EDChannel: channel prediction of backscatter communication network based on encoder-decoder

Dengao Li1,3 · Yongxin Wen2,4 · Shuang Xu1,3 · Qiang Wang2,4 · Ruiqin Bai2,4 · Jumin Zhao2,4

Abstract
Backscatter communication networks have attracted much attention due to their small size and low power waste, but their spectrum resources are very limited and are often affected by link bursts. Channel prediction is a method to effectively utilize the spectrum resources and improve communication quality. Most channel prediction methods have failed to consider both spatial and frequency diversity. Meanwhile, there are still deficiencies in the existing channel detection methods in terms of overhead and hardware dependency. For the above reasons, we design a sequence-to-sequence channel prediction scheme. Our scheme is designed with three modules. The channel prediction module uses an encoder-decoder based deep learning model (EDChannel) to predict the sequence of channel indicator measurements. The channel detection module decides whether to perform a channel detection by a trigger that reflects the prediction effect. The channel selection module performs channel selection based on the channel coefficients of the prediction results. We use a commercial reader to collect data in a real environment, and build an EDChannel model based on the deep learning module of Tensorflow and Keras. As a result, we have implemented the channel prediction module and completed the overall channel selection process. The experimental results show that the EDChannel algorithm has higher prediction accuracy than the previous state-of-the-art methods. The overall throughput of our scheme is improved by approximately 2.9% and 14.1% over Zhao’s scheme in both stable and unstable environments.

Keywords Backscatter communication · Channel prediction · Deep learning · Encoder-decoder

1 Introduction
Backscatter communication network, a short-distance, ultra-low power waste, low-cost wireless communication technology, is widely used to build large-scale deployment of sensor networks [1,2]. With the rapid development of the Internet of Things technology and the increasing demand for wireless sensor networks, backscatter communication technology has attracted widespread attention from academic circles and industrial circles, and many potential application scenes have also been mentioned a lot such as target positioning, gesture recognition, and medical testing [3], etc. At the same time, backscatter communication networks have been widely used in various wireless platforms such as Radio Frequency Identification (RFID) [4], Wi-Fi, and Bluetooth, etc. As a result, we can expect that in the near future, devices related to
backscatter communication networks will be used on a large scale in everyday life.

However, the backscatter communication network uses the energy obtained by the radio frequency signal at the transmitting end to transmit data [5]. This low-power transmission method makes the quality of the communication not guaranteed in dynamic channels. At the same time, backscatter communication networks are usually deployed in complex network environments, such as factories, shelves and so on. The above-mentioned factors cause the state of the channel to exhibit a high degree of burstiness, which seriously affects the communication quality of the backscatter communication network. Therefore, researchers generally reduce transmission loss in two ways to ensure the higher throughput of the backscatter communication network. One is selecting channels through predicted channel metrics, and selecting channels with better channel quality for communication [6]; the other is adapting rate based on the predicted channel metrics, which dynamically adjusts the communication rate according to the changes in channel quality [7]. Studying the channel prediction in the backscattering communication network is extremely important. Through extensive preliminary experiments, we observe that there are many causes of channel bursts in backscatter communication networks, such as tag collisions due to the addition of new tags, sudden changes in channel metrics due to tag movement, changes in multipath propagation due to the addition of obstacles, or a surge in external interference. Additionally, the inherent dynamic variability and unpredictability of wireless channels also make real-time and accurate channel quality predictions face huge challenges.

Most of the previous research in the field of channel prediction has focused on constructing channel models by designing Markov models [8]. Although this simple model-based approach has specific advantages when computing power and data are limited, their studies generally assume a variety of ideal conditions. And these assumptions are usually difficult to put into actual systems, which leads that such methods are unable to work well in different real environments. In today’s data-driven Internet-of-Things era, channel prediction algorithms based on classic machine learning have been applied to the channel prediction of backscatter communications to estimate the channel parameters when the tag is in different states, such as the eigenvalue decomposition (EVD) based on the received signal covariance matrix [9], the expectation maximization (EM) algorithm [10], and the least squares (LS) algorithm [11].

In addition, with the rapid development of cloud computing and big data technologies over the past decade [12], coupled with the continuous growth of the field of machine learning, researchers have begun to try to apply deep learning - machine learning which consumes more computing power - to channel prediction, so as to obtain more accurate predictions of channel metrics [13,14]. This provides us with an opportunity to design a more robust and accurate channel prediction system. Specifically, we introduce sequence-to-sequence encoder-decoder models based on deep learning, which are very suitable for the problem of mapping input sequences to output sequences [15]. The model has been widely used in tasks such as text generation and natural language translation. Moreover, the applicability of the encoder-decoder model in predicting channel metrics is also proved in [16].

In this paper, an encoder-decoder based sequence-to-sequence deep learning model (EDChannel) is designed for channel prediction in backscattered communication networks with convolutional neural networks (CNN) + long short-term memory (LSTM) as the encoder and LSTM as the decoder. We next design a channel detection scheme that has no requirement in hardware dependence based on the sequence-to-sequence prediction mode. The root mean square error (RMSE) of the real-time feedback sequence of the current channel indicator and the predicted value sequence obtained by the prediction module is used to determine whether to perform channel detection. Finally, we calculate a channel coefficient to select the channel with the best channel quality. Experimental results show that EDChannel algorithm achieves higher accuracy of channel prediction than Zhao’s model [6], ARIMA and autoregression(AR), and the overall throughput in stable environment and unstable environment is also increased by about 2.9% and 14.1% compared with Zhao’s scheme [6].

The major contributions of our work are summarized as follows:

- We propose an encoder-decoder based sequence-to-sequence deep learning model (EDChannel) for channel prediction in backscatter communication networks, which has the ability to predict future data sequences based on historical data sequences.
- We design a channel detection trigger based on the EDChannel’s sequence-to-sequence prediction mode, which is hardware-independent and can cope with channel bursts to a certain extent.
- We calculate a channel coefficient consisting of a normalized mean and variance according to the prediction results of EDChannel and present a channel selection method based on the channel coefficient. Our experimental results show that EDChannel has a superior prediction accuracy compared with state-of-the-art algorithms. The overall throughput of our solution improved by approximately 2.9% and 14.1% over Zhao’s scheme in stable and unstable environments respectively.

In the rest of this paper, Sect. 2 shows the related work. Section 3 gives the detailed design of the system. Section
4 describes the specific implementation details and data collection and evaluates the implemented methodology. Finally, Sect. 5 gives conclusions and describes our future work.

2 Related work

This section firstly introduces the related work of wireless channel prediction. Wireless channel prediction can be roughly classified into two categories: One is the Markov model, which models the channel changes by making simplified assumptions, relying on few network parameters and limited historical data; the other is a machine learning model, which is a channel prediction model used to predict the quality of wireless channels through historical data and machine learning algorithms [11]. We then introduce the related work of channel prediction based on deep learning in the backscatter communication network, and briefly describe the research status of channel detection and channel selection in the backscatter communication network. Finally, aiming at the shortcomings of the existing methods, we design our framework.

Channel prediction based on Markov model Smith et al. propose a linear finite state Markov predictor for channel prediction, and achieve a good balance between complexity and accuracy [8]. Yuan et al. improve modeling accuracy by selecting the optimal Markov model in different scenarios [17]. Thus, the practicality of the Markov model matrix in wireless channel modeling is enhanced. Leveraging a Markov model, Liu et al. develop a deep CNN-based framework to efficiently encode channel state information feedback to improve accuracy and efficiency [18]. While Markov models offer insight into the variability of wireless channels, previous work by Wang et al. suggest that higher order Markov models are necessary to obtain better performance [19]. In addition, Markov models usually assume that the transmitter knows all the channel state information and the receiver has error-free channel estimation, and these assumptions are usually difficult to meet in the actual channel prediction system.

Channel prediction based on machine learning In the channel prediction of backscatter communication networks, previous work mainly focused on the use of machine learning for channel parameter estimation. For example, Wang’s team use blind channel estimators based on EVD, EM algorithms, and LS algorithms to estimate the channel parameters when tags are in different states. And through the Cram’er-Rao Lower Bounds lower bounds to evaluate the performance of the algorithm [9–11,11,20]. Furthermore, Zhao et al. propose a prediction framework that uses a back propagation (BP) neural network-based prediction algorithm to predict the channel quality at the next moment [6]. Hou et al. largely improve the predictive performance of partial time series data by applying an AR based predictor [21]. Wu et al. use ARIMA to predict the wireless channel based on time-series analysis [22].

Channel prediction based on deep learning Deep learning has been widely used in the Internet of Things in recent years [23], but its application to the task of channel prediction in backscattered communication networks has not yet begun. The effectiveness of deep learning has been demonstrated in channel prediction tasks such as LTE, 5G [24], Wi-Fi [25] and multiple input multiple output (MIMO) [26]. Shehzad et al. propose a recurrent neural network (RNN) based massive MIMO channel predictor, which can work on the estimated channel and the compressed version of the estimated channel as well [27]. Similarly, Son et al. use the LSTM networks to improve the channel prediction accuracy of the vehicular communication system [28]. Adita et al. develop a LSTM variant and a gated cyclic unit (GRU) variant based on the internal unit structure used in the encoder and decoder [16]. They can predict future changes in wireless signal strength based on past signal strength data.

Channel detection and channel selection Channel detection in backscatter communication networks is typically accomplished by one or more probes [29]. Triggering methods for channel detection include the use of triggers to detect changes in the position or movement pattern of sensor tags [30], or the use of acceleration sensor to monitor the motion of the object itself [6]. These two methods can trigger channel detection in most cases promptly. However, in the experiments in Sect. 3.3, we found that the received signal strength indicator continued to change significantly when only the obstacle moved and the tag did not make a move. We still need to perform channel detection on this situation to get real-time channel conditions. In addition, these two methods require related hardware such as sensors in order to be implemented.

In the backscatter communication network, the correct channel selection can improve the efficiency of information transmission and reduce the loss of information. In the existing channel selection research, Blink and CARA select directly through the channel metrics detected [29,30]. Although this is convenient, it faces a certain degree of time delay. Zhao et al. selects channels based on predicted channel metrics [6]. Although the predicted value of the channel indicator can be obtained by single-step prediction of BP neural network, the predicted sequence of the channel indicator is short and the amount of information about the channel is small.

Our approach In this article, we design EDChannel for sequence-to-sequence channel prediction. In addition, we consider a hardware-independent channel detection trigger for channel prediction in backscatter communication...
Table 1 The differences between current work and previous work

| Module          | System                                                                 | Previous work                                                                 | Current work                                                                 |
|-----------------|------------------------------------------------------------------------|--------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| Channel prediction | LTE, 5G [24], MIMO [26], and WI-FI [25] etc.                          | AR [21], ARIMA [22], RNN [27], LSTM [28], LSTM or GRU based encoder-decoder etc.| EDChannel with CNN +LSTM as the encoder and LSTM as the decoder.                  |
| Backscatter communication | EVD [9], EM [10], LS [11] and Zhao’s model [6] etc.                | Use of related hardware such as sensors as the trigger of the channel detection [6,30]. | Use the RMSE of the measured value sequence and the predicted value sequence as the trigger of the channel detection. |
| Channel detection | Channel selection by means of detected channel metrics [29,30], or predicted channel metrics based on BP neural network [6]. | Channel selection is based on the sequence of predicted values from the EDChannel. |

networks, and introduce a channel coefficient for channel selection. The accuracy and wide applicability of the EDChannel algorithm is demonstrated through the results of channel prediction of the EDChannel algorithm and the comparison algorithm in a variety of real-world situations. The throughput of our solution is higher than the previous state-of-the-art methods.

The differences between current work and previous work are shown in Table 1. We observe that most of the channel prediction methods for backscatter communication use traditional machine learning [6,9–11], which does not work well in complex real-world environments. We improve the encoder-decoder which performs better in wireless communication and design the EDChannel with CNN +LSTM as the encoder and LSTM as the decoder. In addition, considering the problem that existing channel detection triggers do not cover all scenarios and are also dependent on hardware [6,30], we propose a lightweight and hardware-independent channel detection trigger based on RMSE values. Finally, we calculate channel coefficients for channel selection based on a sequence of predicted values, which takes more information into account than the results of Zhao’s single-step-based prediction [6].

3 System design

3.1 System overview

In order to improve the performance of the backscatter communication system and the accuracy of channel selection, we propose a more accurate channel prediction framework and build a sequence-to-sequence deep learning model based on encoder-decoder, as shown in Fig. 1. Among them, the EDChannel encoder uses the CNN model for feature extraction of sub-sequences, and the LSTM model to help extract features across time steps. The decoder uses the LSTM model to be responsible for reading and interpreting the input sequence model to achieve sequence-to-sequence channel prediction. On the basis of channel prediction, we design the channel detection module and channel selection module. The channel detection module uses the RMSE of the measured value sequence of the channel indicator and the predicted value sequence as the trigger of the channel detection. The channel selection module uses the mean and variance of the normalized sequence of predicted values to determine the channel selection criteria and to measure the signal strength and stability of the channel respectively.
3.2 Channel prediction module

We design EDChannel, as shown in Fig. 2. The encoder is responsible for reading and interpreting the input sequence model, it receives the past signal strength measurement value sequence $X = (X_1, X_2, \cdots, X_n)$, and generates a fixed-length context vector. It compresses and encodes the information of the entire input sequence $X$, and represents the model’s interpretation of the channel strength measurement value sequence. The decoder is a model in charge of explaining each step in the generated output sequence. It uses the context vector output by the encoder and the output $Y_{t-1}$ of the previous time step as input, and then outputs the signal strength prediction value sequence $Y_{t-1} = (Y_1, Y_2, \cdots, Y_k)$.

We use LSTM units as the underlying neural network architecture in each layer of the encoder and decoder. As shown in Fig. 3, LSTM is a special RNN, which has the same structure of neural network repeating module chain like RNN, but the internal structure of repeating modules is different. Compared with the simple layer of RNN, LSTM has four layers, which interact especially [32,33].

LSTM solves the problem of vanishing or exploding gradients by adding gates for controlling access to past information and has been proved effective in many prediction tasks, such as wind speed prediction. Encoders and decoders retain important features through various gate functions in LSTM to ensure that important features will not be lost during long-term transmission. There are three types of gates: input gate $i_t$, forget gate $f_t$ and output gate $o_t$. The calculation formula is as follows:

$$
\begin{align*}
    f_t &= \sigma(W_f [h_{t-1}, x_t] + b_f) \\
    i_t &= \sigma(W_i [h_{t-1}, x_t] + b_i) \\
    \tilde{C}_t &= \tanh(W_c [h_{t-1}, x_t] + b_c)
\end{align*}
$$

where $W_f, W_i$ and $W_c$ represent the corresponding weights. $b_f, b_i$ and $b_c$ represent the corresponding bias terms. $[h_{t-1}, x_t]$ represents the connection of two vectors into a longer vector, and $\sigma$ is the sigmoid function. The forget gate determines how much of the cell state $C_{t-1}$ from the previous moment is retained to the current moment $C_t$. The input gate determines how much of the network’s input $x_t$ at the current moment is saved in the unit state $C_t$. Then update the old unit state $C_{t-1}$ to the new unit state $C_t$. Finally, calculate the output gate $o_t$ and the unit output $h_t$ of the LSTM. The calculation formula is as follows:

$$
\begin{align*}
    f_t &= \sigma(W_f [h_{t-1}, x_t] + b_f) \\
    i_t &= \sigma(W_i [h_{t-1}, x_t] + b_i) \\
    \tilde{C}_t &= \tanh(W_c [h_{t-1}, x_t] + b_c)
\end{align*}
$$
Although LSTM has been proved in dealing with the time dependence of channel prediction, maintaining structural locality and solving such expansion problems remain to be challenged [34]. Therefore, in order to extract features completely, the CNN+LSTM discussed by Trigeorgis and Ringeval [35] is considered as an encoder in this paper. This model has been applied to traffic speed prediction and energy consumption prediction.

Figure 4 shows the overall architecture of a CNN. Among them, each node of the convolution layer extracts features from the input sequence through convolution operation, and generates a feature map. The pooling layer compresses the input feature map, on the hand, it makes the feature map smaller to simplify the network calculation complexity. On the other hand, feature compression is performed to extract the main features. The flattening layer mainly converts the three-dimensional layer in the network into a one-dimensional vector to fit the input of LSTM for long-term learning.

3.3 Channel detection module

In the study of channel detection, we found that with only the obstacle moving and the tag not moving, the indicator of the channel received signal strength still changed significantly, as shown in Fig. 5. In this case, we still need to perform channel detection to get real-time channel conditions. At the same time, considering the design of our channel prediction module, we use the RMSE of the target tag predicted value sequence and the true value sequence as the trigger for channel detection. From a large number of preliminary experiments, we found that the RMSE value of the predicted value and the true value of the entire sequence is generally less than 5 when the signal fluctuation is predictable. So in order not to miss any better channels, we set the threshold at 4. This means that we do not perform channel detection when the RMSE value of the whole sequence prediction and the true value is less than 4. When the RMSE value is greater than 4, a new channel detection scheme is executed.

3.4 Channel selection module

Following the channel detection and prediction module, a sequence of channel indicator predictions is obtained, using the mean $\overline{X}$ and variance $S^2$ to measure the average received signal strength and stability of the predicted sequence, respectively. The specific calculation formula is as follows:

$$
\begin{align*}
\overline{X} &= \frac{\sum_{i=1}^{n} X_i}{n} \\
S^2 &= \frac{\sum_{i=1}^{n} (X_i - \overline{X})^2}{n}
\end{align*}
$$

where $X_i$ is each value of the prediction sequence, and $n$ is the number of data in the prediction sequence. In channel selection, we always prefer the mean value to be as large as possible so that the received channel strength will be greater. At the same time, we also hope that the variance is as small as possible, because the channel will be more stable in this way. However, in the course of our experiments, we found that when we chose the channel with the highest mean value, the variance of that channel was not always the smallest. This means that the channel chosen is not always the strongest and most stable at the same time. We therefore introduce a channel coefficient $\gamma$ to indicate the channel’s strength. The calculation formula of $\gamma$ is as follows:

$$
\begin{align*}
\alpha &= \frac{\overline{X}_j - \overline{X}_{\min}}{\overline{X}_{\max} - \overline{X}_{\min}} \\
\beta &= \frac{S^2_j - S^2_{\min}}{S^2_{\max} - S^2_{\min}} \\
\gamma &= A\alpha + B\beta
\end{align*}
$$

where $\overline{X}_j$ and $S^2_j$ are the mean and variance of the corresponding sequence, $\overline{X}_{\max}$ and $\overline{X}_{\min}$ are the maximum and minimum mean values in all the sequences, $S^2_{\max}$ and $S^2_{\min}$ are the maximum and minimum variances in all the sequences.
is the positive normalization of the mean of the corresponding sequence, $\beta$ is the inverse normalization of the variance of the corresponding sequence, and $A$ and $B$ are the impact factors. We think that $\alpha$ and $\beta$ are equally important, so we set 0.5 for both. In the channel selection process, we always choose the channel with the maximum channel coefficient $\gamma$. When multiple channels have the same $\gamma$ value, the channel with the largest $\alpha$ value is selected. And when two channels have exactly the same $\alpha$ and $\beta$, the channel with a frequency close to that of the channel currently being communicated is selected.

4 Implementation and evaluation

In this section, we use RFID to conduct a lot of experiments to evaluate the performance of our scheme. First, we introduced the experimental equipment, the data collection environment and three evaluation metrics. Then the optimal training data window and optimal step size of the EDChannel algorithm under unstable environment are discussed, and various initialization parameters of the experiment are given. Next, we carried out the prediction experiment of the EDChannel algorithm in stable environment, and then discussed the experimental results of the car moving scene and the pedestrian moving scene. In order to make the evaluation more comprehensive, we used two additional metrics to analyze the experimental results, which proved the superior performance and wide applicability of the EDChannel algorithm. Finally, our experiments comparing the two channel selection methods show that channel selection using the channel coefficients of the prediction sequence can improve the throughput of the network.

4.1 Experimental equipment and data collection

Figure 6 shows that Impinj Speedway R420 reader and two different tags as backscatter nodes in the network. The specific equipment parameters are shown in Table 2. Channel data was collected in a variety of environments as shown in Fig. 7 and a deep learning module based on Tensorflow 2.4.0 and Keras 2.4.3 was used to implement our model.

We collect training data in both stable and unstable environments. Stable environment refers to the situation where the position of the node remains unchanged and there is no interference in the surrounding environment. An unstable environment refers to a situation where a node moves or is blocked or there is other signal interference around it. In the stable environment, we consider different distances and tag types. As shown in Fig. 7a, we arranged 7 interference tags and 1 target tag, and collected the RSSI value of the target tag channel for 20 minutes. In unstable environment, we considered the pedestrian moving scene and the car moving scene, as in Fig. 7b, c, respectively. We attached the target tags to the wrist of the pedestrian and the side of the car, and then collected the RSSI values of the channels for 20 minutes in environment with 6 and 8 interfering tags, respectively. In the experiment, the pedestrian walks back and forth between the antenna and the table. The car moves along the table at a speed of approximately 0.25m/s back and forth. It is a challenge for our prediction task because of the huge variation in data in unstable environments. Therefore we collect more data in unstable environments than in stable ones.

4.2 Evaluation index

The main metrics used for evaluation are RMSE, Mean Absolute Error (MAE) and Relative Error (RE). RMSE and MAE capture the error in the absolute prediction, and RE captures the ratio of the error in the prediction to the actual channel change. The specific calculation formula is as follows:

| Parameters | Values |
|------------|--------|
| Versions   | Keras versions = 2.4.3 |
|            | Tensorflow versions = 2.4.0 |
| Reader     | Model = Speedway R420 |
|            | Operating Region = China 920-925MHz |
|            | Antennas = 1 |
|            | Tx power = 30dBm |
|            | Rx sensitivity = -70dBm |
| Tags       | Common RFID electronic tags |
|            | WISP tags |

Table 2 Device parameters
Fig. 7 Data collection environment display. a Stable environment. b Pedestrian moving scene. c Car moving scene

\[ RMSE_j = \sqrt{\frac{\sum_{i=1}^{m} (\hat{y}_{ij} - y_{ij})^2}{m}} \]
\[ MAE_j = \frac{\sum_{i=1}^{m} |\hat{y}_{ij} - y_{ij}|}{m} \]
\[ RE_j = \frac{\sum_{i=1}^{m} |\hat{y}_{ij} - y_{ij}|}{y_{ij} m} \]

where \( \hat{y}_{ij} \) and \( y_{ij} \) are the predicted value and the true value of the \( i \) test sample of the \( j \) prediction step, and \( m \) is the number of test samples. In most papers related to channel prediction, the researchers mainly use RMSE as an evaluation method [16]. For consistency, we also use RMSE as the main evaluation metric for this paper, supplemented by MAE and RE.

4.3 The effect of prediction step size and training data length

In the following, we discuss the effect of the prediction step size and training data length of the EDChannel algorithm on the prediction accuracy.

4.3.1 The effect of prediction step size

We use a controlling variable approach to discuss the effect of the prediction step size of the EDChannel algorithm on the prediction accuracy. We first fixed the length of the training data to 500 and the number of iterations to 500–5000, then we varied only the prediction step size. The range of prediction step size was 1-30, each step size was added 1 compared to the previous one, and the experiment was repeated 10 times for each same step size. At the end, the differences in RMSE values at different step sizes were evaluated together.

Figure 8a shows the variation of the received signal strength in a real scenes. We observe that the data fluctuates only slightly in the stable environment, more steadily in the car movement scene, while the data fluctuates more dramatically in the pedestrian movement scene. Figure 8b shows the variation of RMSE values from 1 to 30 prediction steps for each of the three scenes. We observe that the value of RMSE increases with increasing prediction step for all three scenes. This is because the uncertainty of the channel state increases as the number of prediction steps increases. Furthermore, in stable environment, the RMSE values fluctuate.
4.3.2 The effect of training data length

We still use the controlling variable approach, fix the prediction step size to 10, and the number of iterations to 500–5000. Then, we varied the training data length only. The range of training data is 10-660, each training data length is added 10 compared to the previous one, and the experiment is repeated 10 times for each same training data length.

Figure 9a shows the variation of the received signal strength in a real scene. We observe that the data fluctuates only slightly in the stable environment, more steadily in the car movement scene, while the data fluctuates more dramatically in the pedestrian movement scene. Figure 9b shows the variation of RMSE values from 10 to 660 for the training data length in the three scenes, respectively. We observe that the RMSE values in the three scenes decreased as the training data increased and the predictions became progressively better. However, the prediction is poorer when there is smaller training data. We analyze the reasons for this are that the training data does not meet the generalization requirements of the EDChannel model, and that the types of fluctuations in the training data are smaller than the types of fluctuations in the predicted data, so that we cannot fully learn the fluctuations in the data. In stable environment, the RMSE value floats around 0.4 when the training data length is greater than 100. In the car moving scene, the RMSE value fluctuates around 4 when the training data length is greater than 170. In the pedestrian movement scene, the RMSE value varies around 5.8 when the training data length is greater than 250.

4.3.3 Summary

Figures 8a and 9a shows the received signal strength variation in a real scene. We observe that the data fluctuates only slightly in the stable environment, more steadily in the car movement scene, while the data fluctuates more dramatically in the pedestrian movement scene. From Fig. 8b, we found that setting the prediction step size at 10 is more suitable for all three scenes, and the computational overhead is also small. However, it is worth noting that the longest stay time of a single channel of the backscatter communication network varies according to regional regulations. For example, in North America, the FCC allows a single channel to
Table 4 Results of RE and MAE of EDChannel, Zhao’s model, ARIMA, and AR in different environments

| Environment            | RE       | MAE       |
|------------------------|----------|-----------|
|                        | EDChannel| Zhao’s    | ARIMA  | AR     | EDChannel| Zhao’s    | ARIMA  | AR     |
| Stable environment     | 0.576    | 0.849     | 0.511   | 0.574  | 0.287    | 0.423     | 0.286   | 0.255  |
| Car moving scene       | 5.755    | 8.381     | 7.622   | 8.301  | 2.567    | 3.738     | 3.702   | 3.399  |
| Pedestrian moving scene| 7.790    | 11.526    | 10.756  | 11.230 | 3.616    | 5.349     | 5.212   | 4.992  |

Bold values indicate the results of this work.

Fig. 10 Comparison of the true value and the predicted value in stable environment. a EDChannel and Zhao’s model. b EDChannel and ARIMA. c EDChannel and AR.

reach 0.4 seconds in 10 seconds. In China, the maximum stay time is 2 seconds [6]. Therefore, we set the prediction time to 2 seconds, which means that the prediction is re-predicted every 2 seconds. At the same time, we can predict the data changes in the next 2 seconds through multi-step forecasting. From Fig. 9b, we found that the training data length is between 250 and 600, and the fluctuation of the RMSE value is relatively stable, which is more suitable for all three scenes. But from Figs. 8a and 9a, we found that in unstable environment, the channel quality changes quickly. So in order to collect more training data, we set the training data length between 300 and 500. In this way, historical data can be fully learned to meet the prediction of future data, and the computational overhead can be controlled within a reasonable range to meet the high-speed iteration of the model. In the actual data collection process, our equipment can collect approximately 39 data points per second. Therefore, to predict the data for the next 2 seconds when the step length is 10, we need to predict about 8 steps. The training data length of 300 to 500 requires us to continuously collect about 7.5s to 13s for a channel.

4.4 Initialization parameters

When using new data to execute the EDChannel algorithm, we need to repeatedly estimate the parameters, which is very low in computational efficiency. For the sequence-to-sequence prediction model of channel metrics for backscattered communication networks, we give the initialisation parameters for the EDChannel model, as shown in Table 4. Hence we can improve the computational efficiency by initialising the parameter search.

As mentioned earlier (Sect. 3.2), in EDChannel, both the encoder and decoder use deep RNNs. The encoder uses CNN+LSTM as the basic unit, and the decoder only uses LSTM as the basic unit. We observe that in order to provide the best performance, the optimal parameter configuration of EDChannel varies with the consideration of the data set. An encoder or decoder is composed of 1 or 2 layers of basic unit stacks, and each layer has 20 to 100 hidden units. The 1D CNN layer uses 4, 8, or 16 filters, 1 convolution kernel, 1 max-pooling layer, and the activation function is ReLU. The number of neurons in the LSTM layer stacked with the CNN
4.5 Channel prediction evaluation in stable environment

We will compare the performance of EDChannel with Zhao’s model, ARIMA, and AR in stable environment. The Zhao’s model, ARIMA, and AR algorithms also consider the history of 300–500 samples in the past to predict the future 10 data and predict the data values of 10 time steps. However, as the ARIMA and AR models have poor long-term forecasting, we update the model immediately after every 1 time step of forecasting.

The RSSI results of the real value and the predicted value of EDChannel, Linear, ARIMA, and AR are shown in Fig. 10. We observe that the RSSI value in stable environment tends to be stable for a long time, fluctuating between (−50, −49).

And the predictions of EDChannel, Zhao’s model, ARIMA and AR also fluctuate between (−50, −49) without much deviation. From Fig. 10a, we observe that the prediction effect of EDChannel and Zhao’s model is similar to the actual situation and can truly reflect the change of RSSI value. From Fig. 10b, c, we observe that the predictions of ARIMA and AR are concentrated around −49.8 and above, converging to the mean of the true values. The predictions are poor. Figure 11 illustrates the RMSE values for the four algorithms in stable environment, we observe that the RMSE value between the predicted value and the true value of EDChannel is always less than 0.45. Compared with Sect. 4.6, the prediction effect under the unstable environment is excellent. Also, as time increases EDChannel’s predictions are better compared to other algorithms. And with 10 time steps, only step 5 is slightly worse than the AR. The experimental results show that EDChannel works better than other methods in the stable environment and is able to meet the task of channel prediction.

4.6 Channel prediction evaluation in unstable environment

In this section, we will compare the performance of EDChannel with Zhao’s model, ARIMA, and AR in unstable environments. The parameter settings of Zhao’s model, ARIMA, and AR algorithms are the same as in Sect. 4.5.

4.6.1 Channel prediction for car moving scene

Figure 12 shows the comparison between the predicted value of RSSI and the true value in car moving scene. We observe that the predictions of all four algorithms are more in line with the real situation. But EDChannel’s prediction is better when the channel metrics are more variable, such as between (40, 70). From Fig. 13, we observe that EDChannel’s RMSE values are smaller than those of the comparison algorithm at most time steps, only slightly larger than Zhao’s model at step
Fig. 14  Comparison of real and predicted values in the pedestrian moving scene. a EDChannel and Zhao’s model. b EDChannel and ARIMA. c EDChannel and AR

Fig. 15  The RMSE of the predicted value and the true value of the channel data in the pedestrian moving scene

4.6.2 Channel prediction for pedestrian moving scene

Figure 14 shows the comparison between the predicted value of RSSI and the true value in the pedestrian moving scene. We observe that EDChannel and Zhao’s model have better predictions, while ARIMA and AR have worse predictions. From Fig. 15, we observe that EDChannel predicts only slightly better than the comparison algorithms in the first 5 time steps, and significantly better than the comparison algorithms in the last 5 time steps. On the one hand, it is because the data in the first 5 time steps vary less compared to the last 5 time steps and the RMSE has a smaller range of variation. On the other hand, it is because EDChannel learns better than the comparison algorithms about abrupt changes in channel metrics in historical data.

4.6.3 Summary

In summary, from Figs. 12 and Fig. 14, we observe that the prediction results of EDChannel are more in line with the real situation, which indicates that the prediction function of EDChannel is effective in the car moving scene and the pedestrian moving scene. Furthermore, we observe that the EDChannel predicts better than the comparison algorithms when the channel metrics are more variable, which is also verified in Figs. 13 and 15. The experimental results show that EDChannel works better compared to other methods in unstable environments and is able to meet the task of channel prediction.

4.7 Results of RE and MAE

The following we use two other evaluation metrics, RE and MAE, for additional illustration of the experiment. Table 4 shows the RE and MAE results for EDChannel, Zhao’s model, ARIMA, and AR for the predicted and real data in different environments. In the stable environment, we observe that the RE and MAE values of the EDChannel are slightly larger than ARIMA and AR. The main reason is that the predictions of ARIMA and AR in the stable environment are concentrated around $-49.8$, converging to the mean of the true values, as shown in Fig. 10b, c. So, it doesn’t mean that EDChannel is poor at predicting in the stable environment. In addition, the RE and MAE values of EDChannel are significantly smaller than Zhao’s model, demonstrating the accuracy of EDChannel’s predictions in the stable environment. In the car moving scene and the pedestrian moving scene, we observe that the performance of EDChannel is higher than the comparison algorithms when evaluated on both RE and MAE metrics. And EDChannel is at least 24% better than the comparison algorithms. In summary, the experiments have proved the accuracy and applicability of EDChannel for RSSI prediction in different environments.
4.8 Channel selection through channel prediction

In the following we evaluate the effectiveness of our prediction algorithm through simulation experiments and five passive tags. In order to obtain experimental data, we use the reader to read the data of each channel in turn under the same circumstances, and collect the RSSI value of each channel. In stable environment, it is easier to keep the environment unchanged and collect 16 channels of data. Under unstable conditions, however, it is more difficult to ensure that the tag is in the same state of motion for each experiment. To ensure that the conditions are consistent from one collection to the next, we attach the labels to the pendulum and release them from the same height each time. We consider that the errors under these experimental conditions do not affect the experimental results. Then, we model and predict each channel separately, and get the prediction sequence of each channel for a period of time. Finally, we select the channel according to the channel coefficient $\gamma$ of each channel prediction sequence. And compare our method with Zhao’s model.

In Fig. 16, channel 11 and channel 7 have the largest channel coefficients in stable and unstable environments, so we choose channel 11 and channel 7 respectively as the communication channel for the next time period. Figure 17 shows the throughput of our method and Zhao’s scheme communicating for 4 seconds in both environments. In stable environments, we found that our method improves only about 2.9% compared to Zhao’s scheme, with a smaller improvement in throughput. We attribute it to the fact that each channel in stable environment is better, and the channel selection does not show enough superiority. In unstable environments, our method has a more significant improvement in throughput compared to Zhao’s scheme, increasing by about 14.1%. On the one hand, because of the channel changing drastically in unstable environment, our method can maintain a better channel for communication. On the other hand, it is because Zhao’s scheme only considers the predicted value of the channel metric at one future step, and the accuracy of the prediction results is lower than EDChannel. The experimental results show that the method of channel selection through channel prediction can effectively increase the network throughput, and our method works better than Zhao’s scheme.

5 Conclusion

In this paper, we investigated the channel prediction problem in backscatter communication networks. We developed EDChannel, an encoder-decoder based sequence-to-sequence deep learning model that learns from previous data sequences to predict future data sequences. In addition, considering the problem that existing channel detection triggers do not cover all scenarios and are also dependent on hardware, we propose a lightweight channel detection trigger based on RMSE values. Finally, we introduce a chan-
nel coefficient for channel selection. Experiments show that EDChannel outperforms Zhao’s model, ARIMA, and AR in predicting a variety of scenarios. Compared to Zhao’s scheme, our prediction scheme improves network throughput by about 2.9% and 14.1% in stable and unstable environments respectively. In the future, we plan to deploy this algorithm on a real backscatter communication network to solve the channel sensing problem. Also, we attempt to improve prediction accuracy in other scenarios and study the application of channel prediction in rate adaptation.

Funding This paper supported by The General Object of National Natural Science Foundation under Grants(61972273); National Major Scientific Research Instrument Development Project (6202780085); National Natural Science Foundation of China(62102280); Fundamental Research Program of Shanxi Province(20210302124167); National Key R&D Project(2018YFB2200900); Shanxi Province key technology and generic technology R&D project(2020XXX007).

Declarations

Conflict of interest All authors have declare that they have no conflict of interest.

Consent for publication Yes

References

1. Mitsugi, J., Kawakita, Y., Egawa, K., & Ichikawa, H. (2019). Perfectly synchronized streaming from multiple digitally modulated backscatter sensor tags. IEEE Journal of Radio Frequency Identification, 3(3), 149–156.
2. Ma, L., Hu, K., Zhu, Y., & Chen, H. (2014). Cooperative artificial bee colony algorithm for multi-objective rfid network planning. Journal of Network and Computer Applications, 42, 143–162.
3. Wang, Q., Zhao, J., Xu, S., & Wang, R. (2021). Retype: Your breath tells your mind! IEEE Internet of Things Journal, 8(11), 9436–9449.
4. Miao, Q., Xiao, F., Huang, H., Sun, L., & Wang, R. (2020). Smart attendance system based on frequency distribution algorithm with passive rfid tags. Tsinghua Science and Technology, 25(2), 217–226.
5. Zhao, J., Li, J., Li, D., & Yang, H. (2020). Optimal data transmission in backscatter communication for passive sensing systems. Tsinghua Science and Technology, 25(5), 647–658.
6. Zhao, J., Tian, H., & Li, D.-A. (2020). Channel prediction based on hp neural network for backscatter communication networks. Sensors, 20(1), 1125.
7. Gong, W., Chen, S., Liu, J., & Wang, Z. (2018) Mobirate: Mobility-aware rate adaptation using phy information for backscatter networks. In IEEE INFOCOM 2018-IEEE conference on computer communications (pp. 1239–1267).
8. Zhang, J., Smith, D., & Chen, Z. (2012). Linear finite state Markov chain predictor for channel prediction. In IEEE 23rd international symposium on personal, Indoor and mobile radio communications (PIMRC) (pp. 2085–2089).
9. Zhao, W., Wang, G., Atapattu, S., & Ai, B. (2018). Blind channel estimation in ambient backscatter communication systems with multiple-antenna reader. In IEEE/CIC international conference on communications in China (ICCC) (pp. 320–324).
10. Ma, S., Zhu, Y., Wang, G., & He, R. (2018). Machine learning aided channel estimation for ambient backscatter communication systems. In IEEE international conference on communication systems (ICCS) (pp. 67–71).
11. Ma, S., Wang, G., Fan, R., & Tellambura, C. (2018). Blind channel estimation for ambient backscatter communication systems. IEEE Communications Letters, 22(6), 1296–1299.
12. Lv, Y., Peng, S., Yuan, Y., Wang, C., Yin, P., Liu, J., & Wang, C. (2019). A classifier using online bagging ensemble method for big data stream learning. Tsinghua Science and Technology, 24(4), 379–388.
13. Ma, L., Cheng, S., & Shi, Y. (2021). Enhancing learning efficiency of brain storm optimization via orthogonal learning design. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 51(11), 6723–6742.
14. Sun, W.-S., & Han, D. S. (2021). Analysis on the channel prediction accuracy of deep learning-based approach. In International conference on artificial intelligence in information and communication (ICAIC) (pp. 140–143).
15. Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). Doha, Qatar: Association for Computational Linguistics (pp. 1724–1734). https://aclanthology.org/D14-1179
16. Kulikarni, A., Seetharam, A., Ramesh, A., & Herath, J. D. (2020). Deepchannel: Wireless channel quality prediction using deep learning. IEEE Transactions on Vehicular Technology, 69(1), 443–456.
17. Yuan, L., Guo, T., He, Y., & Cheng, Z. (2018). The application of Markov model optimization method in wireless channel modeling. In 2018 10th international conference on communications, circuits and systems (ICCCAS). IEEE (pp. 315–319).
18. Liu, Z., Del Rosario, M., & Ding, Z. (2021). A Markovian model-driven deep learning framework for massive mimo csi feedback. IEEE Transactions on Wireless Communications., 2, 4478.
27. Shehzad, M. K., Rose, L., Wesemann, S., & Assaad, M. (2022). ML-based massive mimo channel prediction: Does it work on real-world data? *IEEE Wireless Communications Letters*, 69(1), 518–528.

28. Son, W.-S., & Han, D. S. (2021). Analysis on the channel prediction accuracy of deep learning-based approach. In *2021 international conference on artificial intelligence in information and communication (ICAIIC)*. IEEE (pp. 140–143).

29. Gong, W., Liu, H., Liu, K., Ma, Q., & Liu, Y. (2016). Exploiting channel diversity for rate adaptation in backscatter communication networks. In *IEEE INFOCOM 2016-the 35th annual IEEE international conference on computer communications* (pp. 1–9).

30. Zhang, P., Gummeson, J., Ganesan, D. (2012). Blink: A high throughput link layer for backscatter communication. In *Proceedings of the 10th international conference on mobile systems, applications, and services* (pp. 99–112), ser. MobiSys ’12. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2307636.2307646

31. Park, S. H., Kim, B., Kang, C. M., Chung, C. C., & Choi, J. W. (2018). Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture. In *IEEE intelligent vehicles symposium (IV)* (pp. 1672–1678).

32. Thapa, K. N. K., & Duraipandian, N. (2021). Malicious traffic classification using long short-term memory (Istmn) model. *Wireless Personal Communications*, 119(3), 2707–2724.

33. Koushik, C., & Vetrivelan, P. (2020). Heuristic relay-node selection in opportunistic network using rnn-lstm based mobility prediction. *Wireless Personal Communications*, 114(3), 2363–2388.

34. Li, S., Wang, Q., Liu, X., & Chen, J. (2018). Low cost lstm implementation based on Stochastic computing for channel state information prediction. In *IEEE Asia Pacific conference on circuits and systems (APCCAS)* (pp. 231–234).

35. Trigeorgis, G., Ringeval, F., Brueckner, R., Marchi, E., Nicolaoiu, M.A., Schuller, B., & Zafeiriou, S. (2016). Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 5200–5204).

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.
Jumin Zhao is a Professor and PhD Instructor at College of Information and Computer in Taiyuan University of Technology (TYUT), Shanxi, China. Her current research fields include signal and information processing and wireless sensor network. She received a PhD of Circuits and Systems College of Information Engineering from Taiyuan University of Technology. Email: zhaojumin@tyut.edu.cn