Detection of traffic patterns in the radio spectrum for cognitive wireless network management

Detection of traffic patterns in the radio spectrum for cognitive wireless network management

Miguel Camelo, Tom De Schepper, Paola Soto, Johann Marquez-Barja, Jeroen Famaey and Steven Latré
University of Antwerp - imec, IDLab, Department of Mathematics and Computer Science,
Sint-Pietersvliet 7, 2000 Antwerp, Belgium

***Abstract*—Dynamic Spectrum Access allows using the spectrum opportunistically by identifying wireless technologies sharing the same medium. However, detecting a given technology is, most of the time, not enough to increase spectrum efficiency and mitigate coexistence problems due to radio interference. As a solution, recognizing traffic patterns may lead to select the best time to access the shared spectrum optimally. To this extent, we present a traffic recognition approach that, to the best of our knowledge, is the first non-intrusive method to detect traffic patterns directly from the radio spectrum, contrary to traditional packet-based analysis methods. In particular, we designed a Deep Learning (DL) architecture that differentiates between Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) traffic, burst traffic with different duty cycles, and traffic with varying rates of transmission. As input to these models, we explore the use of images representing the spectrum in time and time-frequency. Furthermore, we present a novel data randomization approach to generate realistic synthetic data that combines two state-of-the-art simulators. Finally, we show that after training and testing our models in the generated dataset, we achieve an accuracy of $\geq$ 96 % and outperform state-of-the-art methods based on IP-packets with DL.**

***Index Terms*—Traffic recognition, Cognitive Radio, Spectrum Management, Deep Learning, Convolutional Neural Networks**

I. Introduction

The number of connected devices has surpassed 18 billion and will reach 28.5 billion by 2022 [1]. Similarly, the number of different available wireless technologies has already increased significantly, and this growth will continue with the releases of new IEEE 802.11 (Wi-Fi) standards (e.g., IEEE 802.11ax) and Long-Term Evolution (LTE) technologies (e.g., LTE-Unlicensed (LTE-U)) [2], [3]. These different wireless technologies coexist next to each other at identical or overlapping physical locations, often competing for the same finite and limited radio spectrum resources. This situation, combined with the increasing demands in the spectrum usage, can potentially lead to significant Quality of Service (QoS) degradation and performance loss [4]. This is, among others, the case for LTE-U and Wi-Fi in the 5 GHz frequency band, as well as for Wi-Fi, ZigBee, and Bluetooth in the 2.4 GHz band. Consequently, there is a need for intelligent and more efficient use of spectral resources [2], [5].

The principle of Cognitive Radio (CR) has been introduced to allow the coexistence of different technologies and networks over the same spectrum [5]. One of these mechanisms is Dynamic Spectrum Access (DSA) [5], which uses the sensing capability of the CR systems to identify if a, possibly unknown, technology is accessing the same spectrum and then take appropriate measures to mitigate performance degradation due to interference. However, detecting a given technology is often not enough to increase spectrum efficiency, as no information is provided about the exact spectrum usage within the transmitted signal of the technology in question.

One way to solve this problem is by complementing the identification of technologies with information about the network's traffic patterns. Nonetheless, traffic identification is traditionally performed by intrusive methods using traffic analysis at packet level such as Deep Packet Inspection (DPI) and IP packet traces [6], [7], which rely on having a monitoring device that is part of the network. In contrast, we believe that a better approach is to learn the traffic behaviour of other networks directly by observing the spectrum and then offload its traffic during the interference-free time-slots given the learned pattern. This allows us to significantly increase spectrum efficiency, especially in places with a dense and overlapping presence of wireless networks.

To this extent, we present a Deep Learning-based approach that is capable of detecting different transport protocols, traffic patterns, and different Transmission (TX) rates. To the best of our knowledge, this is the first approach that performs traffic recognition directly using spectral data. The contributions of this paper are fourfold. First, we propose a novel data randomization approach to generate large amounts of (synthetic) data, by combining two state-of-the-art simulators, namely the NS-3 network simulator and the WLAN Matlab toolbox. This allows us to generate large realistic datasets efficiently, which can be used to train more robust models capable of coping with the uniqueness of different wireless environments. Second, we present three different models, which are based on a Convolutional Neural Network (CNN) architecture, to recognize i) TCP and UDP traffic, ii)burst traffic with different duty cycles, and iii) different types of TX rates. Third, we explore and compare two different approaches to represent the sensed spectrum as the input of our models: time and time-frequency image representations. Finally, we compare our proposal with a state-of-the-art DL-based approach using raw IP-packet.

The remainder of this paper is structured as follows. We present the related works in Section II. The data collection framework is introduced in Section III, and our CNN architecture and traffic recognition models are presented in Section IV. Finally, we show the evaluation results of the different models in Section V and conclusions are drawn in Section VI.

II. Related work

Traditionally, traffic recognition takes place at gateways or routers in a network aiming to identify individual traffic streams entering or exiting the controlled network environment. The typical use cases are network management (e.g., providing QoS or billing) by Internet Service Providers (ISPs) and intrusion detection in the area of network security. Two typical approaches exist: DPI and methods that are based on Internet Protocol (IP) packet traces. Historically, DPI has long been the default approach where information is extracted from the headers and payload of individual packets, typically in a rule-based system. While DPI methods are very accurate, they require much computational power, are very intrusive (i.e., towards privacy), and often require manual signature maintenance, and can not always cope with the encryption of many advanced end-to-end services [6].

As an alternative to DPI, new methods based on statistical and Machine Learning (ML) have been proposed. These methods do not require packet inspections but are based on captured packet traces, typically at the IP level [7]. They are based on feature selection of traffic flow statistics like packet sizes, flow duration, inter-packet times, source and destination ports, and IP addresses. Lately, also DL approaches have been proposed. For instance, authors in [8] propose a combination of a stacked Autoencoder, which uses byte representation of the IP packets as input, to identify applications with an accuracy of 98 % and traffic characteristics with a precision of 93 %. However, these approaches do not capture the time-dependent features of the traffic, require a large amount of data for training, and the input data pre-processing adds a considerable overhead that makes these approaches not feasible for real-time systems.

Within the domain of CR, the detection and classification of wireless radio signals are important to optimize spectrum usage. Recently, DL techniques are mostly applied to identify different modulation schemes and radio technologies (e.g., through their channel access methods). To this extent, Schmidt et al. use a CNN architecture to discriminate, with an accuracy of 95 %, between 19 different variants of modulation types and symbol rates within the 2.4 GHz band [9]. Similarly, Kulin et al. and Camelo et al. have shown how CNNs significantly outperform expert learning systems for the task of classifying both known [10] and unknown radio technologies [11] using raw In-phase and Quadrature (IQ) samples.

Finally, in the domain of data generation to support machine learning applications for cognitive radio, Davaslioglu et at. have shown the capabilities of generating additional synthetic training data to improve classifier accuracy via data augmentation and domain adaptation using Generative Adversarial Networks (GANs) [12]. As a result, the use of augmented training data increased the classifier accuracy significantly and provided robustness under spectrum conditions changes.

III. Data generation framework

In Section II, we have mentioned how traditional traffic recognition approaches are typically operating with wired packet data. As CR approaches are focusing on technology and modulation recognition, no accurately labelled datasets for traffic recognition in the wireless spectrum are currently available. Furthermore, as the characteristics of wireless networks (e.g., interference, latency) tend to differ significantly across different environments, training a model in a general fashion is not straightforward and requires large volumes of training data. However, recent breakthroughs in robotics have shown how models trained purely with synthetic data are still able to be deployed in a real-life setting with very high accuracy.

While simulators do not perfectly resemble the complex real-world, they do allow collecting massive amounts of data in an easy, safe, and reliable manner. By using a persevering form of randomization, it has been proved that this so-called reality-gap can be bridged. This technique, called Domain Randomization, allows training learning models based only on the simulated data by randomizing all non-essential aspects of the simulation [13]. Besides, as no simulator currently exists that extensively covers all aspects of wireless devices, we opt for two state-of-the-art solutions, each covering different parts of the network stack. We use the discrete-event network simulator NS-3 (ver. 3.29) to include the higher layers of the stack. Regarding the lower layers, we consider the WLAN toolbox (ver. 2.0) integrated with Matlab R2018b, in particular, to address the physical layer, as the toolbox can generate wireless signals and Radio Frequency (RF) spectrograms.

The overall architecture of the framework is depicted in Figure 1. As can be seen, there are three main components. First, the experiment generator receives as input a scenario description file in JSON format. This file contains, for several selected parameters, a list of values to be used in the simulations. The considered parameters are network topology (number of the nodes, their locations, and mobility patterns), traffic descriptions (number of flows, transport protocol, traffic patterns, and rates), and Wi-Fi related parameters (Modulation and Coding Scheme (MCS), channel width, standard, and guard interval). The experiment generator (1) generates an experiment setup for each unique combination of parameter values. Next, the NS-3 simulator (2) executes each configuration individually and creates a log file containing all transmitted packets and the information needed to generate the transmitted waveform in Matlab. This log can be seen as a regular packet dump (PCAP), but augmented with the specific type of packet (e.g., beacon, association request, data), the technology depended on information (e.g., for Wi-Fi: STBC, PSDU length, AMPDU), and the MCS value.

Afterwards, Matlab (3) performs the following steps: first, one MAC packet is generated per each line of the trace file in combination with the information provided by the scenario description file. This MAC packet is used to create a waveform, i.e. the IQ samples, that is compliant with the 802.11 standard. After each packet generation, the IQ samples are stored in binary format in a file. Second, the captured IQ samples are augmented by passing them through both a modelled fading channel (according to the 802.11 standard) and an Additive White Gaussian Noise (AWGN) channel. The latter adds white Gaussian noise to the signal, with a signal to

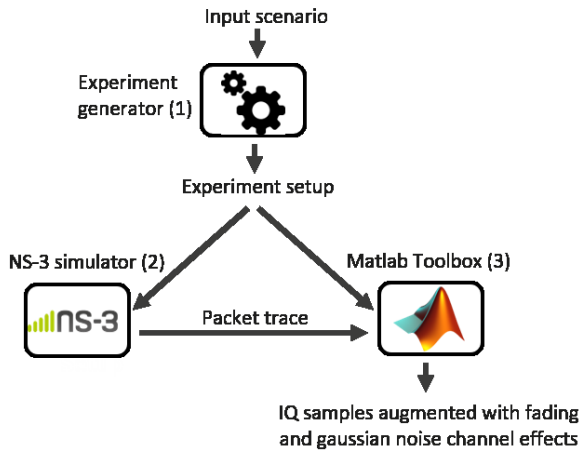


Fig. 1. Architectural overview of data generation framework

noise ratio (SNR) varying from 0dB to 30dB in steps of +3dB, as such creating different circumstances for each IQ sample, i.e. randomizing the spectrum conditions.

One of the main properties of traffic recognition using spectral data is that traffic features are visible in periods longer than hundreds of milliseconds. This contrast to other related tasks, i.e. technology and modulation identification, where features used to discriminate between classes are visible in shorter periods (at most hundred of microseconds) [10]. For example, the modulation recognition can have an input of 128 IQ samples and represents 128 us of the signal at a sampling rate of 1 MSamp/sec [14]. At the same sampling rate, representing 0.5 ms of spectrum data for traffic recognition would require 0.5 M of IQ samples. However, this input size makes impossible the implementation of ML models due to the enormous computational requirements to train and run them.

To have a broader view of the spectrum in time, while providing a small representation of the input data, we perform the following transformation: given a time window w and raw spectrum data as input, the data collected during w seconds is plotted in time (amplitude of the IQ samples), and time-frequency (by applying the Short-Time Fourier Transform (STFT) on the IQ samples) and saved as images. In this paper, we capture spectrum data during $w = 0.5s$, which generates 10 M floating points at a sampling rate of 20 Msps, and these points are interpolated to get 256 values with a magnitude in the range $[0, 1]$ in steps of 0.0052 units (192 values). We perform this interpolation using the compression-filtering algorithm of png graphics file-format. Each resulting image requires between 3.3 KB and 114 KB of storage, which is around 1% of the required storage for the raw IQ samples.

To generate the labels of the input data, we focus on the recognition of 3 different properties of the generated traffic: the transport protocol (TCP versus UDP), the traffic pattern (constant versus three types of burst traffic with a duty cycle of, respectively, 25, 50, or 75 %), and the transmission rate of the generated traffic (100 Kbps, 1 Mbps, 10 Mbps, or 50 Mbps).

IV. Traffic Recognition Models

Let $X = \{x_1, \ldots, x_N\}$ and $Y = \{y_1, \ldots, y_N\}$ be the sets of N examples and their corresponding labels, respectively, where $x_i \in X$ and $y_i \in Y$ for all $i \in [N] := \{1, 2, \ldots, N\}$. In Supervised Learning (SL), the goal is to learn a mapping from X to Y given a training set of pairs (x_i, y_i), where y_i is the label of the ith example x_i. In other words, given the sets X and Y, SL tries to find a function f such that $y = f(x)$.

In traffic recognition using spectrum data, the problem is to find a function f that given a representation of the spectrum $x_i \in X$ (e.g. raw IQ samples or STFT of the IQ samples) it is able to predict $y_i \in Y$. In this case, Y is a set of labels that represent properties of the traffic, such as the transport protocol, traffic pattern of the application, or the transmission rate. In this paper, we use DL architectures, and specifically Deep Neural Network (DNN) models to build f by a combination of many simpler functions that are connected in a hierarchically way. In this section, we give a brief description of the employed approaches to recognize the generated synthetic data obtained by the framework described in section III.

A. Image-based Deep CNN architecture for traffic recognition using spectral data

To exploit the new representation of the input data as an image, we design and implement three DL models based on a CNN architecture. Under this assumption, the forward function, i.e., moving the data from input to output through the neural network, is more efficient and its computational complexity for learning is lower than traditionally DNN based on fully-connected layers. Note that these kinds of DNNs have been shown to perform well in tasks such as modulation recognition with either raw data of the spectrum as input (e.g., IQ or Fast Fourier transform (FFT) samples) or the image representation of that raw data [14], [15].

We design a baseline architecture that is shared by all the three models. It is composed of three Convolutional (Conv) layers with Rectified Linear Unit (ReLU) activation function, each one followed by a Maxpooling, batch normalization, and a Dropout layer for regularization. After the Conv layers, two final fully-connected layers, one with ReLU and the final one with a soft-max activation function, were added for classification. Figure 2 shows an overview of the resulting architecture and Table I and Table II the parameters used in each layer per model.

Some specific parameters for the optimizer, e.g., the type of optimizer, learning rate and batch size, and for each layer, e.g., the number of filters, pool size, and dropout rate, and optimizer training, were determined using hyperparameter swapping. In total, the CNN models have 25.2 million trainable parameters. We trained the models during 100 epochs using the Adam optimizer with a learning rate of 0.0005, cross-entropy loss function [16], and mini-batches of 128 RGB images, which were re-sized to [height,width] = [196,256] pixels due to memory constraints in the GPU. We implemented our model using the Keras library and TensorFlow as the back-end.

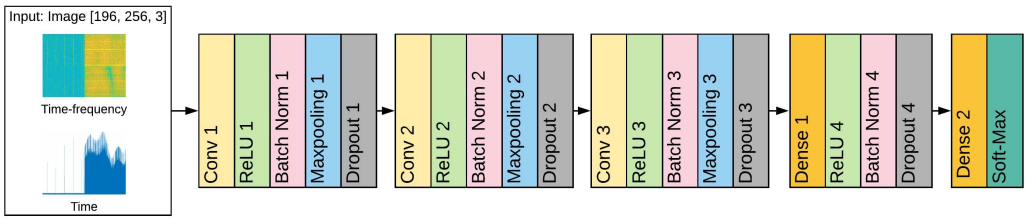


Fig. 2. Image-based CNN architecture for traffic recognition

TABLE I
SHARED HYPERPARAMETERS AMONG ALL THE MODELS

Source	Model	Domain	Task	Filters	Kernel Size	Pool Size (all MaxPool layers)	Dense 1 Neurons	Dense 2 Neurons	Dense 3 Neurons
This paper	**CNN 1**	Time	Traffic Pattern	Conv 1 : 16 Conv 2 : 32 Conv 3 : 64	Conv 1 : 5 x 5 Conv 2 : 5 x 5 Conv 3 : 5 x 5	2 x 2	512	4	N/A
	CNN 2	Time-Frequency							
	CNN 3	Time	Transport Protocol					2	
	CNN 4	Time-Frequency							
	CNN 5	Time	Transmission Rate					4	
	CNN 6	Time-Frequency							
Lotfollahi et al. [8]	**CNN 7**	Packet	Traffic Pattern	Conv 1: 200 Conv 2 : 200	Conv 1 : 4 x 4 Conv 2 : 5 x 5		200	100	50
	CNN 8	Packet	Transport Protocol						
	CNN 9	Packet	Transmission Rate						

TABLE II
SPECIFIC DROPOUT RATE PER IMAGE-BASED MODEL

Dropout Rate	CNN 1	CNN 2	CNN 3	CNN 4	CNN 5	CNN 6
Layer 1	0,1	0,6	0,7	0,6	0,7	0,5
Layer 2	0,3	0,5	0,5	0,3	0,4	0,4
Layer 3	0,6	0,5	0,5	0,3	0,4	0,4
Layer 4	0,6	0,3	0,3	0,1	0,2	0,2

B. Deep CNN for traffic recognition using raw IP packets

As mentioned in Section II, traditional approaches on traffic recognition are based on packet inspection. However, those techniques are invasive and do not cope well with encrypted traffic. Thus, following a more traditional approach, the authors of [8] propose a state-of-the-art CNN that performs identification of real-life network traffic. In this paper, we use their proposed architecture to implement a packet-based model as a baseline for our novel spectral-based model. The model consists of two consecutive Conv layers, followed by a pooling layer. Then, the resulting output is supported by a 3-layered fully connected network. Batch normalization combined with dropout layers (0.05) were used after each Conv and fully-connected layer, ReLU was used as the activation function. Lastly, a softmax classifier is used.

The CNN architecture is summarized in Table I. The model was trained during 300 epochs using Adam optimizer with a learning rate (0.001) and categorical cross-entropy as loss function with the PCAPs that were generated in the second phase of the framework described in Section III.

V. RESULTS AND DISCUSSION

In this section, we present and analyze the performance of our two proposed image-based DL models (CNN 1-6), and we compare them against the state-of-the-art DL model based on raw IP packets (CNN 7-9).

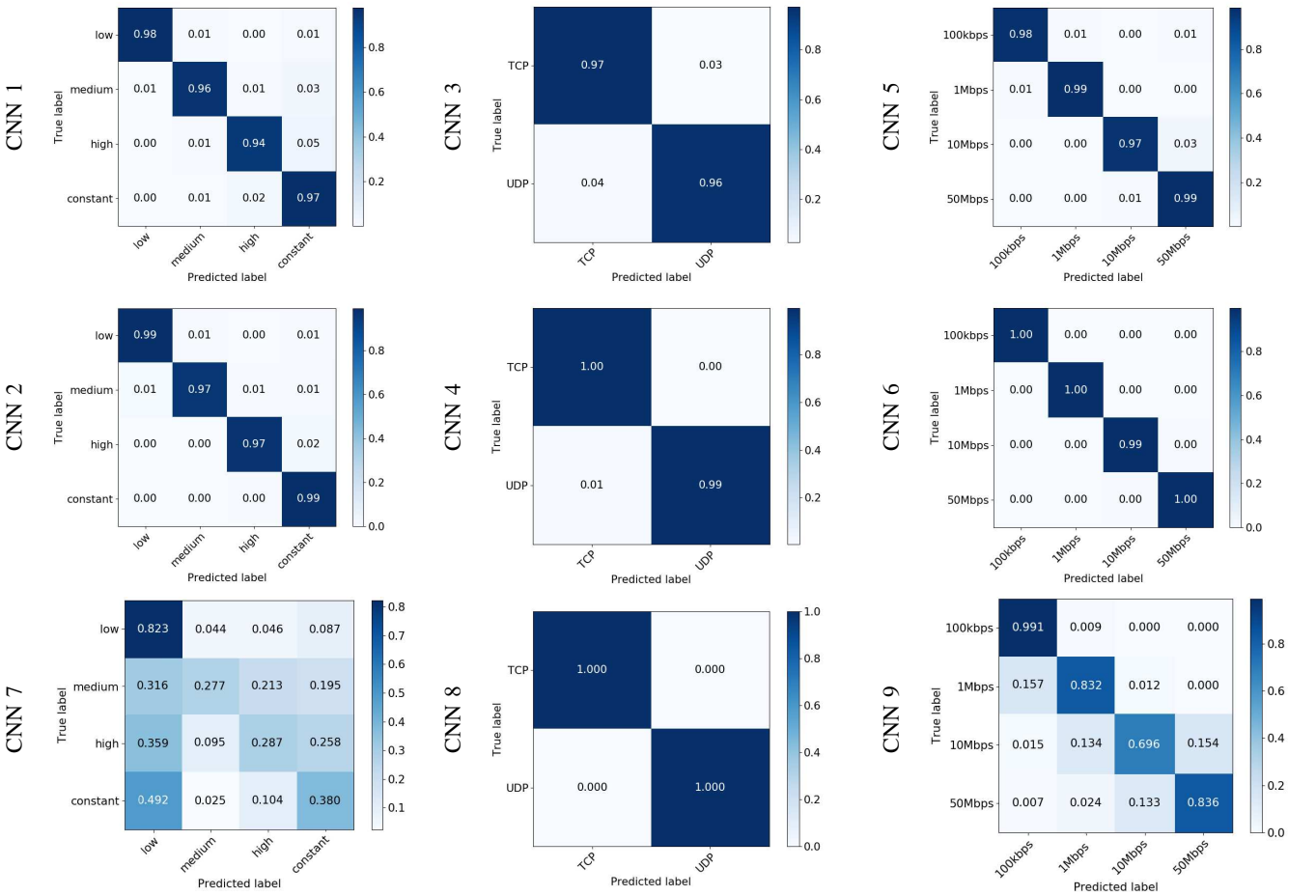
A. Dataset description

To evaluate the designed models, we use the data collection framework presented in Section III to generate the training, validation, and test datasets used for all experiments presented in this section. As a topology, we assumed one station connected to a Wi-Fi access point (AP) at a distance of 5 m. A single flow of traffic was transmitted for 5 s between the two devices with the varying rate (100 Kbps,1 Mbps,10 Mbps, and 50 Mbps), transport protocol (TCP,UDP), and transmission pattern of a so-called On-Off application (with 25, 50, 75, 100 % duty cycle). Furthermore, we varied the following Wi-Fi parameters: the standard and frequency (802.11n on 2.4 GHz, 802.11n on 5 GHz, 802.11ac on 5 GHz), the channel width (20 GHz, 40 GHz), and guard interval (short, long), while assuming the presence of the dynamic Minstrel Rate control algorithm.

Based on the previous parameters, 384 unique experiments were constructed and augmented, leading to over 800 GB of IQ samples. As discussed in Section III, these samples are processed into two image datasets, time and time-frequency domain, each one composed of 49920 images and requiring around 6 GB of storage. This transformation reduces the storage requirement to less than 1% of that needed for the raw IQ dataset. Regarding the PCAP files used by the packet-based approach, 527389 packets were obtained. Finally, each domain dataset was randomly split to create the training (80%), validation (10%), and test (10%) datasets. In the remainder of this, we compare the performance of the three traffic recognition models using the image and packet-based datasets.

B. Recognition of different burst traffic patterns

To solve this task, we design and train models to recognize 4 types of burst traffic with 25 (low), 50 (medium), 75 (high), 100 (constant) % duty cycle. Table III shows that both models,



(a) Traffic Pattern (b) Transport Protocol (c) TX Rate

Fig. 3. Test dataset normalized confusion matrices using time (Top) and time-frequency domain images (Middle), and raw IP packet (Bottom)

TABLE III
MODEL ACCURACY USING TRAINING, VALIDATION AND TEST DATASETS

Model			Accuracy		
			Train	Validation	Test
CNN 1	Traffic Pattern	Time	0.999	0.962	0.962
CNN 2		Time-Frequency	0.999	0.991	0.990
CNN 7		Packet	0.443	0.449	0.438
CNN 3	Transport Protocol	Time	0.995	0.965	0.968
CNN 4		Time-Frequency	0.999	0.998	0.997
CNN 8		Packet	1.0	1.0	1.0
CNN 5	TX Rate	Time	0.987	0.979	0.978
CNN 6		Time-Frequency	0.999	0.999	0.998
CNN 9		Packet	0.836	0.821	0.811

time and time-frequency, achieved above 96 % accuracy in validation and test, but the higher accuracy is obtained using time-frequency images. Regarding the packet-based model, higher accuracy (82.3%) was only achieved when detecting low traffic patterns. Similar to other studies [10], the better performance in time-frequency images than only-time ones is expected since high noise together with fading channel effects have a bigger impact in images representing time domain. Additionally, the effects mentioned above cannot be seen in the packet domain, giving the worst performance of this model of the three tasks ($\leq 43\,\%$). Therefore, information about time dependence is not taken into account during classification.

According to the confusion matrix shown in Figure 3(a), the medium and high traffic patterns were the hardest to discriminate, and some of the examples were miss-classified as constant traffic. This behaviour can be explained due to the MCS algorithm. Given a fixed TX rate and a fixed time window w to create the images, If the transmission of a signal is using a low MCS, then more spectrum will be used in comparison to the same signal using a higher MCS.

C. Recognition of TCP and UDP traffic

In this task, the created models were trained to recognize two different transport protocols (UDP and TCP). According to the results shown in Table III, we can see how all models have accuracy above 96 % in validation and test, and even obtain an accuracy of above 99 % using time-frequency images and packets. The confusion matrix for this task, as shown in Figure 3(b), also indicates that the miss-classification distribution among the two classes is similar. This result, in both time and time-frequency domains, is due to the spectrum being highly occupied in high TX rates and duty cycles, causing the resulting images to look more similar for both protocols. It can also be observed how easy it is for the packet-based model to discriminate between UDP and TCP due to the evident differences in the transport header for these two protocols.

D. Recognition of different traffic rates

Finally, we also designed and trained models for recognizing the TX rate used to transmit the signals. The results in Table III and Figure 3(c) show that it is possible to discriminate between different TX rates using images accurately. Furthermore, most of the miss-classification of the data was in the dataset containing signals generated at 10 Mbps in both domains, which were miss-classified as signals generated at 50 Mbps. Similar to the results of the previous two classification tasks, a combination of a high TX rate and a high duty cycle can generate examples that may look similar in the spectrum, especially if they are augmented with fading effects and noise, and therefore confusing the classifier. However, these kinds of examples are also responsible for providing a better generalization capacity to the DL models to learn and generalize to unseen data better. This is also verified given the high accuracy in both validation and test datasets. Note that classifying traffic patterns and traffic rates can be replaced by regression models to predict the continues values of these datasets.

The packet-based model achieved a surprising high-accuracy on this classification task. However, this performance was not because the model found time relationships between packets, as there was no timestamp information in the packet headers, but due to an efficient function that the CNN learned based on the IP identification (IP-id) field of the packets generated by NS-3 on each experiment. In each experiment, NS-3 initializes the IP-id field with a zero value, and each new packet will increase the IP-id by one if there is no IP packet fragmentation (as in our experiments).

Roughly speaking, we generate 40 packets for the 5-seconds experiments with 100 Kbps TX rate, 400 for 1 Mbps, 4000 for 10 Mbps, and 20000 for 50 Mbps. The packet-based model creates boundaries between the different TX-rates by using only the range of values in the IP-id header. For example, the IP-id range between 0-40 is optimally mapped to the label 100 Kbps to maximize the accuracy of this class, we got 0.991%, while miss-classifying other traffic as this one has a limited negative impact ($\leq 15\%$ in 1 Mbps traffic and $\leq 1.5\%$ in 10 Mbps). Note that 10 Mbps traffic has the worst classification accuracy, which was $\leq 70\%$. This is a direct consequence of having 10% of their IP-ids in the 1 Mbps traffic and using 20% of the IP-ids of the 50 Mbps traffic.

VI. Conclusions

In this paper, we have shown that it is possible to perform traffic classification at the spectral level accurately. We have presented three classification tasks: the recognition of different 1) transport protocols, 2) traffic patterns, and 3) traffic rates, that were solved using a CNN based architecture. Based on a novel data randomization approach, we have created datasets for these tasks in different domains, namely the time, time-frequency and packet domain. Using the datasets from the time and time-frequency domain, we design and train six different models that each achieve a classification accuracy of above 96% using time-domain data, or above 99%, when using time-frequency domain data in all three tasks. Compared to a state-of-the-art packet-based approach, our models outperformed it on the recognition of traffic patterns and TX rate tasks, while achieving similar performance on the transport protocol task. These results show how our approach learns and uses time-dependent features to recognize data traffic patterns in a non-intrusive manner. In contrast, state-of-the-art packet-based methods failed to do it, and they are limited only to identify static features such as the differences between the TCP and UDP packet headers. Future work includes, among others, the validation of our models with real-life data and to investigate the use of regression models to predict continues values of TX rate and traffic patterns.

References

[1] Cisco, "Cisco Visual Networking Index: Forecast and Trends, 2017—2022," 2017. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.pdf

[2] M. S. Afaqui, E. Garcia-Villegas, and E. Lopez-Aguilera, "IEEE 802.11ax: Challenges and Requirements for Future High Efficiency WiFi," *IEEE Wireless Communications*, vol. 24, no. 3, 2017.

[3] J. G. Andrews, S. Buzzi, W. Choi *et al.*, "What will 5g be?" *IEEE Journal on selected areas in communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[4] F. M. Abinader, E. P. Almeida, F. S. Chaves *et al.*, "Enabling the coexistence of LTE and Wi-Fi in unlicensed bands," *IEEE Communications Magazine*, vol. 52, no. 11, pp. 54–61, 2014.

[5] S. K. Sharma, T. E. Bogale, S. Chatzinotas *et al.*, "Cognitive radio techniques under practical imperfections: A survey," *IEEE communications surveys and tutorials*, 2015.

[6] Z. A. Qazi, J. Lee, T. Jin *et al.*, "Application-awareness in sdn," in *ACM SIGCOMM computer communication review*, vol. 43, no. 4. ACM, 2013, pp. 487–488.

[7] L. Li, Y. Yu, S. Bai *et al.*, "An effective two-step intrusion detection approach based on binary classification and k -nn," *IEEE Access*, vol. 6, pp. 12 060–12 073, 2018.

[8] M. Lotfollahi, R. S. H. Zade, M. J. Siavoshani *et al.*, "Deep packet: A novel approach for encrypted traffic classification using deep learning," *arXiv preprint arXiv:1709.02656*, 2017.

[9] M. Schmidt, D. Block, and U. Meier, "Wireless interference identification with convolutional neural networks," in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*. IEEE, 2017, pp. 180–185.

[10] M. Kulin, T. Kazaz, I. Moerman *et al.*, "End-to-end learning from spectrum data: A deep learning approach for wireless signal identification in spectrum monitoring applications," *IEEE Access*, vol. 6, pp. 18 484–18 501, 2018.

[11] M. Camelo, A. Shahid, J. Fontaine *et al.*, "A semi-supervised learning approach towards automatic wireless technology recognition," in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, Nov 2019, pp. 1–10.

[12] K. Davaslioglu and Y. E. Sagduyu, "Generative adversarial learning for spectrum sensing," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.

[13] J. Tobin, R. Fong, A. Ray *et al.*, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 23–30.

[14] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Engineering Applications of Neural Networks*, C. Jayne and L. Iliadis, Eds. Springer International Publishing, 2016, pp. 213–226.

[15] S. Jeong, U. Lee, and S. C. Kim, "Spectrogram-based automatic modulation recognition using convolutional neural network," in *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2018, pp. 843–845.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.