Spectrum Sensing and Prediction for 5G Radio

Małgorzata Wasilewska(✉), Hanna Bogucka, and Adrian Kliks

Institute of Radiocommunications, Poznan University of Technology, Poznan, Poland
{malgorzata.wasilewska,hanna.bogucka,adrian.kliks}@put.poznan.pl

Abstract. In future wireless networks, it is crucial to find a way to precisely evaluate the degree of spectrum occupation and the exact parameters of free spectrum band at a given moment. This approach enables a secondary user (SU) to dynamically access the spectrum without interfering primary user's (PU) transmission. The known methods of signal detection or spectrum sensing (SS) enable making decision on spectrum occupancy by SU. The machine learning (ML), especially deep learning (DL) algorithms have already proved their ability to improve classic SS methods. However, SS can be insufficient to use the free spectrum efficiently. As an answer to this issue, the prediction of future spectrum state has been introduced. In this paper, three DL algorithms, namely NN, RNN and CNN have been proposed to accurately predict the 5G spectrum occupation in the time and frequency domain with the accuracy of a single resource block (RB). The results have been obtained for two different datasets: the 5G downlink signal with representation of daily traffic fluctuations and the sensor-network uplink signal characteristic for IoT. The obtained results prove DL algorithms usefulness for spectrum occupancy prediction and show significant improvement in detection and prediction for both low signal-to-noise ratio (SNR) and for high SNR compared with reference detection/prediction method discussed in the paper.

Keywords: Spectrum sensing · Spectrum prediction · Machine learning · 5G · LTE · Convolutional neural network · Recurrent neural network · Neural network · Deep learning

1 Introduction

According to the Ericsson Mobility Report [3], there will be 8,9 billion mobile subscriptions in 2025, out of which 2.8 billion 5G subscriptions are forecast. This number adds to 24,6 billion of machines and devices comprising future Internet of Things (IoT). Moreover, data traffic increased by 20–100% as a consequence

This work was supported by the DAINA project no. 2017/27/L/ST7/03166 "Cognitive Engine for Radio environmenT Awareness In Networks of the future" (CERTAIN) funded by the National Science Centre, Poland.

Z. Deze et al. (Eds.): BDTA 2020/WiCON 2020, LNICST 371, pp. 176–194, 2021.
https://doi.org/10.1007/978-3-030-72802-1_13

of COVID-19 lockdowns. Particularly in times of crisis, digital communication capabilities need to be supported. Future IoT communication also poses challenges, never encountered before. Regarding 5G wireless communication, it aims at achieving 1000 times the system capacity; 10 times the data rate, and spectral efficiency, and 25 times the average mobile cell throughput compared with 4G [2]. One of the enablers of meeting these requirements is system's spectrum awareness and flexible spectrum reuse, the concept contained in the notion of cognitive radio. Cognitive features of 5G are indicated already in [2], and become subject of many publications, e.g. [7], even now, after the 3GPP standardization group announced the completion of 3GPP Release 15 – the first full set of 5G standards in 2018, and its update in 2019 [1] and Release 16 in 2020. The mentioned cognition and wireless intelligence is also envisioned as one of the prerequisites for future 6G communication systems [13].

The main issue regarding spectrum occupancy awareness is how to assure its accuracy, i.e. precision (in terms of high probability of detection and low probability of false alarm) and granularity of detected spectrum opportunities, as well as to efficiently take these opportunities for the transmission purposes. An important direction of research to address these challenges is to apply machine learning (ML) methods to detect and predict the spectrum gaps (temporarily unused frequency bands). This is because spectrum occupation has some time-patterns reflecting daily traffic variations. Moreover, patterns in frequency can be observed due to propagation-dependent resource (channels) allocation among the cells, while spatial correlation reflects shadowing effect in the radio communication channel. ML methods can, thus, be efficient in recognizing these patterns in time, frequency and space.

This paper considers the application of neural networks to improve spectrum sensing and spectrum prediction based on energy values calculation. The energy calculation is used in energy detection (ED) sensing method. i.e. [4,10,15], which is simple and is considered as semi-blind, it does not require any knowledge on signal's properties, however, the noise-level cognition is essential [14]. In this paper the noise estimation is not mandatory, as an information on Signal-to-Noise Ratio (SNR) can be obtained in other ways, for example as a value associated with a given location and formerly calculated. Also the noise level is not important in the decision making process, as a calculated energy is not compared with any threshold to decide on spectrum occupancy - this task belongs to the machine learning algorithm. The considered ML algorithms involve a neural network (NN) with dense-layers, a recurrent neural network (RNN) structure with Long Short-Term Memory (LSTM) layers, and a convolutional neural network (CNN). CNNs are broadly used for image recognition, whilst in our paper, the novelty is to consider them for spectrum sensing and prediction using two-dimensional images formed of energy values and additional features in time and frequency dimensions. As far as it is known to authors, the CNN algorithm for spectrum occupancy prediction are usually used as a sensing or prediction tool for two dimensional data in cooperative sensing [11,12,21], where data collected from each of sensing SUs is merged into a set of input information for CNN algorithm. In [16] also NN, RNN and CNN are applied to predict the type, form and number of transmitting users in a frequency band, but data used for detection and prediction is a one-dimensional time-series data. The long-term prediction that has been based on spatial-spectral-temporal data has been addressed in [19], where a hybrid convolutional long short-term memory has been proposed for future spectrum state prediction. Another interesting hybrid DL approach has been presented in [18] which also exploits CNN and LSTM combination. Here, as an input data a set of IQ samples is considered for each moment in time.

The rest of the paper is organized as follows. In Sect. 2, we first define the system model and the spectrum sensing and prediction problem. In Sect. 3, we describe NN-based algorithmic solutions of the stated problem. In Sect. 4, we present computer-simulation results, whilst in Sect. 5, we derive the conclusions.

2 System Model and Problem Definition

Below, we consider a certain area in which an unlicensed user (called secondary user – SU) aims at detecting and predicting 5G transmission activity of the licensed 5G/4G users (called primary users – PUs) in time and frequency. (Note that 4G LTE-A transmission can be considered as a special case of 5G.) Since 5G numerology allows for high flexibility in resource blocks (RBs) assignment, the goal is to opportunistically make use (by SU) of the spectrum gaps (created by PUs).

In order to define the spectrum occupation state, two hypotheses can be considered. Hypothesis $\mathcal{H}_0$ applies to the situation, when the received signal consists of just the Additive White Gaussian Noise (AWGN). Furthermore, hypothesis $\mathcal{H}_1$ is that the received signal consists of the PU's transmitted signal distorted by the radio fading channel and AWGN. Both hypotheses can be described as:

$$\begin{aligned} \mathcal{H}_0 &: y(t) = n(t), \\ \mathcal{H}_1 &: y(t) = h(t) * s(t) + n(t), \end{aligned} \tag{1}$$

where $y(t)$ is the received signal, $n(t)$ depicts AWGN, $s(t)$ is a transmitted signal, and $h(t)$ is an impulse response of a radio fading channel.

A given sensing algorithm chooses the most probable of the two hypotheses. In order to do so, the algorithm defines test function $T(y)$, which is applied on collected received signal samples y. A decision on spectrum occupancy is made by comparing the value of the test function to threshold value λ defined by algorithm. If the test function value is higher or equal to the threshold, the spectrum is considered to be occupied, otherwise, the spectrum is considered free. The spectrum sensing algorithm's performance is determined by the probability of detection P_{d} (the probability of correctly detecting a present signal) and the probability of false alarm P_{fa} (the probability that signal presence is detected, even though it is not true), i.e.:

$$
\begin{aligned}
P_{\mathrm{d}} &= \Pr\{T(y) > \lambda | \mathcal{H}_1\}, \\
P_{\mathrm{fa}} &= \Pr\{T(y) > \lambda | \mathcal{H}_0\}.
\end{aligned} \tag{2}
$$

Transmission detection is the decision regarding the present state of spectrum occupation, i.e., at time moment t, while SU may be interested in prediction of the future states. For the prediction, the main issue is to make a decision on a future spectrum state, i.e. at the next, or several next time moments in the time interval [t, t+τ], where $\tau > 0$. This decision, however, is based on the current signal data, i.e., collected at time moment t.

Although there are multiple well known spectrum detection methods, ML algorithms have proved their usefulness in spectrum sensing area. Future spectrum state prediction is another area in which machine learning performs well thanks to its adaptability and ability to find patterns in input data.

In the problem described in the paper, SU is collecting data samples of a received, distorted by channel signals. We assume a system consisting of one base station (BS) transmitting to multiple users. The considered SU is receiving the BS's signal and is trying to decide by employing machine learning (ML) techniques whether it is possible to transmit now or in near future. SU is collecting signal samples and calculates energy for every RB in every first OFDM symbol in a given time slot. Having this information, SU tries to make a decision whether a considered slot is occupied. By calculating energy in a single OFDM symbol SU has time to make a decision and transmit in the remaining part of the time slot. It would be beneficial however, to simultaneously gain knowledge of occupancy of the same RBs in the future time slots in order to prepare for longer transmission. Proposed ML algorithms try to evaluate occupancy for current and six next time slots. The ML input data is calculated based on energy values per RB. The ML algorithms are trained separately for different signal to noise ratio (SNR) values.

3 Deep Learning for Spectrum Sensing and Prediction - Algorithmic Solution

Deep Learning (DL) algorithms are known for their ability of finding complicated dependencies in input data [8]. In the prediction problem, it is crucial to recognize any patterns that may occur in the receiving signal and DL algorithms should be a good choice. Three DL algorithms have been implemented for spectrum sensing and prediction. First, NN algorithm has been implemented as an example of a simplest algorithm. The second algorithm is a RNN algorithm. The RNNs are usually used in language and audio signal processing, as a tool of predicting sequences [6]. Their particular usefulness in this field is due to the fact that basic elements of RNNs layers called cells feedback their own output as an additional input information which makes possible for RNNs to notice intricate patterns occurring in input data in time. The last algorithm implemented is a CNN. CNNs are broadly used for image recognition, processing and classification [17].

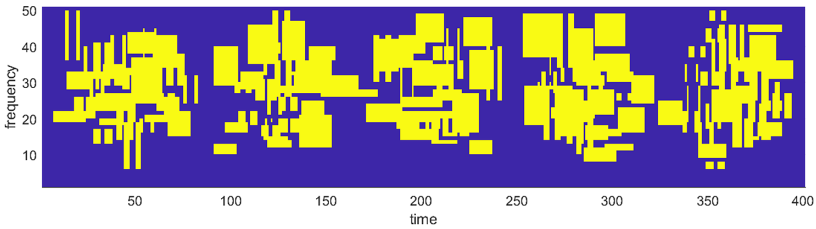


Fig. 1. First dataset - cyclical intensity of RBs occupancy

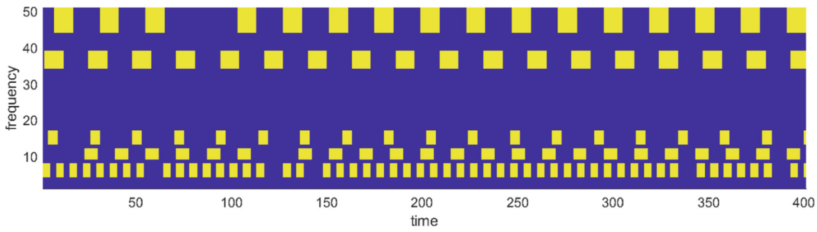
All of the proposed algorithms are supervised classification algorithms. Each one of them is trying to establish an occupancy status of a current of future RBs, so it indicates a binary classification problem. The proposed NN-based method is the simplest and requires least calculations, but is able to perform only a single RB classification. This means that based on the current input data, the NN classifies one current or one of the future RBs' as free or occupied. On the other hand, the RNN as a more complicated method, is able to perform a detection and prediction for several next RBs, but only for single frequency range. The CNN is able to perform most complex calculations and classify multiple RBs both in time and in frequency.

Each proposed algorithm receives slightly different input data based on RB energy values. A single input dataset for NN algorithm consists of four values that characterize a single RB: frequency index (values from range 0–49), time slot index (values from range 0–79), energy value for the considered RB, and sum of energies of neighboring RBs. Each element of an input sequence of RNN consists of 3 values: time slot index, energy value, and sum of neighboring RBs' energy values. To take full advantage of CNN advantages, the input data in proposed algorithm is constructed as an 2D image, whose first dimension is frequency, second dimension is time, and pixels contain RB energy values. Input images also have three layers, similarly as color images have three RGB components. The first layer consists of aforementioned RBs' energy values, the second layer contains frequency index values, and the third one, the time slot indexes.

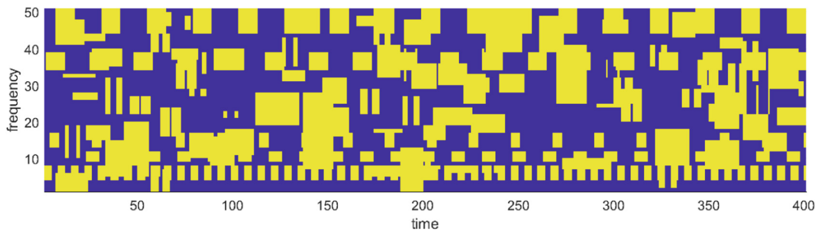
4 Simulation Experiment

4.1 Assumptions and Settings

Two different cases of signals received by SU are considered. First signal is a symbolic representation of daily fluctuations in traffic intensity typical to wireless communication systems. Figure 1 presents signal for 400 slots as RBs in frequency and time. The yellow areas indicates occupied RBs, and blue means free RBs. The intensity dependency in time is clearly visible, as an intensity rises and drops every 80 slots. The signal is correlated in time, also, probability of occupied



(a) Resources occupied only by sensors signals



(b) Resources occupied by sensors signals and random signals

Fig. 2. Second dataset - sensor network signals

RB in frequency is not uniform, which is dictated by the fact that the channel may prevent effective transmission on certain frequencies to some users. In this example it is assumed, that signal is most probable to appear on the middle frequencies, and least probable on the marginal frequencies.

The second considered case concerns a system, where there are multiple sensor-like devices that need to transmit information in the form of short signals and with high periodicity. The signal occurs in every cycle with high probability, although from time to time devices hold back the transmission. The simulated sensors' transmitted signal is presented on Fig. 2a. Additionally to the sensors' signals, a random signal which is characterized by a certain correlation over time is transmitted with uniform probability throughout the band. The random signal with sensors signal is presented on Fig. 2b.

As mentioned before, three DL algorithms have been implemented. Figure 3 shows a structure of the NN model. It consists of three dense layers, proceeded by a Softmax layer [5], which convert received data into probability values. First two dense layers consist of 10 neurons. The last dense consists of two neurons.

Neural network is used here for classification problem, so as a loss function, Sparse Categorical Cross Entropy [8] has been used. The output data consists of two probabilities – probability of a considered RB belonging to 'occupied' category and probability of belonging to 'free' category. Those two probability values sum up to one. In order to achieve a classification result for current RB's signal detection or for further RB occupancy prediction, separate NNs must be trained. In the experiments shown in the paper, detection, and prediction from first to sixth next time slot is performed, which required to create seven NNs

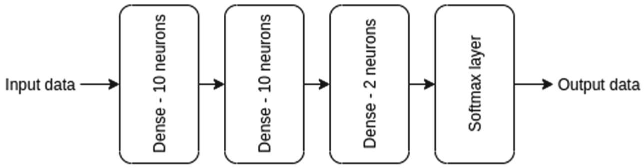


Fig. 3. NN algorithm model

one four each application. The Stochastic Gradient Descent optimizer [20] has been used in training process.

The proposed RNN model consists of three Long Short-Term Memory (LSTM) layers [9]. A dropout of 0.5, 0.3 and 0.2 is applied after each of LSTM layers respectively to prevent overfitting. The last layer is a time distributed dense layer consisting of 4 neurons. A sequence consisting of 100 feature sets is provided as input. The output consists of sequence of probabilities. First probability value concerns probability of current RB being occupied. The three next probabilities are for predicting occupancy of next RBs for the same frequency, but future time slots. Since the RNN accepts as an input a one-dimensional data, there is a need of training separate RNNs for each frequency separately. The Adam optimizer [20] has been used, with learning rate 0.001. As a loss function, binary cross entropy has been implemented. Figure 4 portrays the RNN model.

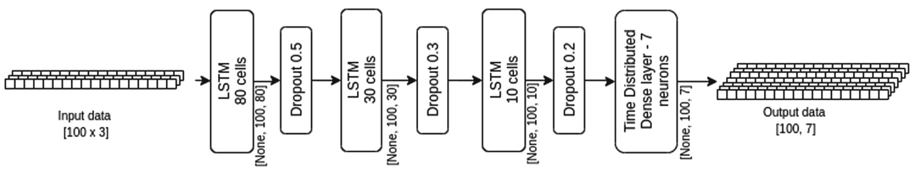


Fig. 4. RNN algorithm model

The last implemented algorithm is a CNN. Figure 5 shows proposed algorithm's model. This method accepts as an input a spectrogram-like image of energy values and other features per each RB. The input data is padded with zero values on the top, bottom and on the right side. Output results achieved by this network, are two-dimensional layers of the size of 50 pixels by 7 pixels. It means that a results is a detection and prediction results for each of 50 frequencies for current and next 7 time slots. One pixel represents one RB.

The created CNN model consists of four convolutional layers. The first one has 8 kernels (filters) are the size of 9 by 80 pixels. This layer returns image of the same size as input. The second layer uses 16 kernels of size 5 by 50, and the third one uses 32 kernels of size 3 by 25. The growing number of kernels in each layer is to ensure better recognition of any more abstract features of input data. The output layer has only two kernels, each for every RB's occupancy category - free or occupied. The filters are the size of 1 by 27, to ensure a proper output image size. Each of the layers uses rectifier function (ReLU) as an activation function, except of the last layer, which uses softmax function. As an optimizer, Adam algorithm is implemented with learning rate equal 0.0001. Since the output consists of two categorical probabilities, the Sparse Categorical Crossentropy is used as a loss function.

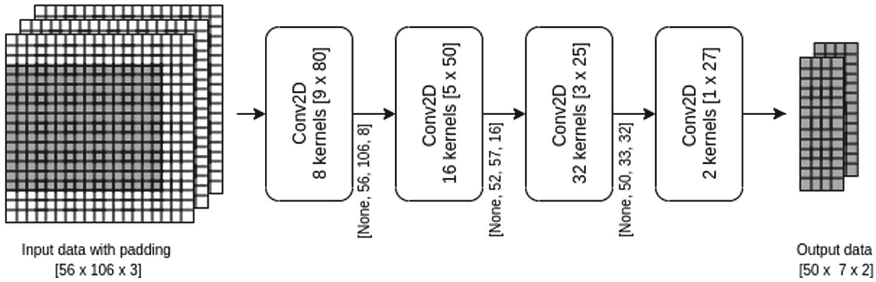


Fig. 5. CNN algorithm model

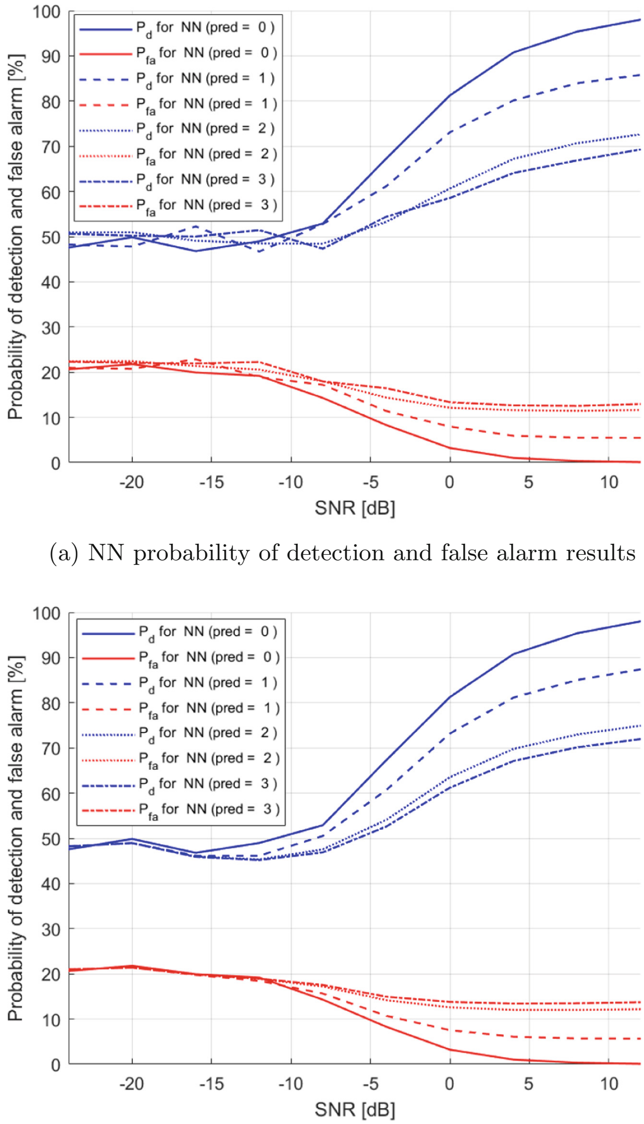
4.2 Simulation Results

In order to test algorithms' performance on detection and future spectrum state prediction, experiments have been conducted for different SNR values. Additionally a primitive algorithm (PA) for prediction has been proposed in order to evaluate whether DL algorithms introduce any improvement into prediction. This primitive method uses detection results of a currently considered ML and assumes that all RBs for a given frequency in every future time slot that is considered will have the same occupancy state as those that just have been detected. A simple prediction method as that can give quite good results if the spectrum is occupied most of the time, or it is occupied in continuous in time groups of RB. The latter case is true in both of considered datasets.

For all of the considered ML algorithms and their corresponding PA results, plots of P_{d} and P_{fa} for a range of SNR values have been obtained. To facilitate the interpretation of the results, the bar charts have been drawn of the probabilities P_{d} and P_{fa} for each of prediction steps. These charts are created for highest SNR value, where the difference between ML and PA is the biggest. Additionally, an overall measure of improvement - a total difference between both probabilities has been determined, according to the formula:

$$D_{total} = P_{\mathrm{d}_{\mathrm{ML}}} - P_{\mathrm{d}_{\mathrm{PA}}} + P_{\mathrm{fa}_{\mathrm{PA}}} - P_{\mathrm{fa}_{\mathrm{ML}}}, \tag{3}$$

where D_{total} is a total evaluation measure, $P_{\mathrm{d}_{\mathrm{ML}}}$, $P_{\mathrm{fa}_{\mathrm{ML}}}$, $P_{\mathrm{d}_{\mathrm{PA}}}$, $P_{\mathrm{fa}_{\mathrm{PA}}}$ are probabilities of detection and false alarm for a considered ML and for a corresponding PA respectively.



(a) NN probability of detection and false alarm results

(b) PA probability of detection and false alarm results based on NN)

Fig. 6. NN and PA results depending on SNR for first dataset

First Dataset. Results for first dataset have been achieved by applying all of the proposed DL algorithms. Figure 6 shows plots of probabilities of detection and false alarm for detection and prediction for next three slots. Figure 6a shows probabilities for NN algorithm, while Fig. 6b contains results for primitive prediction based on NN detection results. It can be observed that for low SNR values, NN algorithm is able to achieve probability of detection equal around

50%, while keeping lower values of probability of false alarm around 20%. Typical detection algorithms like ED method usually achieve the same values of P_{d} and P_{fa} for low SNR, so results achieved by NN are beneficial, although P_{fa} could be considered as high depending on the detection/prediction requirements for transmission. The results for prediction and detection for both NN and PA are the same for low SNR. This is a common rule among all of the following results. For low SNR, the noise prevents any time-depending sequences being found, hence there are no differences in the detection and prediction results. At the same time, there are some dependencies of resource allocation in time, which is a reason for a significant difference between P_{d} and P_{fa}. All of the results of NN and PA for high SNR, are very similar, although PA shows a slight advantage.

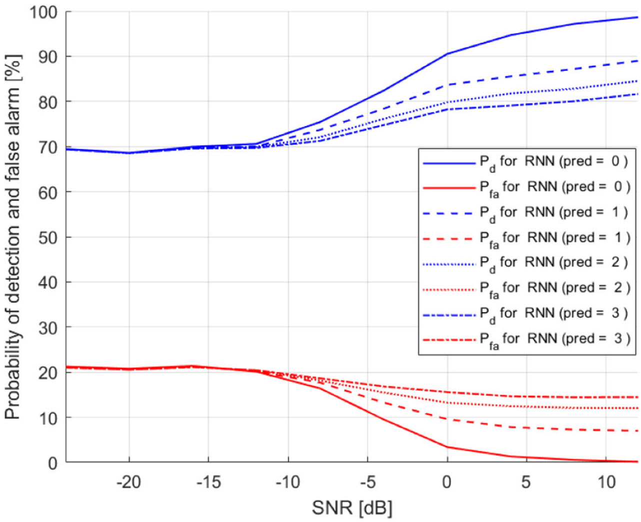
The differences in results for SNR = 12 dB are much easier to compare on Fig. 9. Here, all of the blue bars correspond to P_{d} and P_{fa} NN results on top and bottom plot respectively. The grey bars overlapping blue bars represent PA results based on NN detection. It can be observed that P_{d} for all predictions for NN is lower than PA results, but at the same time P_{fa} results are lower. The overall prediction evaluation is presented on Fig. 10. Figures 9 and 10 are described in more detail below.

For second set of results, the RNN results are presented on Fig. 7. Here, the differences between RNN and PA results are more visible, especially for P_{d}, which is significantly higher for RNN for prediction of next second and third slots. The gap between P_{d} and P_{fa} results for low SNR is even more substantial than for the NN. Here, the P_{fa} stays on value of 20%, while P_{d} reaches values of even 70%. For high SNR values and for bigger number of prediction steps, RNN results are also compared in more detail on Fig. 9.

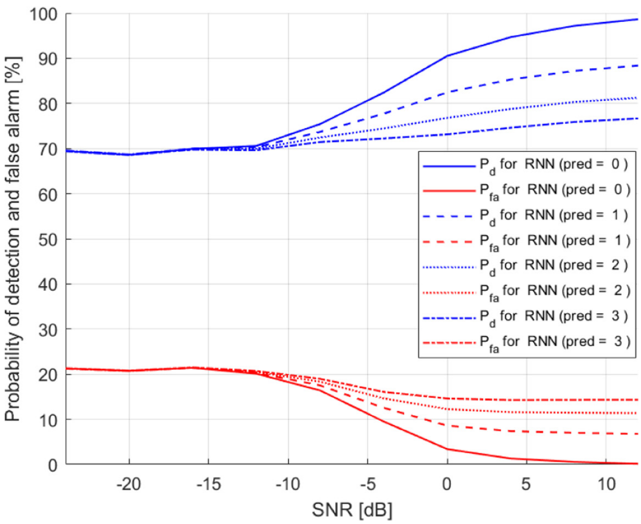
CNN results are presented on Fig. 8. It can be expected, that CNN would work as good or even better than RNN algorithm. The achieved results prove it's true. The results of P_{d} for high SNR are almost as good as those of RNN, and P_{fa} are even lower. Although the P_{d} values for low SNR are not as high as RNN P_{d}, the P_{fa} are lower, which ensures that the difference of these results is preserved, and still equal to around 50%.

As mentioned above, all of the detection and prediction results for high SNR are compared collectively in order to evaluate which of the algorithms works best on the first dataset. On Fig. 9 one can observe differences between P_{d} (top chart) and P_{fa} (bottom) for each of DL algorithms and their corresponding PA. Each of the DL bars has a corresponding overlapping grey PA bar. It can be observed that PA results of prediction are very dependent on ML algorithm detection that they are based on. For instance PA based on NN are significantly worse than PA based on RNN and CNN, although all three ML algorithms reach very similar results for detection (prediction = 0). The P_{fa} for NN based PA grow much faster with each prediction step. In fact for sixth predicted slot, the P_{fa} and P_{d} values of PA are only 15% apart.

The NN P_{d} results are usually a few percent worse than PA results, but since $P_{\mathrm{fa_{PA}}}$ values grow fast with each prediction step, $P_{\mathrm{fa_{NN}}}$ reach lower values, although are still higher than P_{fa} of other DL algorithms. It is also quite clear



(a) RNN probability of detection and false alarm results

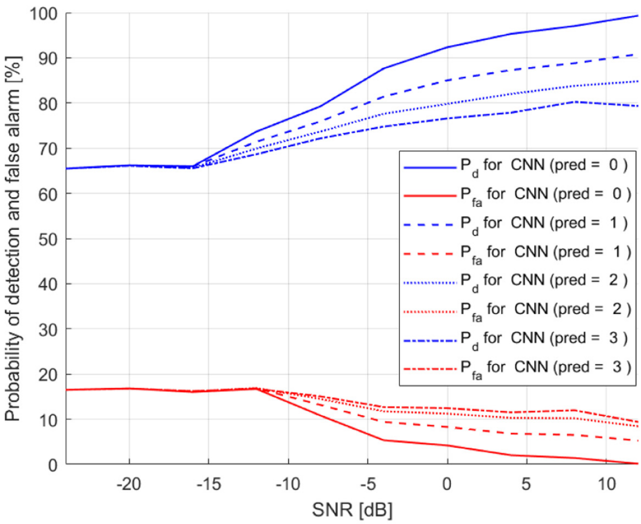


(b) PA probability of detection and false alarm results based on RNN

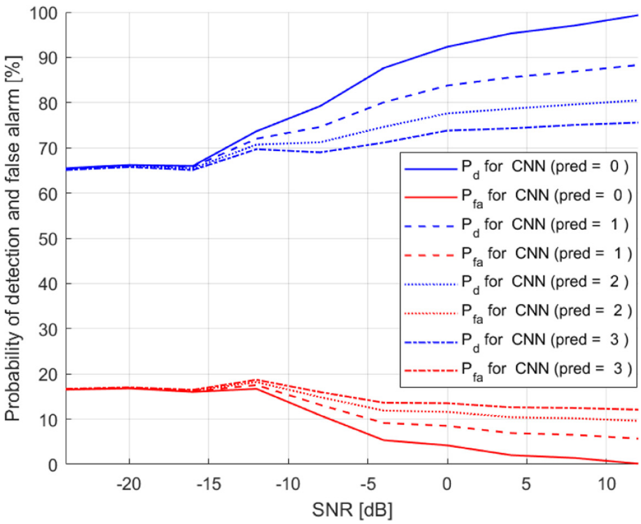
Fig. 7. RNN and PA results depending on SNR for first dataset

that the best P_{d} results are achieved by RNN, but the best P_{fa} belong to CNN algorithm. Either way, both methods work comparably for first dataset.

The top graph of Fig. 10 shows how each of DL algorithms performs comparing to it's PA. These results were obtained through use of Eq. 3. This bar graph represents whether it is better to use simple PA method based on a chosen ML algorithm and how much better or worse performance can be achieved. Figures 9



(a) CNN probability of detection and false alarm results



(b) PA probability of detection and false alarm results (CNN)

Fig. 8. CNN and PA results depending on SNR for first dataset

and 10 provide the complete set of information needed for the assessment of a given method. The negative results indicate that a PA is better than a ML for a given prediction step. The bottom part of Fig. 10 presents also comparison between ML results and PA results, although only one set of PA results is chosen to be compared with every ML result. It is dictated by the fact that some PA

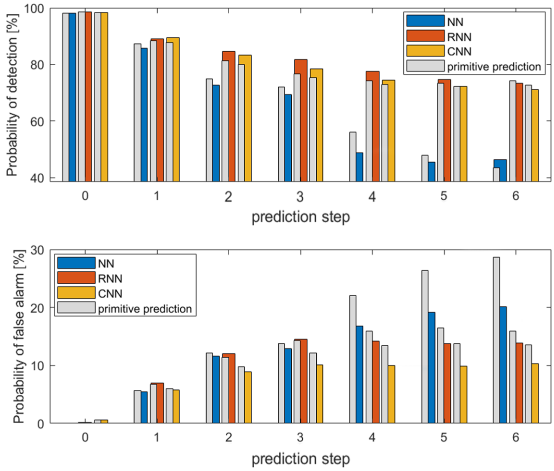


Fig. 9. Comparison of probability of detection and probability of false alarm values for SNR = 12 dB (first dataset)

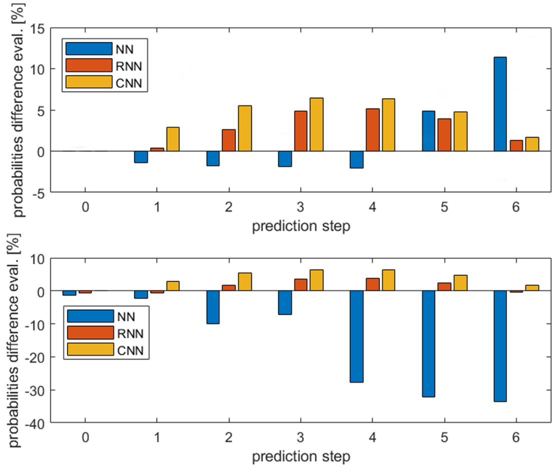
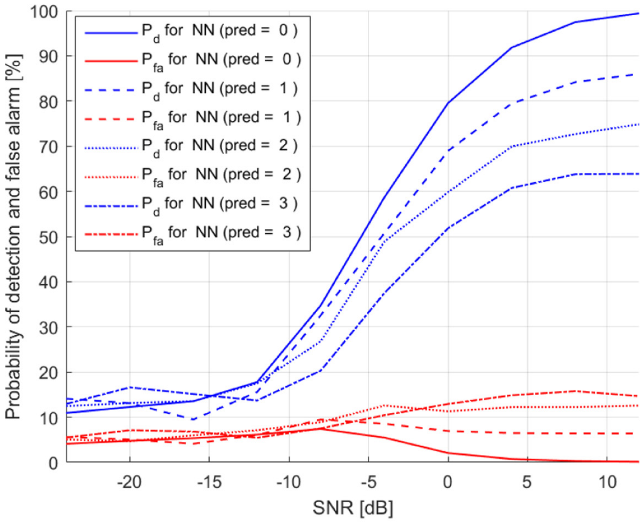
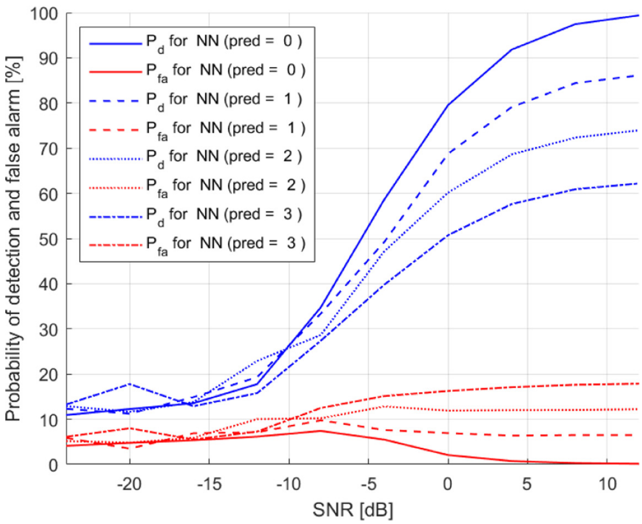


Fig. 10. Final results evaluation in compare with PA results for SNR = 12 dB (first dataset)

results for a given DL can be much worse than ML and give false impression on a top graph that a given ML performs better than the others. The example of this phenomenon is evaluation result for NN algorithm for prediction 6, where the $P_{\mathrm{fa}_{\mathrm{PA}}}$ are so high that NN appears to achieve significant improvement, when it fact it still performs much worse than RNN and CNN. To address this issue, on the bottom graph, all of the DL results have been compared, to the PA results of CNN, which are the best. It is clearly visible now that NN is the worst algorithm for this dataset.



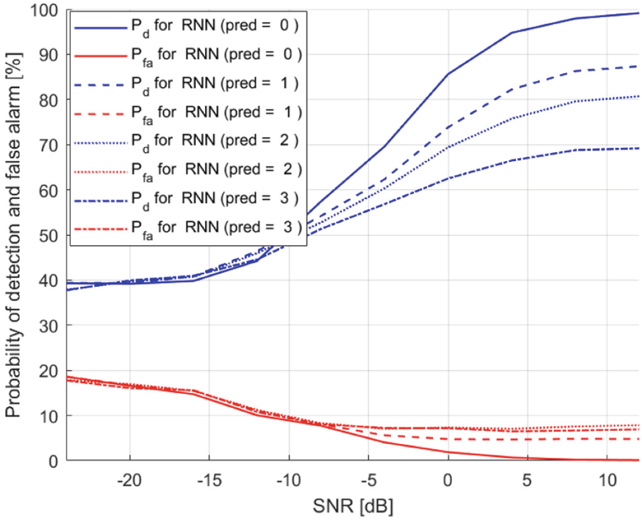
(a) NN probability of detection and false alarm results



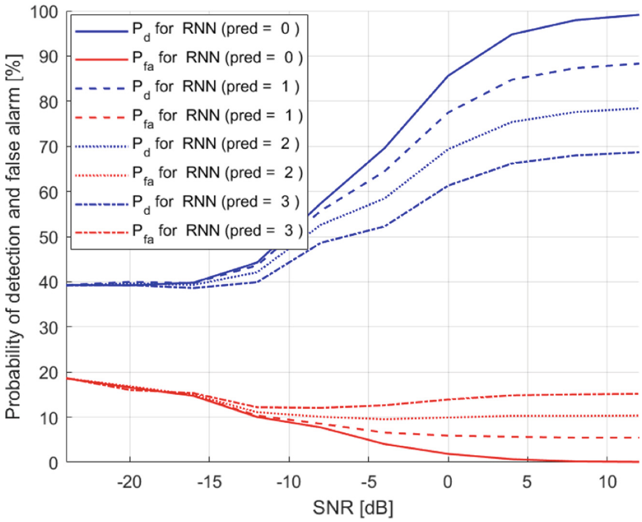
(b) PA probability of detection and false alarm results (NN)

Fig. 11. NN and PA results depending on SNR for second dataset

Second Dataset. As a second dataset a signals from sensor network has been considered. Unlike the results of first dataset, this time the gap between P_d and P_fa for low SNR for all of the considered DL algorithms. Figure 11 presents results for NN and NN-based PA. The P_d for low SNR is equal only around 12%, while P_d equals 5%.



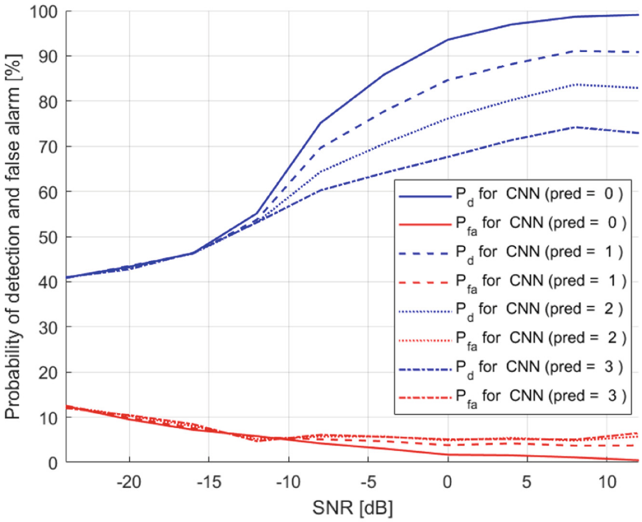
(a) RNN probability of detection and false alarm results



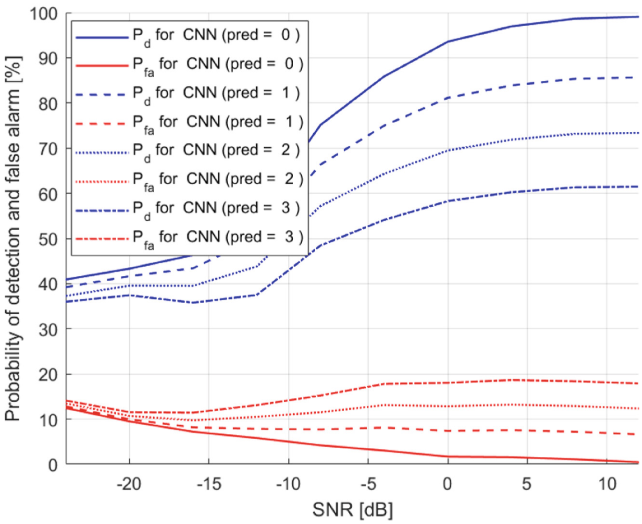
(b) PA probability of detection and false alarm results (RNN)

Fig. 12. RNN and PA results depending on SNR for second dataset

Figure 12 presents results for RNN algorithm. Same as for the first dataset, in case of the second dataset RNN also performs generally better in terms of both P_{d} and P_{fa} than NN. The RNN is better suited for recognizing the patterns in signal despite the noise and random signals. For low SNR values it is not possible anymore to associate a specific time with higher or lower communication traffic intensity, but it is still possible to predict sensor's activity, hence the gap between



(a) CNN probability of detection and false alarm results



(b) PA probability of detection and false alarm results (CNN)

Fig. 13. CNN and PA results depending on SNR for second dataset

P_{d} and P_{fa}. It is worth noting that for high SNR, values P_{fa} of RNN are lower than corresponding values of P_{fa} for RNN-based PA.

Figure 13 shows results for CNN. In this case, the differences between the two CNN and CNN-based PA results can be seen more clearly than before. The advantage of the CNN algorithm over PA in terms of high SNR P_{d} is growing with each prediction step. Also it is very clear that P_{fa} can be achieved much lower thanks to the use of CNN than PA.

Just like in case of first dataset, the comparison of high SNR P_{d} and P_{fa} results for multiple prediction steps has been collectively compared on Fig. 14. As previously, the top graph presents evaluation results of each ML results compared to their corresponding PA results and the bottom graph shows evaluation of each ML results compared with the best PA results, which are in this case RNN-based PA. Also in that case, the CNN algorithm appears to be the best choice in terms of overall P_{d} and P_{fa} performance in compare to other two algorithms, which is clearly visible on Fig. 15. The NN is not able to outperform RNN-based PA for any prediction step higher than 0 although it does outperform its own PA results for prediction from 1 to 3.

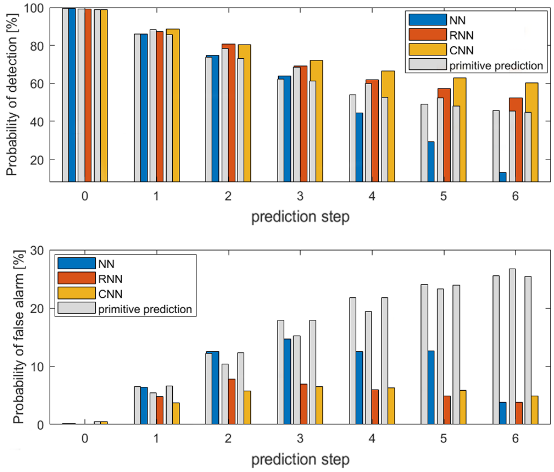


Fig. 14. Comparison of probability of detection and probability of false alarm values for SNR = 12 dB (second dataset)

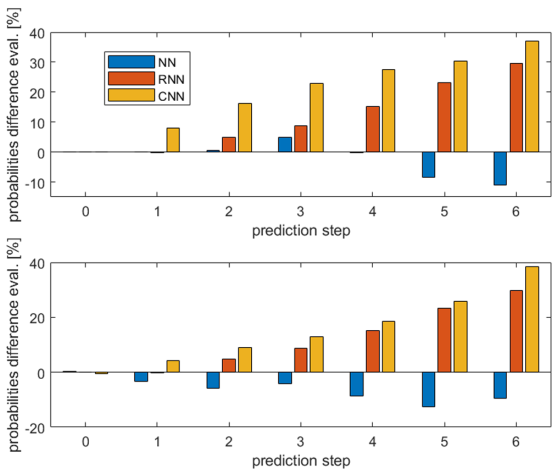


Fig. 15. Final results evaluation for SNR = 12 dB (second dataset)

5 Conclusions

In this paper, we considered three deep learning algorithms for 4G/5G spectrum detection and prediction. Two datasets have been investigated. The chosen datasets represent different cases of communication systems, namely the periodic changes in communication traffic intensity and sensor networks with repeating signals. Neural Network, Recurrent Neural Network and Convolutional Neural Network have been used for detection and future spectrum state prediction. All of them present different set of advantages and disadvantages, although the CNN turned out to be the best fitting method in both of the considered cases. All three of the algorithms have been compared to the very simple detection/prediction method based on each of the machine learning methods, called as primitive algorithm in the paper. In the paper, there has been derived an overall evaluation of proposed algorithms in order to facilitate the selection of the most appropriate algorithm for specific needs.

References

1. 3GPP technical specification group services and system aspects. https://www.3gpp.org/release-15
2. 5G vision: the next generation of communication networks and services (2015). https://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf
3. The Ericsson mobility report, June 2020. https://www.ericsson.com/49da93/assets/local/mobility-report/documents/2020/june2020-ericsson-mobility-report.pdf
4. Alom, M.Z., Godder, T.K., Morshed, M.N., Maali, A.: Enhanced spectrum sensing based on energy detection in cognitive radio network using adaptive threshold. In: 2017 International Conference on Networking, Systems and Security (NSysS), pp. 138–143 (2017)
5. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
6. De Mulder, W., Bethard, S., Moens, M.F.: A survey on the application of recurrent neural networks to statistical language modeling. Comput. Speech Lang. **30**(1), 61–98 (2015)
7. Ahmad, W.S.H.M.W., et al.: 5G technology: towards dynamic spectrum sharing using cognitive radio networks. IEEE Access **8**, 14460–14488 (2020)
8. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
10. Joshi, M., Borde, S.D.: Comprehensive analysis of various energy detection parameters in spectrum sensing for cognitive radio systems. In: 2014 International Conference on Advances in Communication and Computing Technologies (ICACACT 2014), pp. 1–4 (2014)
11. Lee, W., Kim, M., Cho, D.: Deep cooperative sensing: cooperative spectrum sensing based on convolutional neural networks. IEEE Trans. Veh. Technol. **68**(3), 3005–3009 (2019)

12. Liu, H., Zhu, X., Fujii, T.: Ensemble deep learning based cooperative spectrum sensing with semi-soft stacking fusion center. In: 2019 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6 (2019)
13. Latva-aho, M., et al.: Key drivers and research challenges for 6G ubiquitous wireless intelligence. 6G Research Visions 1 (2019)
14. Mariani, A., Giorgetti, A., Chiani, M.: Effects of noise power estimation on energy detection for cognitive radio applications. IEEE Trans. Commun. **59**(12), 3410–3420 (2011)
15. Ribas, A.O.P., Dias, U.S.: On the double threshold energy detection-based spectrum sensing over κ-μ fading channel. In: 2015 IEEE Radio and Wireless Symposium (RWS), pp. 82–85 (2015)
16. Omotere, O., Fuller, J., Qian, L., Han, Z.: Spectrum occupancy prediction in coexisting wireless systems using deep learning. In: 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), pp. 1–7 (2018)
17. Qin, Z., Yu, F., Liu, C., Chen, X.: How convolutional neural network see the world-a survey of convolutional neural network visualization methods. arXiv preprint arXiv:1804.11191 (2018)
18. Roy, D., Mukherjee, T., Chatterjee, M., Pasiliao, E.: Primary user activity prediction in DSA networks using recurrent structures. In: 2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), pp. 1–10 (2019)
19. Shawel, B.S., Woldegebreal, D.H., Pollin, S.: Convolutional LSTM-based long-term spectrum prediction for dynamic spectrum access. In: 2019 27th European Signal Processing Conference (EUSIPCO), pp. 1–5 (2019)
20. Sra, S., Nowozin, S., Wright, S.J.: Optimization for Machine Learning. MIT Press, Cambridge (2012)
21. Yu, L., et al.: Spectrum availability prediction for cognitive radio communications: a DCG approach. IEEE Trans. Cogn. Commun. Netw. **6**(2), 476–485 (2020)