Proceedings of the 11th International Conference on Applied Informatics
Eger, Hungary, January 29–31, 2020, published at http://ceur-ws.org

IEEE 802.11n/ac/ax Hot Zone Traffic Evaluation with Neural Compute Stick Based RNN Methods

Zoltán Gál[a], Péter Polgár[b]

[a]Faculty of Informatics, University of Debrecen, Debrecen, Hungary
gal.zoltan@unideb.hu

[b]Faculty of Informatics, University of Debrecen, Debrecen, Hungary
pulgarpeter@gmail.com

Abstract

The longer than twenty-two-year success marching of the IEEE 802.11 communication technology continues in the next years with new standard editions having transfer rate in the multi Gbit/s range. Realistic evaluation of the WiFi controller supervised hot zone service level becomes more and more critical because of the very high number of frames transmitted per unit of time. Online evaluation of the content transmission efficiency on radio channel is affected by several conditions including environment reflection characteristics, multipath influences, movement behaviour of the users and time dependence of the mobile terminals population in the service area. Based on our anterior investigations we found that in special places of the coverage area with WiFi hot zone service high ratio of transmitted frames are temporarily control and management frames even in case of communications with low level of the radio signals. To scan and evaluate IEEE 802.11/n/ac/ax channel usage efficiency we developed a complex scanner and evaluator tool based on neural network stick hardware. The software prototype developed utilize Long-Short Term Memory and Gated Recurrent Unit functions to determine periodically the percent of data frames of the total transmitted radio frames. Constant number of frames and constant time intervals, respectively are applied as two basic approaches of our evaluation methods. Advantages, weaknesses and usability cases in practice of the proposed solutions will be given in the paper.

Keywords: Internet of Things (IoT), Wireless Fidelity (WiFi) Hot Zone, Quality of Service (QoS), Recurrent Neural Network (RNN), Long-Short Term

Memory (LSTM), Gated Recurrent Unit (GRU), Convolutional Network, routing, clustering, time series classification.

MSC: 65C60, 60G35, 91A28

1. Introduction

Due to the advantageous properties of IEEE 802.11 (WiFi) wireless technology, it has been present with unbroken popularity in the access environment of network infrastructures for over twenty years of its development. Several stable versions (a/b/g/n/ ac/ad/ax) have been used recently for SOHO and other corporate LANs. Apart from its security aspects, each newer standard has been designed to improve communication efficiency and enhance the user experience. There has been relatively little comprehensive research on the complex analysis of frame streams by category, since measurement and processing require mathematical proficiency in addition to technological knowledge. It is also cumbersome to interpret traffic which is grouped by over seventy subtypes of the three main types of WiFi frames. Access to these has only recently become possible with a public software-based protocol analyzer. Using this changed feature, we performed WiFi network measurements, and analysis by traffic type.

The further structure of the paper is as follows: the second section lists some of the related neural network applications used for wireless networks. In the third section, we introduce the most important features of IEEE 802.11 transmission technology. Introducing the neural network stick and using it for fast data processing is discussed in the fourth section. The fifth section deals with the deep learning analysis of IEEE 802.11n/ac/ax traffics. The last section provides a summary of the analysis work described and its possible continuation of the research work.

2. Related work

In recent years, many application of neural networks (NNs) have been used in wireless networks. This is because using NNs allows us to, among other things, map the input data to predictions or classes, e.g. error rate prediction, signal classification, to predict throughput. NNs can be used to model non-linear systems to ensure quality of service (QoS), predict device location, and more. The use of NNs can be divided into two main parts: supervised learning; unsupervised learning. The most common types of supervised NNs used in wireless networks are Multilayer Feed-Forward NNs (ML-FFNNs), and Recurrent NNs (RNNs). The methods described in this paper relate to supervised learning and RNN. ML-FFNNs can learn the relationship between the given inputs and their corresponding outputs through the training process. Using the resulting NN model, it is possible to evaluate outputs for new inputs. RNNs are similar to ML-FFNNs except that they also have a cyclic nature. This allows us to introduce time stamps, which make it possible to model time dependent systems; i.e. the output for the following input example may depends on the previous examples. Use of RNNs in WiFi frame type ratio regression has not been used till now as we know.

Some supervised NN applications in wireless networks are summarized in the section below. Battiti et al. have created an ML-FFNN-type NN that can predict the x and y coordinates of a given device location based on RSSI signal intensity values from three APs [1]. Ahmad et al. have examined various modular ML-FFNN networks that can handle situations where the signal is not available from certain APs [2]. Shareef et al. wanted to determine the position of a mobile device (WSN localization) based on RSSI signals from three APs using NNs. Three different types of NNs were used: ML-FFNNs, Recurrent NN, Radial NN. According to the research, Radial NNs are the most accurate, but more expensive in terms of memory and computing, while ML-FFNNs are the least intensive [3]. The application from Laoudias et al. assigns one of the 100 reference regions to a mobile node location based on RSSI values from 10 APs. Radial Basis NNs provided the most accurate results, performing better than ML-FFNN with CRNN [4]. Taok et al. claim that the localization with Ultrawide Band (UWB) radio signals using ML-FFNN provides accurate results, as the adapted radio signals are more robust to multipath interference and noise [5].

Altini et al. worked on localization with Bluetooth signals using multiple NNs, each NN specific to a different positional orientation of a user, and each NN is a ML-FFNN [6]. Li et al. researched localization of non-GPS Bluetooth phones connected to GPS beacons using Recurrent NN. This method provides GPS coordinates for the non-GPS phones [7]. Ju and Evans developed application of ML-FFNN for MANET QoS routing, the network takes delay and packet loss as inputs, and predicts link load and max. link bandwidth as outputs. The routing protocol uses these two metrics to predict incremental throughput [8]. Barabas et al. studied load balancing routing, predicting traffic characteristics using ML-FFNN for the links that use the actually available transfer rate and delay. These predictions are useful for deciding which alternative routes to take when traffic is excessive [9]. Zhi-yuan et al. used wavelet NN to decide what primary and secondary nodes will be chosen for MultiPath load balancing in MANETs [10].

Baldo and Zorzi proposed usage of ML-FFNN to predict delay, throughput, and reliability using the SNR and error rate as inputs [11]. Katidiotis et al. took signal strength and link quality as input to predict max. throughput using ML-FFNN. Past max. throughput values have also been used to predict future throughput values [12]. Tumuluru et al. consider that the spectrum consists of slots. Each slot has a past status, which is either busy or idle. The recorded past status of these slots is used by an ML-FFNN to predict future slot status [13]. Moustapha and Selmic evaluated fault detection in WSN using a recurrent NN. Deploying an RNN on each node that takes input data from adjacent nodes as well. The data sensed by the node is compared with the output of the RNN. If the difference between the two is greater than a set value, the sensor is declared faulty [14]. Capka and Boutaba researched mobility prediction in cellular networks using an ML-FFNN. Use of past mobile terminal-AP connectivity to predict which AP would take over management of the service [15]. Stegmayer and Chiotti proposed a model based on a Time-Delayed Neural Network (TDNN) that has the capability of learning and predicting the dynamic behavior of nonlinear amplifiers which part of wireless transmitters. The purpose of the model is to speed up system deployment by reducing modeling time [16].

3. Behavior aspects of the IEEE 802.11n/ac/ax technology

The IEEE 802.11 transmission technology further continues the success story of its development. The most important technological features of the latest three standard versions are summarized in Table 1.

Viewpoint / Feature	**IEEE 802.11n HT (High Throughput)**	**IEEE 802.11ac VHT (Very High Throughput)**	**IEEE 802.11ax HEW (High Efficiency W-less)**
Freq. domain	2,4 | 5 GHz	5 GHz	2,4 | 5 | 6 GHz
Modulation	OFDM; DSSS/CCK	OFDM; DSSS/CCK	OFDM; 1024-QAM
Ch. Bw	20 | 40 MHz	20 | 40 | 80 | 160 MHz	20 | 40 | 80 | 160 MHz
Max. Tx rate	600 Mb/s	6,93 Gb/s	11 Gb/s
Max. range	indoor: 70 m | outdoor: 250 m	$\sim$ 80 m	$\sim$ 40 m
MIMO	2 Tx | 3 Rx antennae	4 Tx | 4 Rx antennae	multiuser
Backward compatibility	Yes	Yes	Yes

Table 1: Most important features of IEEE 802.11n/ac/ax technologies

The IEEE 802.11 standard is basically a WiFi communication technology that operates in the bottom two layers of the OSI reference model. In our case, we will describe the key elements of the physical layer of the 802.11n/ac protocol, as this technology was available for us to measure wireless traffic. The task of the physical layer is to appear the bit sequence in its original, error-free state on the receiving node where the bit sequence was transmitted on the medium between the two communicating devices. In our case, the medium in question is the electromagnetic field.

In order to avoid eavesdropping and signal interference, and to increase fault tolerance, several transmission modes have been developed for data transmission in the electromagnetic field. 802.11n uses HT-OFDM (Higher Throughput Orthogonal Frequency Division Multiplexing) technology [17]. As is the case with other frequency division multiplexing, the given frequency spectrum is again divided into several smaller channels. The peculiarity of the OFDM is that these channels partially overlap each other, unlike conventional techniques, where there are smaller spare frequency bands having separating functionality between the channels [18].

Basically, the task of the media access and control (MAC - Medium Access Control) sub-layer is to reliably transmit frames to be transmitted even in noisy environments. Since the transmission medium is not at all as reliable as, say, on an optical line, "Best effort" transmission would not have been an expedient mechanism. Therefore, for the sake of reliability, the receiver always acknowledges the received frames. To conclude, the simplest frame transfer event looks like the following. The source station sends the desired data with a collision prevention strategy (CSMA/CA) then the receiving station acknowledges these. If the acknowledgment is not received, the sender retransmits the bit sequence previously transmitted by itself [19]. The structure of an elementary IEEE 802.11 frame is shown in the following figure.

FC	Dur./ID	Addr.1.	Addr.2.	Addr.3.	Seq.Cont.	Addr.4.	Data	FCS	
2	2	6	6	6	2	6	0-2312	4	bytes

Figure 1: Frame structure of IEEE 802.11 transmission technology

The FC (Frame Control) field contains the protocol version, the main and subtype of the frame, the distribution system bits, the fragment bit, which indicates whether the current frame is the last fragment of a longer frame. This field contains the send-retry bit, the bit to proclaim energy conservation, the bit to indicate more data. The latter indicates that at least one more frame is currently buffered at the base station for the mobile station. Other fields are the WEP encryption bit and the request defining bit, which indicates whether the given data frame is needed for a well-defined service [20].

The Duration/ID field has a duration length or an identifier function. In the first case, it specifies the time slice as much time the sending device needs to use the radio wave channel for transmitting the frame. In the second case, the mobile station identifies the receipt of its frame buffered by the base station.

The four address fields (Addr.1, ..., Addr.4) contain the physical addresses of the destination and source, as well as the recipient and sender addresses. Source addresses are unique, and destination addresses can be multicast/broadcast addresses. The Sequence Control field contains the sequence number and the fragment number. These can help you deal with communication disorder caused by the possible occurrence of duplicate frames.

The Data section contains the payload bytes. If the frame contains bytes of data type, this field occupies a significant part of the size of the radio frame [21].

FCS (Frame Check Sequence) enables error checking of the ordinary IEEE 802 standard. The receiver compares the received code number with the code value calculated from the frame and, if they are the same, records the receipt of the frame as error-free delivery. The possible discrepancy gives the error rate. At a large Hamming distance, the frame is damaged and therefore discarded, but can repair a one-bit error.

With this communication technology, it is not enough to talk about just one kind of frame. The “Type” field in the FC record allows you to distinguish three frame types: i) Management frames; ii) Control frames; iii) Data frames. There is also a “Subtype” field, which further subdivides the types into subtypes.

Binary values of Type field	**Binary values of Subtype field**	**Frame type**
00	0000 ... 1111	Management
01	0000 ... 0101	Control
01	0110	Extended control
01	0111 ... 1111	Control
10	0000 ... 1111	Data
11	0000 ... 1111	*Reserved*

Table 2: IEEE 802.11 protocol data element types and subtypes

From the viewpoint of the paper one of the dominantly important subtypes of management frames is the Beacon frame. Beacon frames are theoretically sent by base stations at periodic intervals, but in practice this period may sometimes vary slightly due to the base station’s occupancy due to the transmission of the current frame.

Type and subtype code	**Extended control frame**
0x0160	*Reserved*
0x0161	*Reserved*
0x0162	Poll
0x0163	Service Period Req
0x0164	Grant
0x0165	DMG Clear-to-send
0x0166	DMG Denial-to-send
0x0167	Grant Ack

Type and subtype code	**Extended control frame (cont.)**
0x0168	Sector Sweep
0x0169	Sector Sweep F.back
0x016a	Sector Sweep Ack
0x016b	*Reserved*
0x016c	*Reserved*
0x016d	*Reserved*
0x016e	*Reserved*
0x016f	*Reserved*

Table 3: Extended control frame types of the IEEE 802.11 protocol

Beacon frames advertise basic information about the base station to other mobile terminals that want to use this WiFi service. From the viewpoint of the paper less important but noteworthy subtypes are the Authentication and Deauthentication frames, which serve to authenticate terminals. Like the Beacon frame, Probe Request and Response advertise base station information, such as supported data transmission rates [22].

The control frame also has subtypes. Some of these frames are used to prevent collision. The dialogue with the control frames of subtypes RTS (Request to Send) and CTS (Clear to Send) determines whether the entity wishing to send can start sending the data frame. The potential sending terminal requests transmission permission with RTS control frame from the base station. If it receives a CTS control frame response, it can begin transmitting its data frame. The ACK (Acknowledge) subtype acknowledges the received data frame to the sender [23]. It is important to note that expanded control frames in further subdivision identify special frame types of newer technologies (IEEE 802.3 ac/ad) too. Because the FC field is 16 bits, the last hexadecimal digit of the 0x16X code identifying special control frames is encoded by changing the function of the other bits of the FC field.

The subtypes of the data frame differ only in those that also contain smaller control information for the transmitted data, such as an ACK (Acknowledgment) acknowledgment, which confirms the successful transmission of the previous data frame [24].

4. Neural network stick for fast data processing

The Intel Neural Compute Stick 2 (Intel NCS2) is a plug-and-play device that looks like a standard USB drive designed to execute inferences using neural network and input data in a quickly and energy efficiently way. It is also suitable for real-time inference. It requires no cloud infrastructure and can be used with low-powered devices such as the Raspberry Pi 3 and other ARM host devices. It can be connected to USB 3.0 Type-A compatible connectors. Inside the device is an Intel® Movidius ™ Myriad ™ X Vision Processing Unit (VPU) that contains 16 processing cores and a dedicated deep neural network hardware accelerator. Multiple NCS 2s can be used to infer at the same time, and can even be combined with CPU and GPU. NCS 2 is primarily used to deploy vision-oriented solutions using convolutional neural networks (CNNs).

The general workflow as shown in the Figure 2. is the following: First, the raw data is preprocessed, resulting in the training set, validation set and test set. These sets are used to train the model. The model is trained using frameworks or formats supported by Intel OpenVINO software, and this process should be done on the development hardware, not on the stick.

Briefly, OpenVINO is a cross-platform (Windows, Ubuntu, macOS, etc.) software that enables deep learning inferences, supports various heterogeneous execution across an Intel Integrated Graphics, Intel FPGA, NCS 2, Intel CPU, etc., supports well-known deep learning frameworks and formats (TensorFlow, MXNet, Caffee, ONNX, etc.) and includes optimized calls for computer vision libraries such as OpenCV, OpenCL and OpenVX. Model Optimizer and Inference Engine API are part of OpenVINO. Once the model is complete, it must be translated with the Model Optimezer into the so-called Intermediate Representation (IR) format. This format consists of two files, .xml describes the network topology, and .bin contains the network weights and bias values. Finally, the user application uses IR and infers new data both using the Inference Engine API, which is executed on Intel NCS 2, and outputs the result.

Some real-time applications need to process large amounts of data in a short amount of time, which in our case means that our application must infer for hundreds of incoming frames in less than 1 second. This requires a powerful computing

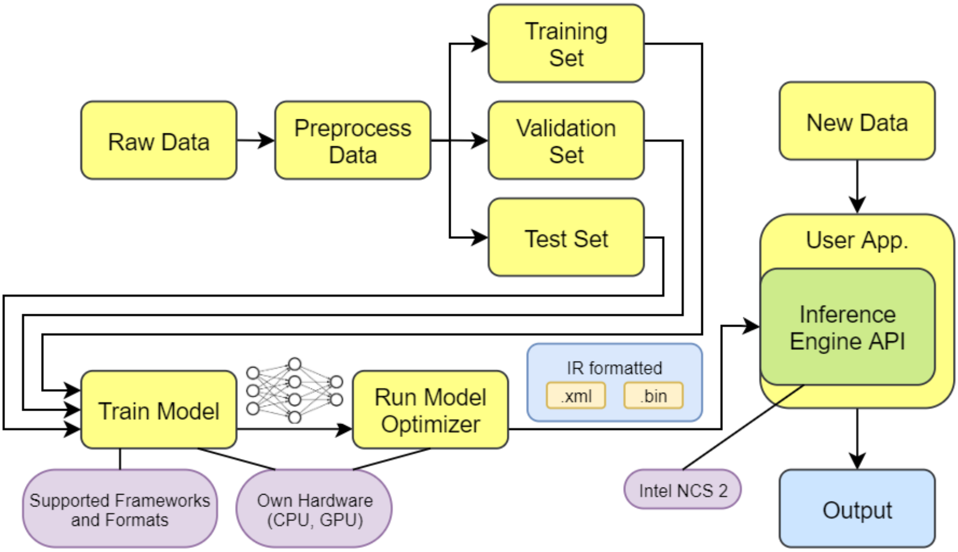

Figure 2: General workflow to apply Intel NCS 2

device, such as the Intel NCS 2, which is optimized for neural network inferences. Experiments have already been made by us to use this device for enhancement of RNN calculations, but issues have been detected while running the Inference Engine. Further in-depth research and software development is in progress to eliminate these compatibility problems concerning neural network topologies supported explicitly by the Openvino.

5. Deep learning analysis of IEEE 802.11n/ac/ax traffics

There were captured WiFi frame streams at a four floor building with 26 base stations managed by a common controller. The measurements were executed step by step in 56 physical dispersed in space in different time moments. Measured values were: Frame Length (B), Frame Type/Subtype, Frame Arrival Time (sec) and Received Signal Strength Indicator - RSSI (dBm). Duration of one measurement was in the scale of ten seconds. Sampling of the WiFi frames was made with Wireshark protocol analyser software running on a notebook. It was activated deep mode of the Npcap v0.9984 driver to capture IEEE 802.11 format of all the frames instead of the basic mode showing just IEEE 802.3 frames. In this way all the management, control and data type frames transmitted on the wireless channel were captured by the software. Based on the investigation results described in papers [25, 26, 27, 28] we oriented to use recurrent neural networks for analysis of the WiFi channel usage efficiency. By channel usage efficiency we mean ratio of data bytes to the total number of bytes belonging to all frame types. This efficiency depends on the channel quality because noisy or congested radio channel requires more control frames and even retransmission of the data frames.

As an example frame length of different types captured at sampling point no. 23 can be seen in Figure 3. In this case majority of the frames were Beacon management (34.5%) and Request-to-Send control (17.69%) frames. It should be noted that the IEEE 802.11n/ac/ax communication mechanism use intensive management and control frame transmissions to supervise nodes and deliver data type frames between the wireless nodes. Intensity of the unidentified type frames was 0.61% which is relatively small but not negligible and are sent by new smart phones without access point responses. We enrolled such frames in a separated type named Other.

Because it exists correlation between the intensity of different types of the frames we grouped subtypes of the frames in four: management (MNG), control (CTR), data (DAT) and other (OTH). We split each ten second measurement in sequences having total duration of one second. Each sequence belongs to the event series of frame arrivals during one second sampling time interval. The sequence has three predictor variables: v1: Frame Length (B), v2: Frame Interarrival Time (sec) and v3: Received Signal Strength Indicator - RSSI (dBm). The interarrival time is the time difference between sampling times of two consecutive frames. The number of events is given by the number of frames, n.

The sequence is characterized by the four metrics: relative amount of p1: MNG, p2: CTR, p3: DAT and p4: OTH bytes transmitted on the radio channel during the sequence time interval. We have p1 + p2 + p3 + p4 = 100% in reality.

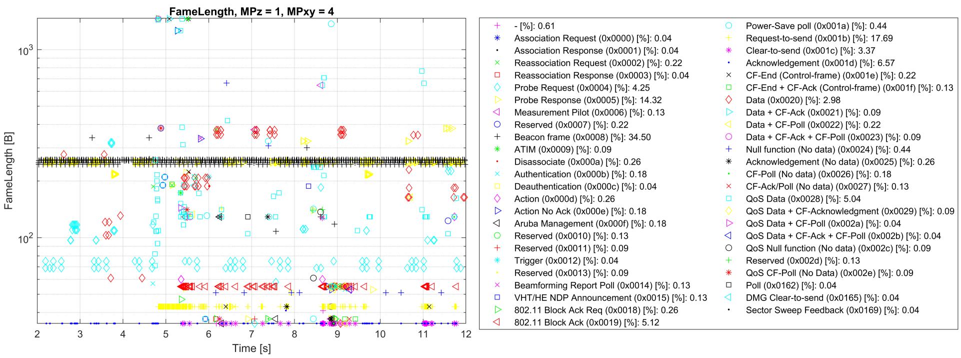

Figure 3: Measured frames with type/subtype classes in sampling point no. 23. Percentages in the right represent relative intensity of the byte subtypes during the initial ten second measurement interval.

In this way we got N = 591 sequences, each having three predictor variables: v1, v2, v3 and four score metrics (ratios): p1, p2, p3, p4. It is obvious that the sum of these four metrics gives 100% for each sequence. The length of predictor variables is n_i, $i = 1, 2, \dots, N$. Because the number of frames during the sequence sampling time varied in function of the WiFi traffic intensity, length n_i is not constant and takes values in the range of [22, 550]. Ratios p1, p2, p3, p4 of the whole data set are given on Figure 4. The initial sequence set with N elements is divided in three parts: train set (300 elements), validation set (150 elements) and test set (141 elements). Number of frames of the test data set is given in Figure 5.

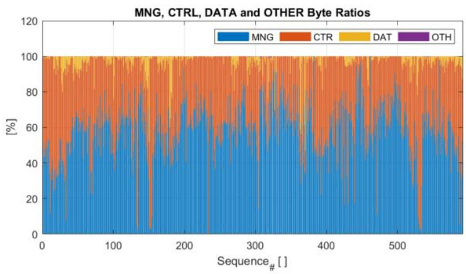

Figure 4: Ratios of MNG, CTR, DAT and OTH of the sampled sequences (p1, p2, p3, p4).

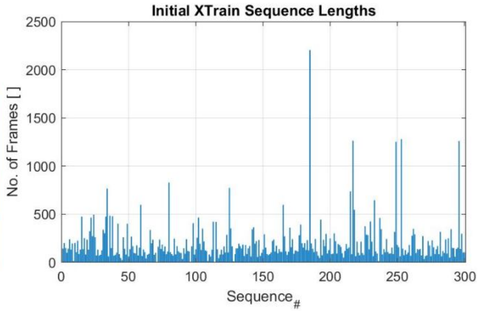

Figure 5: Length of initial XTrain sequences, n_i. Ordering is used to train the RNN.

Parameters of the tensor data analysed with different recurrent neural networks are given in Table 4. For each neural network learning it was used Adaptive Moment Estimation (Adam) optimizer with GradientDecayFactor = 0.9, SquaredGradient-DecayFactor = 0.9990 and InitialLearnRate = 0.01. Topology of these networks are including Long-Short Term Memory (LSTM) or bidirectional LSTM (BiLSTM) layers (see Figure 6 and 7). For faster convergence two fully connected (FC) layers were used. The first layer has 512 number of classes and the second layer just four because we are predicting scores p1, p2, p3 and p4 of the test dataset. Number of hiden units in LSTM or BiLSTM layers was set to be 100 or 200.

Parameter	**RNN1**	**RNN2**	**RNN3**	**RNN4**	**RNN5**	**RNN6**	**RNN7**	**RNN8**
RNN Type	LSTM	LSTM	LSTM	BiLSTM	BiLSTM	BiLSTM	BiLSTM	BiLSTM
miniBatch-Size	50	100	300	50	50	100	100	100
maxEpochs	360	180	180	360	360	180	180	360
LSTM numHidde-nUnits	100	100	100	100	200	100	200	100
FC num-Classes	512	512	512	512	512	512	512	512
Learning time [sec]	348	107	84	378	137	154	154	271

Table 4: Parameters of the used recurrent neural networks

Table 5. shows correlation coefficients of the test predicted scores. It can be

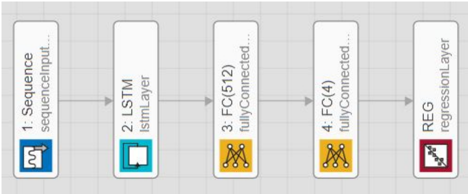

Figure 6: NN topology A
(RNN1, RNN2, RNN3)

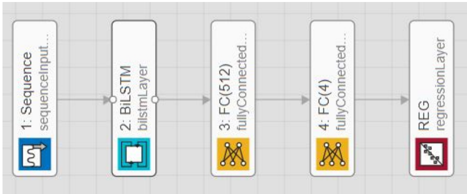

Figure 7: NN topology B
(RNN4, . . . , RNN8)

stated that for the sampled WiFi channel sequences BiLSTM has better performance than LSTM. Having information about the past and the future sequences of the traffic in WiFi channel helps to train better the neural network. To have best performance of the RNN needs at least 200 hidden neurons of the recurrent layer. Computation capacity has impact on the training time. To decrease the computation time dedicated hardware tool, like compute stick is required.

Best neural network to predict the IEEE 802.11n/ac channel usage efficiency was found to be RNN5. Prediction with 0.70 correlation is based on frame size, interarrival time of the frames and the received signal strength intensity.

Coeff. of correlation	**RNN1**	**RNN2**	**RNN3**	**RNN4**	**RNN5**	**RNN6**	**RNN7**	**RNN8**
R: MNG	0.00	0.10	0.16	0.42	**0.56**	0.52	0.53	0.53
R: CTR	0.04	0.12	0.01	0.41	**0.52**	0.52	0.49	0.49
R: DAT	**0.26**	**0.20**	**-0.46**	**0.55**	**0.70**	**0.45**	**0.54**	**0.39**
R: OTH	0.11	0.09	0.08	0.01	0.07	0.04	0.10	0.12

Table 5: Correlation between the test and predicted scores

Figure 8 shows the real and predicted scores of the channel usage efficiency. Considerable correlation exists between the real and predicted usage efficiency of the IEEE 802.11 channel by the recurrent neural network. Not only the data but the management and control bytes sent on the channel are predicted with reasonable accuracy.

6. Conclusions and future work

In this paper we analysed IEEE 802.11n/ac/ax technology behaviour based on neural network modelling. Based on the special features of the wireless medium access communication mechanism there were captured all the management, control and data frames transmitted on the radio channel. It is settled that majority of the wireless bandwidth is used for management and control byte transmissions and just 50% or less of the channel capacity is allocated to transmit data bytes. The proposed neural network contains recurrent layers and the learning process is supervised. There were compared usability of LSTM and BiLSTM layers for

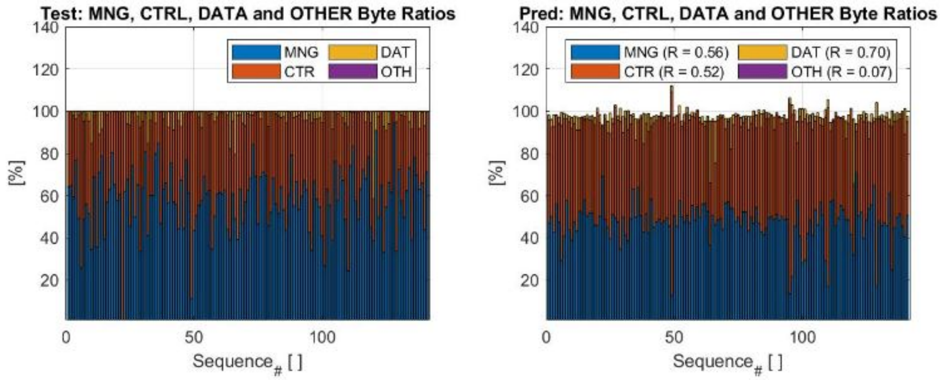

Figure 8: Best final scores of the used recurrent neural networks.
p1, p2, p3, p4 for RNN5. (Left: Test ratios; Right: Predicted ratios)

prediction of the WiFi channel usage efficiency. It was found that BiLSTM has better performance to prognoses transmitted byte ratios belonging to management, control and data classes. Possible continuation of this research may be oriented to evaluate other topologies of the deep neural network.

Acknowledgements. This work was supported by the construction EFOP-3.6.3-VEKOP-16-2017-00002. The project was supported by the European Union, co-financed by the European Social Fund. and EFOP-3.6.3-VEKOP-16-2017-00002, respectively. This paper was supported also by the FIKP-20428-3/2018/FEKUTS TRAT project of the University of Debrecen, Hungary and by the QoS-HPC-IoT Laboratory.

References

[1] Battiti, R., Le, N.T., Villani, A., Location-aware Computing: A Neural Network Model for Determining Location in Wireless Lans, *Technical Report. University of Trento*, (2002).

[2] Ahmad, U., Gavrilov, A., Nasir, U., Iqbal, M., Cho, S.J., Lee, S., In-building localization using neural networks, *2006 IEEE International Conference on Engineering of Intelligent Systems. IEEE*, (2006), 1–6.

[3] Shareef, A., Zhu, Y., Musavi, M., Localization using neural networks in wireless sensor networks, *Proceedings of the 1st International Conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering)*, (2008), 4.

[4] Laoudias, C., Eliades, D.G., Kemppi, P., Panayiotou, C.G., Polycarpou, M.M., Indoor localization using neural networks with location fingerprints, *Artificial Neural Networks–ICANN 2009. Springer*, (2009), 954–963.

[5] Taok, A., Kandil, N., Affes, S., Neural networks for fingerprinting-based indoor localization using ultra-wideband, *Journal of Communications* 4 (4), (2009), 267–275.

[6] Altini, M., Brunelli, D., Farella, E., Benini, L., Bluetooth indoor localization with multiple neural networks, *2010 5th IEEE International Symposium on Wireless Pervasive Computing (ISWPC). IEEE*, (2010), 295–300.

[7] Li, S., Liu, B., Chen, B., Lou, Y., Neural network based mobile phone localization using bluetooth connectivity, *Neural Computing Applications* 23 (3–4), (2013), 667–675.

[8] Ju, S., Evans, J.B., Scalable cognitive routing protocol for mobile ad-hoc networks, *2010 IEEE Global Telecommunications Conference (GLOBECOM 2010)*, IEEE (2010), 1–6.

[9] Barabas, M., Boanea, G., Dobrota, V., Multipath routing management using neural networks-based traffic prediction, *EMERGING 2011, The Third International Conference on Emerging Network Intelligence*, (2011), 118–124.

[10] Zhi-yuan, L., Ru-chuan, W., Jun-lei, B., A multipath routing algorithm based on traffic prediction in wireless mesh networks, *Fifth International Conference on Natural Computation, 2009. ICNC'09*, Vol. 6. IEEE (2009), 115–119.

[11] Baldo, N., Zorzi, M., Learning and adaptation in cognitive radios using neural networks, *5th IEEE Consumer Communications and Networking Conference, 2008. CCNC 2008. IEEE*, (2008), 998–1003.

[12] Katidiotis, A., Tsagkaris, K., Demestichas, P., Performance evaluation of artificial neural network-based learning schemes for cognitive radio systems, *Comput. Electr. Eng.* 36 (3), (2010), 518–535.

[13] Tumuluru, V.K., Wang, P., Niyato, D., A neural network based spectrum prediction scheme for cognitive radio, *2010 IEEE International Conference on Communications (ICC). IEEE*, (2010), 1–5.

[14] Moustapha, A.I., Selmic, R.R., Wireless sensor network modeling using modified recurrent neural networks: application to fault detection, *IEEE Trans. Instrum. Meas.* 57 (5), (2008), 981–988.

[15] Capka, J., Boutaba, R., Mobility prediction in wireless networks using neural networks, *Management of Multimedia Networks and Services*, (2004), Springer, 320–333.

[16] Stegmayer G., Chiotti O., Neural Networks applied to wireless communications, *Bramer M. (eds) Artificial Intelligence in Theory and Practice. IFIP AI 2006. IFIP International Federation for Information Processing*, Vol 217. (2006), Springer, Boston, MA.

[17] IEEE Std 802.11n-2009 — Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.

[18] Casas, E. F.–Leung, C., OFDM for Data Communication Over Mobile Radio FM Channels – Part I: Analysis and Experimental Results, *IEEE Transactions on Communications*, (May 1991).

[19] IEEE Wireless LAN Edition, A compilation based on IEEE Std 802.11™-1999 (R2003) and its Amendments, *IEEE*, (2003).

[20] IEEE Std.802.11TM-2007, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", *IEEE Computer Society*, (June 2007).

[21] Colbach, G., The WiFi Networking Book: WLAN Standards: IEEE 802.11 bgn, 802.11n, 802.11ac and 802.11ax, (June 2019).

[22] Perahia, E., Stacey, R., Next Generation Wireless LANs, (2013).

[23] Soni, N., Wireless Networking: Introduction to Bluetooth and WiFi.

[24] Gast, M. S., 802.11 Wireless Networks: The Definitive Guide, Second Edition, (2005).

[25] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T., Caffe: Convolutional architecture for fast feature embedding, *Proceedings of the ACM International Conference on Multimedia*, (2014), 675–678.

[26] Sharma, G., Kalena, P., Malde, N., Nair, A., Parkar, S., Visual Image Caption Generator Using Deep Learning, *2nd International Conference on Advances in Science & Technology (ICAST)*, (2019).

[27] Johnson, J., Karpathy, A., Fei-Fei, L., Densecap: Fully convolutional localization networks for dense captioning, *IEEE CVPR*, (2016).

[28] Karim, F., Majumdar, S., Darabi, H., Insights into LSTM Fully Convolutional Networks for Time Series Classification, *IEEE Access*, (2019).