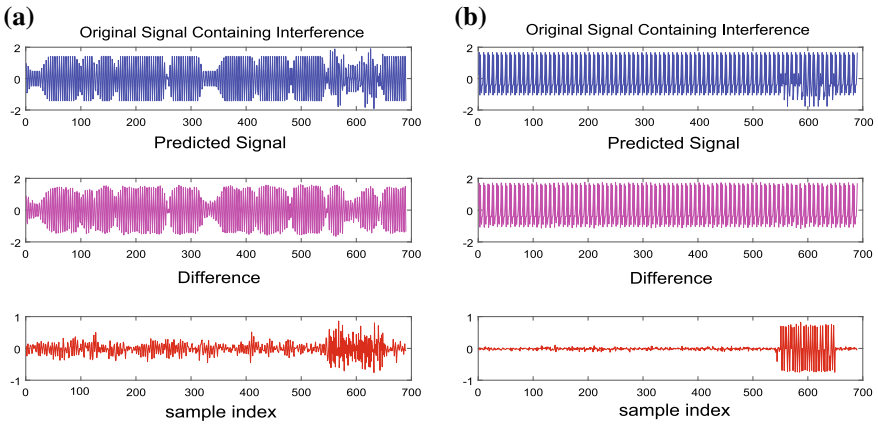

Fig. 4. a is the difference between predictive signal and original signal (digital modulation), and **b** is the difference between predictive signal and original signal (analog modulation)

4 Interference Detection

4.1 Classifier for Interference Detection

The features calculated by Sect. 3 can be used for interference detection. In order to overcome the factor of sample imbalance, select support vector data description (SVDD) here.

During the detection process, it is found that when the bias of each sampling point is used as the feature to train SVDD classifier, it is hard to determine the specific location of interference signal. So the feature is windowed according to the time step and is detected according to the waveform and correlations of the plurality of feature points to overcome the contingency. The windowing process is expressed as Fig. 5a. The length of the window (Wlen) we selected is 10, and the step size (Step) is 1 here. The experimental results are shown in Fig. 5b. The part marked with purple in the first picture is the interference-containing signal. The second picture shows the result of test. The part with a value of −1 is the signal that is judged to be interference, and the part with a value of 1 is the signal that is judged to be normal. It can be seen that most of the interference signal sampling points can be detected.

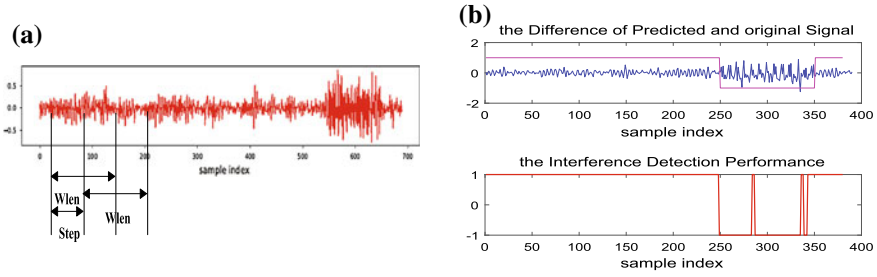

Fig. 5. a is the windowing process, and **b** is the interference detection performance at 12 db (QPSK modulation)

In order to evaluate performance comprehensively, four typical experimental scenarios are set as follows, which represent various types of interference that may occur in the actual communication system. In addition, to reduce the influence of the amplitude, we have done the amplitude normalization on each signal.

- Normal signal is interfered by interference signal with different modulation mode:

1. The 16QAM signal is normal and the superimposed QPSK signal with the same sampling rate, symbol rate, and carrier rate is interference (QAM-QPSK).
2. The 16QAM signal is normal and the superimposed FM signal is interference (QAM-FM).
3. The FM signal is normal, and the superimposed DSSS signal is interference (FM-DSSS).

- Normal signal is interfered by interference signal with different symbol rates:

4. The 16QAM signal is normal, and the superimposed 16QAM signal with only different symbol rate is interference (QAM-QAM).

4.2 Performance Evaluation

We repeat these experiences in three models above-mentioned and get Fig. 6 which show the F1 score of the performance of three models under different SINR. $F1_{\text{score}}$ is calculated as $F1_{\text{score}} = \frac{2PR}{P+R}$ (where P represents precision and R represents recall). Obviously, when using the noiseless signal as label, we get the best performance, especially at low SINR. In the case of noise label, although the MAE and MAPE values of Bi-LSTM are smaller than those of LSTM, the detection accuracy is not much higher.

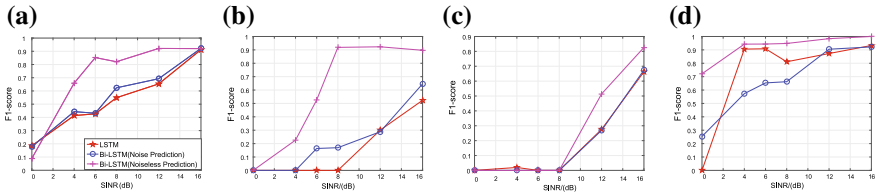

Fig. 6. Interference detection performance of three models: **a** is the detection performance of QAM-FM signal, **b** is the detection performance of QAM-QPSK signal, **c** is the detection performance of QAM-QAM signal, **d** is the detection performance of FM-DSSS signal

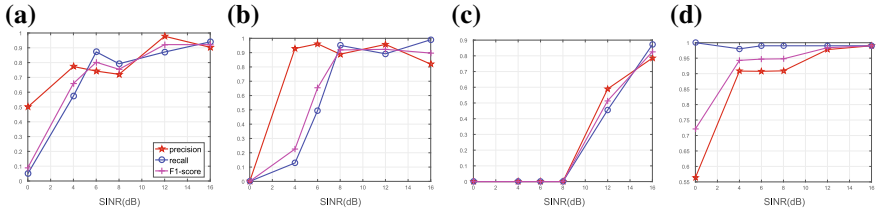

Fig. 7. Interference detection performance of BI-LSTM (noiseless prediction) under different evaluation criteria: **a** is the detection performance of QAM-FM signal, **b** is the detection performance of QAM-QPSK signal, **c** is the detection performance of QAM-QAM signal, **d** is the detection performance of FM-DSSS signal

Figure 7 shows the experimental results of BI-LSTM when the noiseless signal as label. We can see that the detection precision and recall significantly be improved with the increase of SINR. Almost all the interferences are determined to be normal when the SINR = 0 db. When the SINR is higher than 6 db, perfect detection performance is achieved. Among the four scenarios, only the QAM interference with different symbol rates is the most difficultly to detect. The DSSS interference in FM modulation is the most easily to detect, which shows that the signal of digital modulation is hard to predict because of the randomness of its own symbols.

5 Conclusion

In this paper, a more robust method is proposed for wireless signal interference detection at low SINR and is capable of localizing to each interference signal sampling point. The results show that the model can implement the interference detection which not applicable to traditional methods, and the detection accuracy is better than other neural network models in the time–frequency overlapped interfering signal. We believe the result can be used in interference detection in complex communication environments. However, the experiment proves that the detection result is relatively poor for the case where the SINR is lower than 6 db. So we will work on how to improve detection performance under low SINR condition next.

References

1. Rayanchu S, Patro A, Banerjee S (2011) Airshark: detecting non-WiFi RF devices using commodity WiFi hardware. In: ACM SIGCOMM conference on internet measurement conference
2. Lakshminarayanan K, Sapra S, Seshan S, Steenkiste P (2009) RFDump: an architecture for monitoring the wireless ether. In: International conference on emerging networking experiments & technologies
3. Chen X, Chen H, Chen Y (2011) Spectrum sensing for cognitive ultra-wideband based on fractal dimensions. In: Fourth international workshop on Chaos-fractals theories & applications
4. Merchant K, Revay S, Stantchev G, Nousain B (2018) Deep learning for RF device fingerprinting in cognitive communication networks. IEEE J Sel Top Sig Process 12(1):160–167
5. Sainath TN, Li B (2016) Modeling time-frequency patterns with LSTM vs. convolutional architectures for LVCSR tasks. In: Interspeech
6. Mirza AH, Cosan S (2018) Computer network intrusion detection using sequential LSTM neural networks autoencoders. In: 2018 26th signal processing and communications applications conference (SIU), May 2018, pp 1–4
7. O'Shea TJ, Clancy TC, McGwier RW (2016) Recurrent neural radio anomaly detection. CoRR, arXiv:1611.00301
8. Malhotra P, Vig L, Shroff G, Agarwal P (2015) Long short term memory networks for anomaly detection in time series. In: Proceedings. Presses universitaires de Louvain, p 89
9. Xiao Q, Si Y (2017) Time series prediction using graph model. In: 2017 3rd IEEE international conference on computer and communications (ICCC), pp 1358–1361
10. Cui Z, Ke R, Wang Y (2018) Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. CoRR, arXiv:1801.02143
11. Chen Y, Qian J, Yang J, Jin Z (2016) Face alignment with cascaded bidirectional LSTM neural networks. In: 2016 23rd international conference on pattern recognition (ICPR), pp 313–318
12. Kolbk M, Tan Z, Jensen J (2016) Speech enhancement using long short-term memory based recurrent neural networks for noise robust speaker verification. In: 2016 IEEE spoken language technology workshop (SLT), pp 305–311