IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2020)

Predictive Interference Management for Wireless Channels in the Internet of Things

Ali Nikoukar*, Yash Shah*, Adel Memariani*, Mesut Güneş* Behnam Dezfouli†
*Communication and Networked Systems (ComSys) - Faculty of Computer Science
Otto-von-Guericke University, Magdeburg, Germany
†Internet of Things Research Lab, Department of Computer Engineering, Santa Clara University, USA
{ali.nikoukar, yash.shah, mesut.guenes}@ovgu.de, adel.memariani@st.ovgu.de, bdezfouli@scu.edu

***Abstract*—Wi-Fi and Bluetooth are two wireless technologies, available in every smart-phone, tablet, and laptop. Wi-Fi Access Points (APs) and Bluetooth beacons are deployed in most indoor environments to provide service for the Internet of Things (IoT) applications. Although, Bluetooth and Wi-Fi target different applications, they both share the 2.4 GHz frequency band. The re-transmissions caused by interference with Wi-Fi packets is costly for BLE in terms of energy consumption. Techniques such as Adaptive Frequency Hopping (AFH) in BLE addresses this problem. However, the static nature of AFH is not performing well for highly dynamic environments. Therefore, there is a need for a predictive model to optimize the spectrum usage. In this paper, we propose a machine learning model based on Long Short-Term Memory (LSTM) to predict the wireless activities in the 2.4 GHz frequency band and its impact on BLE channels. We apply the proposed model to analyze the Wi-Fi interference trend on these channels. The Root Mean Squared Error (RMSE) results for several experiments on both channels indicate the high performance of the proposed LSTM model over Auto Regressive Integrated Moving Average (ARIMA) model. This improvement is significant up to approximately 50% reduction in error.**

***Index Terms*—Long Short-Term Memory (LSTM), Bluetooth Low Energy (BLE), Machine Learning, Wi-Fi, Internet of Things (IoT)**

I. Introduction

Most wireless technologies employed by Internet of Things (IoT) devices operate independently and they are not aware of the traffic and spectrum occupancy of each other. Due to the global availability of 2.4 GHz band, many wireless technologies use this band, e.g., Wi-Fi (IEEE 802.11b/g/n), Bluetooth Low Energy (BLE), Bluetooth classic, ZigBee, cordless phones, and even non-networking devices such as Microwave ovens [1].

It is expected that the number of BLE devices in smart buildings rise from 95 million in 2018 to 360 million in 2022 [2]. Meanwhile, the collisions caused by spectrum scarcity leads to re-transmission delay, unreliability, and higher battery consumption [3]. Wi-Fi and BLE are two dominant technologies in indoor environments. We are surrounded by Wi-Fi Access Points (APs) and BLE beacons in most of the indoor environments. The concept of Cognitive Radio (CR) [4] has been defined to address the scalability issue by efficiently managing the allocation of the spectrum in time, frequency, and power domain. Operation of heterogeneous wireless technologies in the same frequency band with different channel assignment, spectrum access mechanism, and bandwidth raise challenges regarding scheduling the spectrum as a shared resource.

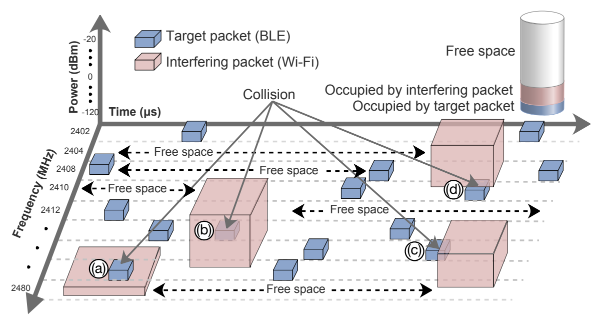

Fig. 1. 2.4 GHz spectrum shared by BLE and IEEE 802.11b/g/n

Figure 1 denotes the BLE and WiFi packets as target and interfering respectively. The figure demonstrates the spectrum schematic and shows how two packets sharing the same frequency and at the same time are colliding. Several types of collisions are possible: First, Figure 1(a) shows a BLE packet colliding with Wi-Fi packet. Because the interfering packet has lower transmit power, the target packet survives. In Figure 1(b), the target packet is completely lost. Although the domain of interfering packet in Figure 1(c) and (b) are damaging the target packet, the packet may be detectable by the receiver or using techniques such as Forward Error Correction (FEC). Additionally, in Figure 1, it is clear that most of the spectrum is unoccupied, and a collision occurs only because of poor resource management.

Machine learning-based time series prediction can be employed to improve spectrum management. This means BLE radio can improve Packet Reception Rate (PRR) by spectrum sensing and interference prediction on each channel. Wi-Fi interference prediction on BLE channel is critical for several reasons: i) It increases battery life and reduces the delay in transmission. ii) Indoor Positioning Systems (IPSs) based on BLE beacons can benefit from this prediction by considering the impact of noise on range estimation. iii) It can be used for an intelligent blacklisting scheme to improve Adaptive Frequency Hopping (AFH) performance in BLE data channels. iv) It helps researchers to develop a decision-making algorithm adaptively using *physical layer* to employ different

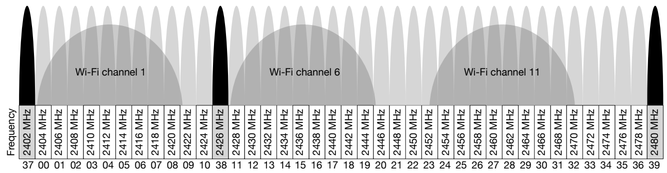

Fig. 2. 2.4 GHz ISM frequency band shared by Wi-Fi and BLE channels. The black channels are advertisement channels used by BLE beacons.

power levels, packet sizes, and modulation rates for each channel condition.

Since BLE has been primarily designed for devices with low processing power, sophisticated deep learning algorithms are not suitable on these devices. Besides, due to high dynamicity of wireless channels, achieving timely and accurate predictions is essential.

In this paper, we present a Light Weight Interference Prediction Model (LWIPM) based on Long Short-Term Memory (LSTM) [5] for resource constraint IoT devices to predict the Wi-Fi interference on BLE channels. To the best of our knowledge, there is no study focusing on Wi-Fi interference prediction for BLE channels.

This paper is organized as follows. In Section II we elaborate the background of the research on CR. The methodology of the research and the concept of LSTM is explained in Section III. Section IV discusses the outcome of the study and compares the obtained results. Finally, we conclude the paper in Section V.

II. Related work

Interference modeling and analysis have been studied in [6]–[9]. Furthermore, in [10] BLE signal propagation and impact of interference on advertisement channels have been analyzed. The authors conducted extensive experiments in several indoor and outdoor environments to develop a simulation model by relying on a realistic dataset. Similarly, authors in [11] studied interference modeling to develop a simulation platform. They captured Wi-Fi interference on IEEE 802.15.4 channels and proposed a model to use in ns-2 simulation. Although these studies help to have a better understanding to develop an interference model, they do not consider the effectiveness of using time series prediction algorithms. On the other hand, CR attempts to fulfill this gap by employing machine learning algorithms. The main idea is to sense the wireless spectrum and train the machine learning algorithm to predict the spectrum holes for successful data transmission. In this direction, several machine learning algorithms used for channel interference prediction have been discussed in [12]. In this study, the authors emphasize the advantage of using Recurrent Neural Networks (RNNs) over *Game Theory*, *Support Vector Machine*, and *Bayesian* non-parametric approaches, due to the high and continuous interference variation. In this direction [13], proposed a kernel-based reinforcement learning approach to predict the interference in the wireless channel to achieve the optimal back-off period in CSMA-CA. Additionally, due to the development of deep learning algorithms in recent years, there has been a growing interest in using machine learning and time series prediction methods to efficiently manage the wireless spectrum [14]. However, a channel prediction model for BLE is missing in the literature. In addition, because the application of machine learning in wireless channel modeling is a recent topic, there is insufficient study covering the prediction algorithms such as LSTM. This highlights the need for developing a lightweight, real-time, and precise interference level prediction algorithm.

III. Methodology

This section discusses the interference in 2.4 GHz band and its impact on BLE channels, as well as the detailed description of LWIPM to characterize and predict the interference on channel 38 and 15.

A. Interference in 2.4 GHz Frequency Band

The 2.4 GHz is a globally unlicensed band used by several technologies such as IEEE 802.11b/g/n and BLE. They employ different mechanisms to cope against interference. The objective of this study is the characterization and prediction of BLE in the presence of Wi-Fi interference. As can be seen in Figure 2, the advertisement channels are strategically placed to avoid overlapping with commonly used IEEE 802.11b/g/n channels i.e., channels 1, 6, and 11. Data channels, on the other hand, compete with Wi-Fi. Therefore, higher interference is expected in data channels compared to advertisement channels, and this makes BLE's data channels more vulnerable. BLE employs AFH and channel blacklisting in data channels to avoid transmission on highly affected channels. The blacklisting scheme excludes noisy channel from the hopping list. However, low accuracy in selecting the crowded channels reduces the efficiency of the network in terms of packet delivery ratio, delay, and energy consumption. By employing a robust prediction model and computing the probability of packet loss based on interference level, BLE can make more accurate decisions for tuning its transmission parameters such as power, frequency, and time.

We selected the BLE channel 38 and 15 as advertisement channel and data channel, respectively. It allows us to observe the performance of LWIPM under two utterly different interference levels. Our model is influenced by ON/OFF model [15] in CR which uses energy detection [16] for spectrum sensing. To this end, using nRF52840 and every $9\mu s$, we capture the Wi-Fi interference in a particular frequency without detecting the source Wi-Fi device. Then we develop a LWIPM based on the recorded dataset to forecast the Wi-Fi interference. The objective of LWIPM is to predict the interference level accurately valid for a more extended period and using minimum samples for training (save time and battery power).

B. Model design

It is crucial to find the optimal parameter for the developed machine learning model to avoid latency (because of over-training) and improve accuracy. For example, how many

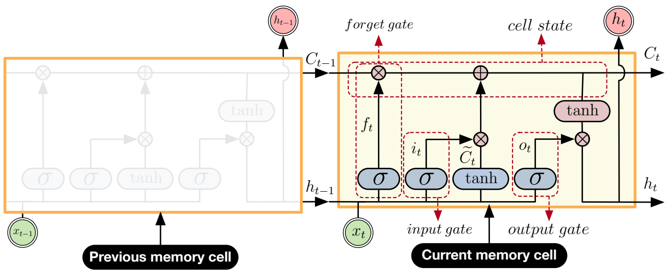

Fig. 3. Memory cell architecture for Long Short-Term Memory (LSTM).

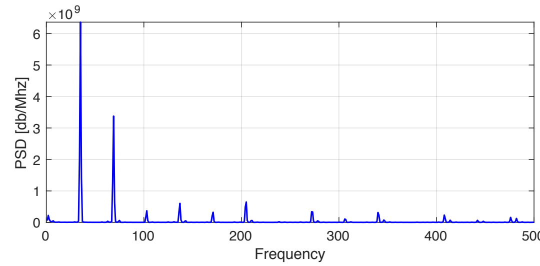

Fig. 4. Figure shows *Periodogram* computation to estimate power on channel 15 by *Welch* method where data is divided into L number of overlapping segments of length K. It is computed for each segment. Here x-axis shows frequency value and y-axis shows PSD [18] in unit Decibel (DB) per Megahertz (MHz).

samples are needed to achieve the desired accuracy and how long the prediction will remain valid. The main goal of this study is to develop an accurate and straightforward algorithm for resource-constrained IoT devices. Although, short training time will lower the model accuracy, over-training will cause delay in transmission. Besides, over-training has negative effect on prediction accuracy. Thus, the goal is to find the shortest possible training time with the highest prediction accuracy.

C. Long Short-Term Memory (LSTM)

The Wi-Fi interference behavior for the selected channels is highly variable. Even though Recurrent Neural Networks (RNNs) are one of the most common methods for learning from the sequential data and time series, they suffer from $vanishing\ gradient$ problem [17] where the gradient value decreases to the extent that it does not provide any significant contribution. Gradient values are used to update neural network weights during *backpropagation*. However, it is challenging for RNNs to remember the long sequences entirely. LSTM was created to address this issue of short-term memory. The key advantage of using LSTM is to model sequences with different lengths and capture their long-term dependencies for accurate predictions.

LSTM has similar chain-like structure as of RNNs. Figure 3 shows the block diagram of LSTM. It contains three layers: input, cell, and output. A cell is made up of three gates (input, forget, output) and a cell unit. A gate uses $sigmoid$ activation function whose output is a number between zero and one, describing how much of each component of the information should be let through, while cell state and input are converted using $tanh$. The updating procedure of a LSTM cell is executed in four steps.

Step 1: Calculate the value of current cell state C_in_t. It takes the previous output (h_{t-1}) and input data (x_t), and outputs a number between 0 and 1. W_{xc} and W_{hc} are the weights of input data and previous output, respectively.

$$C\_in_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$

Step 2: Decide what new information needs to be stored in the cell state (input gate) i_t and calculate its value.

$$i_t = \sigma_i(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$$

To tune the effect of previous information on current cell state, the value of forgotten gate f_t is calculated.

$$f_t = \sigma_f(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f)$$

Furthermore, the old cell state C_{t-1} is updated to the new cell state C_t where old state is multiplied by f_t.

$$c_t = f_t \odot c_{t-1} + i_t \odot C\_in_t$$

where $\odot$ shows point-to-point product.

Step 3: This step is for deciding on the output. A $sigmoid$ layer, which takes real valued input and gives output between 0 and 1, decides which parts of the cell state should give output by calculation of output gate o_t.

$$o_t = \sigma_o(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)$$

Step 4: To compute the new output h_t, tangent of newly calculated cell state $\tanh(c_t)$ is multiplied to $sigmoidal$ output gate o_t shown in step 3.

$$h_t = o_t \odot \tanh(c_t)$$

D. Data pre-processing

A time series is a sequence of observations taken at successive equally spaced points in time. In this regard, time series prediction involves fitting a model by previously observed data and predicting future observations. Therefore, a time series problem can be viewed as a supervised machine learning problem: the previous measurement is used as an input to predict the next outcome.

In order to use time series prediction models, it is necessary to remove temporal dependency structures like trend and seasonal fluctuations from the data i.e., convert any non-stationary series to a stationary series. The model which describes non-stationary data will have a variance in its accuracy at different time-steps (training a model on time series with trend and seasonal fluctuations are possible, but it leads to degraded performance). Since in this study, LWIPM is not stationary, as seen in Figure 6, first we need to convert the data to stationary series by using *difference transformation* technique. This technique for a given interval subtracts the

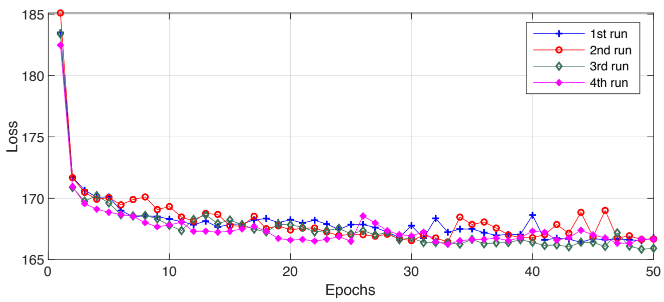

Fig. 5. The value of loss function during training the model on Channel 15 for 4 iterations.

value at the interval from the current time-stamps and the number of times the differencing should be performed is defined by difference order. Difference order of 1 is used to de-trend the time series. For removing seasonality, we use power spectrum estimates in the form of *periodograms* which are generated using *Welch* method. The result for channel 15 is shown in Figure 4 where the signal is converted from the time domain to the frequency domain. The difference order is selected based on the frequency, which shows high power spectral density.

E. Power Spectral Density

To visualise and interpret the seasonal component in the data, we visualize PSD of the series. PSD function is used to see the power of variations in the given data as a function of frequency. As seen in Equation (1),

$$\begin{aligned} F\{r(t)\} &= |X(\omega)|^2 \sim PSD \\ r(t) &= x(t) \times x(-t) \end{aligned} \tag{1}$$

PSD $|X(\omega)|^2$ is Discrete Fourier Transformation (DFT) and $F\{r(t)\}$ shows power distribution for the given time series or an auto-correlation sequence $r(t)$ as defined by Equation (1). One of the methods to visualize power spectrum is *Periodogram* [19]. For a given periodic signal, it takes samples from $X(\omega)$ for a consecutive length and obtains estimations by taking DFT to obtain its coefficient $\hat{x}(\omega)$. The *periodogram* is generated by using a quantity proportional to the square of the coefficient and divided by that length.

$$S_{xx}(\omega) = \lim_{L\to\infty} \frac{1}{L}\left[|\hat{x}(\omega)|^2\right] \tag{2}$$

As seen in Equation (2), when the length L approaches infinity for power spectral density S_{xx}, *periodogram's* disadvantage is highlighted. The variance at a particular frequency does not decrease with the increment in the number of samples used in the computation. It becomes almost equal to the square of the power spectrum at a given frequency, and thus, it is an inconsistent estimator. Here, *Welch* method is used as an improvement over *Bartlett's* method [20] for reducing the variance in the *periodogram*.

TABLE I
MODEL PARAMETERS

Parameter	Value
Epochs	20
Batch size	80
Input dimension	5
Hidden layer nodes	20
Difference order	30

1) Model Description: A well-known approach to tune the hyperparameters of a machine learning model is *grid search*. It is a *brute-force* search in a subset of possible candidates for each hyperparameter. For our experiment, we selected four hyperparameters to tune with the grid search.

1) **Epochs**: One forward and backward pass of the training data through the entire network.
2) **Batch size**: Determines the portion of training data to pass through one epoch.
3) **Input dimension**: Number of nodes in input layer of our neural network.
4) **Hidden layer nodes**: Number of nodes in the hidden layer.

The *difference order* whose optimal value we found by employing *Welch* method. The optimal hyperparameters found during grid search are listed in Table I.

Adam optimizer is used to update network weights and Rectified Linear Unit (ReLU) employed as an activation function of the LSTM model. To calculate the loss function we used Root Mean Squared Error (RMSE). ReLU has an advantage of lower training time and better generalization over *sigmoid* and *tanh* activation functions due to its gradient characteristic. The LSTM network is trained for 20 epochs with the batch size of 80. For test and training dataset, approximately 64,000 samples recorded in an office environment. Figure 5 illustrates the changes in loss function during each epoch. Figure 5 shows the loss function has a clear decreasing trend, indicating the model is in the process of learning. However, the decreasing trend changes to a steady pattern after epoch 20. This means the increase in the training process will not lead to a decline in the loss function. It also justifies the optimal number of epochs (20 epochs) we selected using *grid search* (see Table I). Figure 7 explains the structure of the proposed LSTM model. To implement the LWIPM, we used Python and the *Keras* library.

IV. RESULT AND DISCUSSION

In this section, we evaluate the performance of LWIPM. In addition, we implemented an Auto Regressive Integrated Moving Average (ARIMA) [21] algorithm as a baseline for comparison. We used the seasonal variant of ARIMA and tuned the hyperparameters using grid search and fit the model to our respective datasets. The ARIMA, together with the experimental dataset as a reference, enables us to compare LWIPM performance with competitive prediction algorithms and reality. The RMSE based evaluation of our proposed

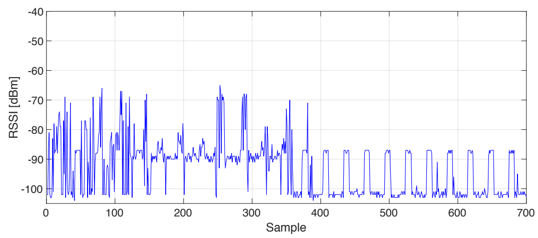

Fig. 6. Randomly sampled time-series plot of BLE channel 15. There are multiple seasonal components seen at different intervals.

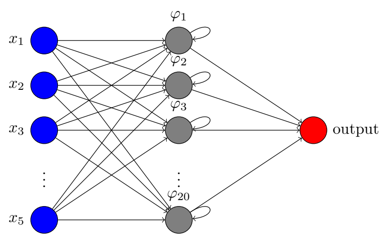

Fig. 7. Network structure according to parameters described in Table I. The blue, grey, and red color show the input layer of five dimensions, a hidden layer with twenty nodes and a single output node, respectively. Each node in the input layer is fully-connected with every node of the hidden layer. x_i and φ_j are notations for input nodes and nodes in hidden layer respectively.

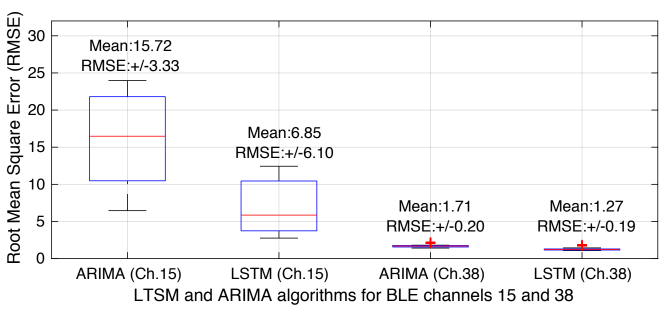

Fig. 8. Statistical distribution for RMSE for LSTM and ARIMA in two different BLE channels.

model for BLE channels 15 and 38 are also shown along with plots in Figure 9. From the recorded experimental dataset, we randomly select multiple data chunks.

The number of samples for each selected chunk is random as well. We split the selected part into training and test set, taking care of the sequence to evaluate model performance for various conditions. After training the model with the training set, we validated the prediction on the data samples by comparing the actual data in the test set with the corresponding prediction values. This results in a sliding training/validation window in time. Figure 8 shows the validation results. We used RMSE in Equation (3) factor as the evaluation metric. The RMSE error here is the difference between the actual and predicted values. It is denoted by $\hat{y}_i - y_i$, where y_i is the actual value for the i^{th} sample and $\hat{y}_i$ is the predicted value.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\hat{y_j} - y_i}{\sigma_i}\right)^2} \quad (3)$$

All the 10 subplots in Figure 9 with their respective RMSE results confirm that the proposed LSTM based LWIPM is able to outperform the ARIMA for both the channels on every train and test sets. In general, due to the uncertainty in Wi-Fi interference behaviour, developing an impeccable algorithm prediction time series model is challenging. Additionally, it is difficult to distinguish if the time series has stochastic state, deterministic chaotic behaviour, or a combination of both [22]. Thus, it is pertinent to perform time series analysis instead of just forecasting. Removing temporal dependency structures is another major factor in increasing the quality of the forecast. Current complex machine learning models can memorize the training set [23], which can lead to estimator even modeling the interference in the data which poorly generalizes to unseen data, which is also known as over-fitting. In Figure 9, it is clear that, while the LSTM and ARIMA models are acting similarly in the low variation scenarios such as Figure 9(b), the performance benefits of the model over the baseline is evident in fluctuation scenarios. Specifically, this can be noticed in Figure 9(a) and Figure 9(i) where ARIMA fails to predict the high fluctuation, but the LSTM model matches the dataset trend more accurately. Similarly, the higher accuracy of the model over the ARIMA model is evident in Figure 9(c) and Figure 9(e). Overall, the results validate the functionality of the proposed model to detect the Wi-Fi interference.

V. Conclusion

In this paper, we studied the characteristics of wireless channels in IoT. We focused on BLE's channels 15 and 38 and developed a LSTM based LWIPM to forecast the Wi-Fi interference level resulting in opportunistic using of the wireless spectrum. The results of the study show that making the time series stationary by incorporating a proper seasonal parameter order can improve the prediction quality. In addition, LWIPM does not suffer from the drawbacks associated with deep learning in CR. For example, deep learning approaches require a large size of samples, high processing power due to the algorithm complexity, and extended processing time. These challenges not only make the deep learning algorithms inoperable for resource constrained IoT devices but also does not meet the real-time decision-making requirements for highly dynamic Wi-Fi interference. As a result, simplicity, together with accuracy is the main advantage of LWIPM over deep learning algorithms.

For future work, it is interesting to develop mobility and network-aware channel prediction models. For example, excluding the interference effect caused by a mobile Wi-Fi device from the training process in machine learning may result in higher accuracy.

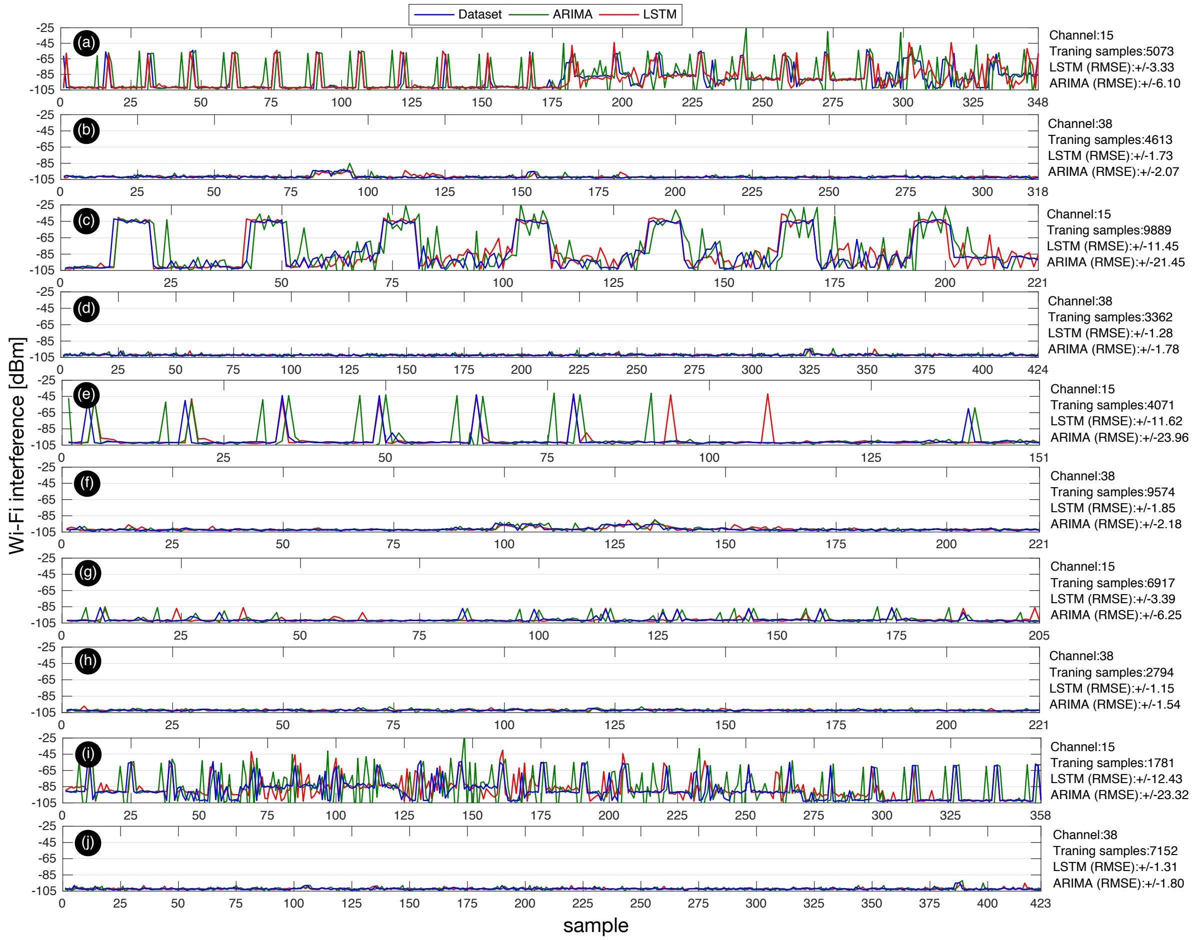

Fig. 9. The result of prediction on BLE channel 38 and 15.

ACKNOWLEDGMENT

This work is funded by the German Federal Ministry of Education and Research (BMBF) as part of the project “DoRIoT” (Dynamic runtime environment for organic (dis-)aggregating IoT-processes) under grant number 01IS18071A. See the project website at http://www.doriot.net.

REFERENCES

[1] A. Nikoukar, S. Raza, A. Poole, M. Güneş, and B. Dezfouli. Low-power wireless for the internet of things: Standards and applications. *IEEE Access*, 6:67893–67926, 2018.

[2] Bluetooth market update 2019 [Online]. Available:https://www.bluetooth.com/bluetooth-resources/2019-bluetooth-market-update/.

[3] Mesut Gunes, Marcel Hecker, and Imed Bouazizi. Influence of adaptive rts/cts retransmissions on tcp in wireless and ad-hoc networks. In *Proceedings of the Eighth IEEE Symposium on Computers and Communications. ISCC 2003*, pages 855–860. IEEE, 2003.

[4] Y. Liang, Y. Zeng, E. C. Y. Peh, and A. T. Hoang. Sensing-throughput tradeoff for cognitive radio networks. *IEEE Transactions on Wireless Communications*, 7(4):1326–1337, April 2008.

[5] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: continual prediction with lstm. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, volume 2, pages 850–855 vol.2, Sep. 1999.

[6] B. Dezfouli, M. Radi, K. Whitehouse, Sh. Razak, and H. Tan. Cama: Efficient modeling of the capture effect for low-power wireless networks. *ACM Trans. Sen. Netw.*, 11(1):20:1–20:43, August 2014.

[7] B. Dezfouli, M. Radi, Sh. Abd Razak, T. Hwee-Pink, and A. Kamalrulnizam. Modeling low-power wireless communications. *Journal of Network and Computer Applications*, 51:102 – 126, 2015.

[8] Felix Juraschek, Mesut Günes, and Bastian Blywis. External interference-aware distributed channel assignment in wireless mesh networks. In *The Fifth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM)*, 2011.

[9] F. Juraschek, M. Günes, and B. Blywis. Measurement-based interference modeling using channel occupancy in wireless mesh networks. In *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6, June 2012.

[10] A. Nikoukar, M. Abboud, B. Samadi, M. Güneş, and B. Dezfouli. Empirical analysis and modeling of bluetooth low-energy (ble) advertisement channels. In *2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pages 1–6, June 2018.

[11] HyungJune Lee, Alberto Cerpa, and Philip Levis. Improving wireless simulation through noise modeling. In *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*, IPSN ’07, pages 21–30, New York, NY, USA, 2007. ACM.

[12] M. Bkassiny, Y. Li, and S. K. Jayaweera. A survey on machine-learning techniques in cognitive radios. *IEEE Communications Surveys Tutorials*, 15(3):1136–1159, Third 2013.

[13] T. Tsiligkaridis and D. Romero. Accelerated reinforcement learning algorithms with nonparametric function approximation for opportunistic spectrum access. *arXiv preprint arXiv:1706.04546*, 2017.

[14] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. Chen, and L. Hanzo. Ma-

chine learning paradigms for next-generation wireless networks. *IEEE Wireless Communications*, 24(2):98–105, April 2017.
[15] Yasir Saleem and Mubashir Husain Rehmani. Primary radio user activity models for cognitive radio networks: A survey. *Journal of Network and Computer Applications*, 43:1–16, 2014.
[16] T. Yucek and H. Arslan. A survey of spectrum sensing algorithms for cognitive radio applications. *IEEE Communications Surveys Tutorials*, 11(1):116–130, First 2009.
[17] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, page III–1310–III–1318. JMLR.org, 2013.
[18] J Merle Elson and Jean M Bennett. Calculation of the power spectral density from surface profile data. *Applied Optics*, 34(1):201–208, 1995.
[19] Scott Miller and Donald Childers. *Probability and random processes: With applications to signal processing and communications*. Academic Press, 2012.
[20] Thomas DiCiccio, Peter Hall, Joseph Romano, et al. Empirical likelihood is bartlett-correctable. *the Annals of Statistics*, 19(2):1053–1061, 1991.
[21] J. Beran. Maximum likelihood estimation of the differencing parameter for invertible short and long memory autoregressive integrated moving average models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(4):659–672, 1995.
[22] A. Golestani and R. Gras. Can we predict the unpredictable? *Scientific reports*, 4:6834, 2014.
[23] R. Caruana, S. Lawrence, and C. Giles. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in neural information processing systems*, pages 402–408, 2001.