Machine Learning For User Traffic Classification in Wireless Systems

Enrico Testi, Elia Favarelli, and Andrea Giorgetti

DEI, University of Bologna
Via Venezia 52, 47521 Cesena, Italy
e-mail: enrico.testi4@unibo.it, elia.favarelli2@unibo.it, andrea.giorgetti@unibo.it

***Abstract*—The ability to answer all important questions about the radio-frequency (RF) scene is essential for cognitive radios (CRs) to be effective. In this paper, we propose a RF-based automatic traffic recognizer that, observing the radio spectrum emitted by a communication link and exploiting machine learning (ML) techniques, is able to distinguish between two types of data streams. Numerical results based on real waveforms collected by a RF sensor, demonstrate that over-the-air user traffic classification is possible with an accuracy of 97% at high signal-to-noise ratios (SNRs). Moreover, we show that using a neural network (NN) very good classification performance can be achieved also at low SNRs (around $2\,$dB). Finally, the impact of the observed RF bandwidth and the acquisition time window on the classification accuracy are analyzed in detail.**

I. Introduction

With the advent of internet of things (IoT), there will be a rapidly growing demand for radio services by billions of devices, making the radio spectrum an increasingly valuable resource. Most of modern communication standards provide a static utilization of the radio spectrum resources, which results in its under-utilization [1].

From this perspective, cognitive radio (CR) devices will have to probe the RF scene in time, space and frequency domain to ensure that a well defined portion of the spectrum is free, making multidimensional spectrum analysis mandatory [2], [3]. On large scale network infrastructures indeed, the classification of transmissions, the spatial localization of sources, and the search for spectrum holes, may benefit by the extensive use of machine learning (ML) algorithms [4].

Traffic classification may allow to automatically recognize the user-level application that has generated a given stream of packets from direct observation of the packets or from the spectrum occupancy. An in-depth knowledge of the composition of traffic, as well as the identification of trends in application usage, may help CRs improving network design and provisioning. Moreover, traffic classification represents the first step in the direction of anomaly detection for the identification of malicious use of network resources, and for security operation such as firewalling and filtering of unwanted traffic. There are many approaches and methodologies for traffic classification proposed in literature [5]. Such methodologies can be grouped into three main categories [6]. *Port-based classification* is used when the protocols are assigned to well-known transport-layer port (i.e., TCP, HTTP). The main issue with this method is that many applications use dynamic port-negotiation mechanisms in order to guarantee the privacy of the user. *Payload-based classifiers* inspect the content of packets beyond the transport layer headers, looking for features in packet payloads that can distinguish an application protocol from the others. These classifiers are usually used when traffic is not encrypted or enclosed into other application-level protocols. *Statistical classification* analyses statistical attributes, also called *features*, of the received traffic in order to perform classification through *data mining* algorithms [5]. This methodology can be applied to encrypted traffic, because the content of packets is never exploited, it is lightweight in terms of sensing, but it can be less accurate than payload-based classifiers.

While traffic classification in wired networks have been extensively investigated, very few works address the problem in wireless systems, despite the emergence of CR technology makes this aspect rather important [6]. This work proposes a ML approach for traffic classification in wireless networks using low-cost radio-frequency (RF) sensors. In particular, the main contributions are the following:

- We propose the use of a blind packet detector which requires inexpensive RF sensors and preserves the user privacy.
- We compare the performance of three ML classifiers such as logistic regression (LR), support vector machines (SVMs) and neural networks (NNs).
- We provide an in-depth analysis of the classifiers performance as a function of the signal-to-noise ratio (SNR) at the RF sensor, the observed RF bandwidth, and the acquisition time window.

The numerical results, based on real waveforms captured by a low-cost software defined radio (SDR), reveal that over-the-air user traffic classification is possible also at relatively low SNRs.

The rest of the paper is organized as follows. In Section II the scenario and the problem setting are described. Section III provides an overview of the classification algorithms. Extensive numerical results are given in Section IV. Conclusion are drawn in section V.

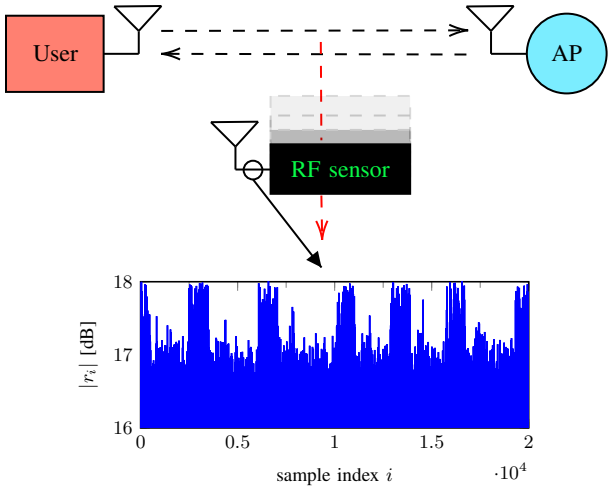

Fig. 1. Experimental setup for over-the-air user traffic inference and an example of the signal envelope received by the RF sensor.

II. System Overview and Problem Setup

Let us consider the scenario sketched in Fig. 1 where a RF sensor, tuned to a specific frequency, performs a downconversion followed by an analog to digital conversion to capture samples of a two-way communication between two wireless devices. For example, without loss of generality, let us consider the case where we are interested to infer the type of traffic generated by a user in a WiFi system where the two devices are a smartphone and an access point (AP) operating in the 2.4 GHz band.

With the purpose of observing the temporal flow of the packets exchanged by the two devices the RF sensor can be rather simple. The goal is in fact to keep the sensor as simple (and cheap) as possible and build an automatic traffic classifier exploiting only features observable by the temporal evolution of the packets flow. Therefore, all the subsequent tasks are performed without demodulating the received signals, so that a simple energy detector (ED) receiver suffices [7] [8]. Using low-cost devices the measurement quality is usually affected by front-end impairments, low sampling frequency, low resolution, and low sensitivity. However, as will be shown in Section IV the samples have been acquired with enough accuracy to allow the ML algorithms to guarantee good performance.

As a case study, this paper focuses on some traffic patterns generated by popular apps such as Youtube™ and Whatsapp™. In particular, after selecting the proper features and a training phase, we aim to recognize which type of application is running on the user's smartphone. We would like to remark that this is a toy example chosen with the purpose to work with readily available data sources.

A. Data pre-processing

The samples, $\{r_i\}_{i=1}^{N}$, of the complex envelope of the received signal at the RF sensor, are pre-processed to detect the packet and estimate their time-of-arrival (ToA). The sample rate, f_s, can be varied according to the needs. On one hand, a high sampling rate can guarantee an accurate estimation of the ToA at the cost of an increasing computational rate required to processing the samples. On the other hand, a low sampling rate may alleviate the computational burden but resulting in a coarser estimate. In Section IV we analyze this trade-off in detail.

Packet detection and ToA estimation can be impaired by the additive white Gaussian noise (AWGN) and channel propagation (fading and non-line-of-sight (NLOS)), making the classification quite challenging, especially in the low SNR regime.[1] For instance, in Fig. 1 we depict an example of signal samples collected at a SNR of 2 dB, where the noise level is remarkably high with respect to the signal one.

The need to extract statistical features of the traffic flow requires an estimate of the ToA of the received packets. To do so, we use the received samples to detect the presence of a packet by conventional binary hypothesis test. In particular, the signal received by the RF sensor can be represented by a vector of N samples, with elements

$$r_i = \begin{cases} x_i + \nu_i, & \mathcal{H}_1 \\ \nu_i, & \mathcal{H}_0 \end{cases} \tag{1}$$

where $\mathcal{H}_0$ and $\mathcal{H}_1$ indicate the null hypothesis and the presence of a packet, respectively, x_i is the i-th sample of the packet related signal and ν_i is the i-th noise sample, with $i = 1, \ldots, N$. When the samples x_i are unknown, the detection problem leads to an ED, represented by the test

$$|r_i|^2 \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \eta \tag{2}$$

where the threshold η can be chosen accordingly to the Neyman-Pearson criterion to guarantee a predefined false alarm probability. Then, the ToA of the k-th packet, t_k, is represented by the time instant corresponding to upward threshold crossing.

After obtaining the ToA of two consecutive packets, t_k and t_{k-1}, it is possible to calculate the k-th *inter-arrival time* $\tau_k = t_k - t_{k-1}$, which represents the data used to extract the features.

During the training phase N_T samples are collected and processed to obtain n_T inter-arrival times, while during the test phase N_O samples are collected and n_O inter-arrival times are obtained. To simplify the notation, in Section II-B the parameters N_T and N_O are indicated with N, while n_T and n_O are denoted by n.

B. Features selection

Starting from the basic idea that Youtube™ traffic can be considered as a dense stream of packets which contains a relatively large volume of data, and Whatsapp™ traffic can be seen as sparse groups of packets representing the messages sent and received by the user, there are four relevant features which characterizes the statistic of packets inter-arrival time:

[1] In the scenario considered in the numerical results only line-of-sight (LOS) propagation has been considered.

- *Sample mean*

$$\mathsf{M}_\tau = \frac{1}{n}\sum_{k=1}^{n}\tau_k. \tag{3}$$

- *Sample variance*

$$\mathsf{V}_\tau = \frac{1}{n-1}\sum_{k=1}^{n}(\tau_k - \mathsf{M}_\tau)^2. \tag{4}$$

- *Kurtosis*, defined as

$$\mathsf{K}_\tau = \frac{\mathsf{m}_4}{\mathsf{m}_2^2} \tag{5}$$

where m_4 and m_2 are respectively the 4th and the 2nd order moments, estimated from samples as

$$\mathsf{m}_q = \frac{1}{n}\sum_{k=1}^{n}(\tau_k - \mathsf{M}_\tau)^q. \tag{6}$$

- *Rate of packets*, R_p, i.e., number of packet arrivals per second.

III. Survey of the ML Algorithms

In this section, we briefly review the algorithms adopted for over-the-air traffic classification: LR, SVM, and single-hidden-layer neural network (SHLNN).

Let us define the *feature matrix* $\mathbf{\Phi} \in \mathbb{R}^{F\times D}$ where D is the number of points, i.e., the number of snapshots considered either in training phase or in the test phase (each snapshot consists in the capture of samples within an observation window), while F is the number of features extrapolated for each point, i.e., $F = 4$ according to Section II.

The matrix $\mathbf{\Phi}$ is related to the *association matrix* $\mathbf{t} \in \mathbb{R}^{D\times C}$, where C is the number of classes (or categories); $C = 2$ in the current setting. The element t_{dc} of $\mathbf{t}$ is 1 when the d-th observation belongs to the c-th class, otherwise its value is set to -1. In case there are two classes, as in our case study, we can equivalently define an association vector, $\mathbf{t}$, with elements t_d, that contains 1's or -1's with the same criterion.

From now on it is convenient to extend the number of features from F to $F+1$ appending a row of ones to the bottom of the feature matrix $\mathbf{\Phi}$.[2] This allows the algorithm to estimate the bias term.

A. Logistic regression

LR is a nonlinear machine learning algorithm often used for classification problems. For LR the error function, known as *cross-entropy error function*, can be written as [9]

$$E(\mathbf{w}) = -\sum_{d=1}^{D}(t_d \ln(y_d) + (1-t_d)\ln(1-y_d)) \tag{7}$$

where y_d are the elements of the vector $\mathbf{y}(\mathbf{\Phi}) \in \mathbb{R}^D$ obtained through a logistic sigmoid acting on a linear function of the feature matrix, i.e.,

$$\mathbf{y}\left(\mathbf{\Phi}\right) = \sigma\left(\mathbf{w}^T\mathbf{\Phi}\right) \tag{8}$$

[2]To ease the notation the augmented matrix is still denoted by $\mathbf{\Phi}$.

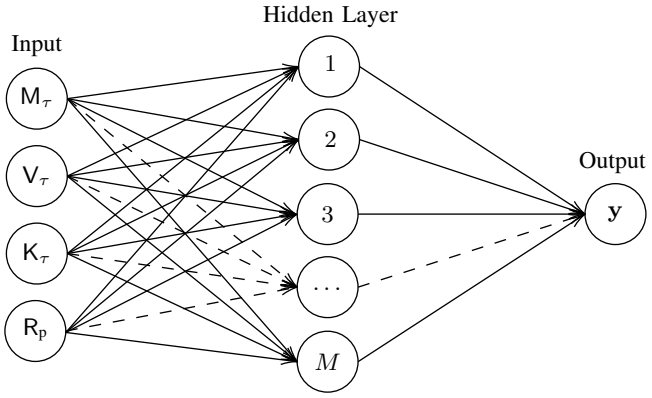

Fig. 2. The single-hidden-layer neural network considered for over-the-air traffic classification.

where $\mathbf{w} \in \mathbb{R}^{F+1}$ is the *weight vector* containing the LR model coefficients. Conventional iterative algorithms like *gradient descent* or *Newton's method*, can be used to minimize the error function and find the optimal $\mathbf{w}$ that ensures the best separation between the two classes.

B. Support vector machine

SVM is an evolution of *perceptron* to overcome convergence problems and search for an optimal solution [9].

To find the best solution in the case of linearly separable data, the SVM define the error function by introducing a regularization term as follows

$$g(\mathbf{w}) = \sum_{d=1}^{D}\ln\left(1 + e^{-y_d(\mathbf{x}_d^T\mathbf{w})}\right) + \lambda||\mathbf{w}||_2^2 \tag{9}$$

where $||\cdot||_2$ is the ℓ_2-norm, and λ is a parameter that controls the trade-off between how well we satisfy the original constraints and the pursue of a large margin classifier.

C. Neural network

As a third classification approach, we propose a SHLNN. The basic idea behind a SHLNN is the following. Any scalar function $f(\mathbf{x})$ with $\mathbf{x} \in \mathbb{R}^S$ can be sampled in D points $\{\mathbf{x}_d\}_{d=1}^D$ with $\mathbf{x}_d \in \mathbb{R}^S$, and represented with a D-dimensional vector $\mathbf{z} = (z_1, z_2, \ldots, z_D)$. Hence, since any vector can be always represented by a linear combination of the so called basis vectors, the same reasoning can be applied to approximate a continuous function. In fact, considering a basis of continuous functions of cardinality M, $\{f_m(\mathbf{x})\}_{m=1}^M$, the sampled function $\mathbf{z}(\mathbf{x}_d)$ can be approximated with a finite summation

$$\sum_{m=0}^{M} f_m(\mathbf{x}_d)w_m \approx \mathbf{z}(\mathbf{x}_d). \tag{10}$$

It is also possible to use adjustable basis functions. In this case, the basis are parametrized functions (*activation functions*) as [10]

$$f_m(\mathbf{x}) = \tanh(c_m + \mathbf{x}^T\mathbf{v}_m) \tag{11}$$

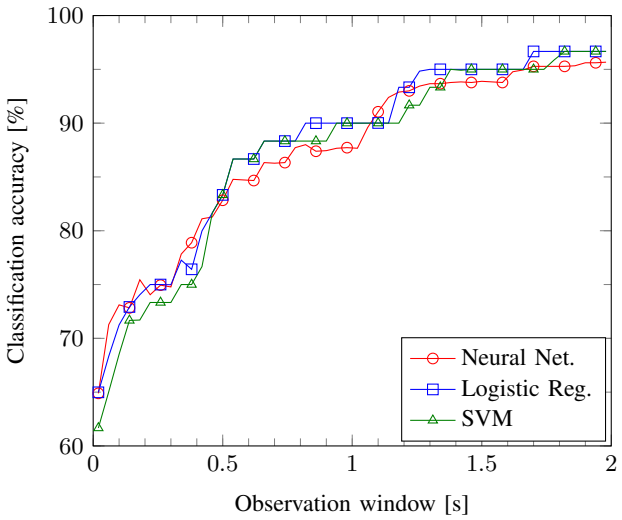

Fig. 3. Classification accuracy as a function of the capture window duration for the three algorithms proposed.

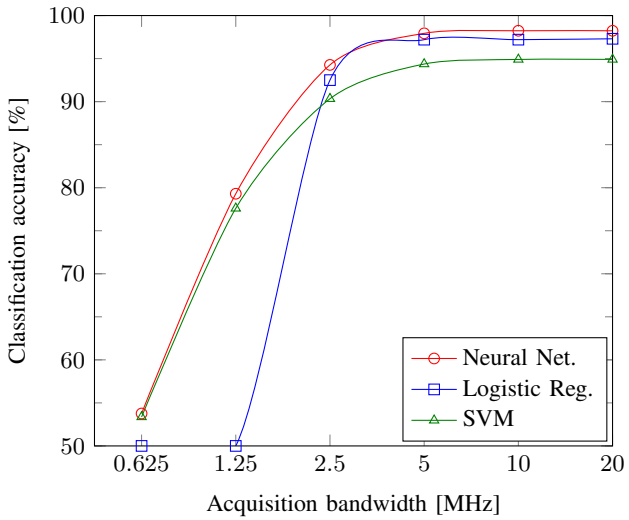

Fig. 4. Classifiers accuracy as a function of the acquisition bandwidth for the three algorithms proposed.

where c_m is the activation function bias, and $\mathbf{v}_m$ are the activation function parameters. In this case, the error function can be defined as [10]

$$g(\mathbf{w}) = \sum_{d=1}^{D} (\mathbf{f}_d^T \mathbf{w} - z_d)^2 \tag{12}$$

where $\mathbf{f}_d = (1, f_1(\mathbf{x}_d), f_2(\mathbf{x}_d), \ldots, f_M(\mathbf{x}_d))^T$. This approach is what in literature is called *single-hidden-layer feed forward NN*. Such NN can be easily used for the classification problem: in that case the samples $\mathbf{x}_d$ are the columns of the feature matrix, i.e., $\mathbf{\Phi} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_D]$ and $S = F$. During the training phase the network tracks the function described by the matrix of the features and finds the classification regions boundary. Once the boundary has been found it is possible to classify new points according to their position on the hyperplane.

IV. Numerical Results

In this section, we present several tests performed to compare the classification algorithms and to reveal when a RF-based traffic classification is possible with satisfactory accuracy. The RF sensor is represented by the SDR platform *HackRF One* operating in receiving mode with bandwidth 20 MHz. The maximum bandwidth allowed to capture the entire IEEE $802.11n$ signal in the 2.4 GHz ISM band. The sensor output is composed by the in-phase and the in-quadrature baseband signals, each one represented with 8 bit/sample. In the pre-processing phase, the threshold η of the ED has been set to guarantee $P_{\text{FA}} = 10^{-3}$. According to (1) the SNR is defined as $\mathsf{SNR} = (\sum_{i=1}^{N} |x_i|^2)/(\sum_{i=1}^{N} |\nu_i|^2)$. Unless otherwise specified, the RF sensor was positioned at a distance of one meter from both user (a smartphone) and AP, on the red line showed in Fig. 1; the acquisition window during the training phase and the test phase was $N \cdot f_{\text{s}} = 5$ s for both classes, i.e., Whatsapp™ and Youtube™, respectively. Then, the $F = 4$ features described in Section II were obtained from the captured data. The performance metric for classification is the accuracy, defined as percentage of correct classifications. The SVM parameter λ was set to 0.1.

A. Size of the acquisition window

This test aims to find a proper acquisition window duration to guarantee that the algorithms reach the maximum achievable accuracy. With this aim, Fig. 3 shows how the accuracy of the classification algorithms depend on the window width. On one hand, if the capture window is too short the accuracy degrades significantly. On the other hand, a capture window larger than 2 s seems to provide only minor performance improvement. This behavior is related to the time scale for which the features selected are effective. Moreover, the figure shows that the three algorithms behave in the same way with short observations, but increasing the window length the accuracy of LR and SVM is slightly better than that of the NN. This is probably due to the different training procedures.

To guarantee the maximum accuracy achievable by the algorithms, a capture window of 5 s has been chosen for the following tests.

B. Acquisition bandwidth

This test aims to investigate the trade-off between the acquisition bandwidth, given by f_{s}, and the performance of the algorithms. In particular, the sampling rate has been changed from 625 kHz (i.e., $1/32$ of the WiFi signal bandwidth) to 20 MHz (i.e., the entire WiFi signal bandwidth), by decimation. As we can see in Fig. 4, the minimum bandwidth that allows a classification accuracy of 90% is 2 MHz for SHLNN and SVM, and 2.5 MHz for LR. Note that a reduction of the bandwidth affects the correct detection of the packets, reducing the classification capability of the algorithms.

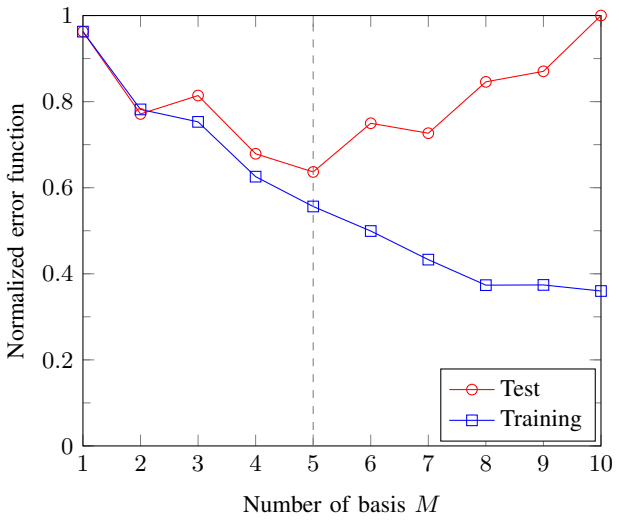

Fig. 5. k-Fold Cross-Validation of the SHLNN.

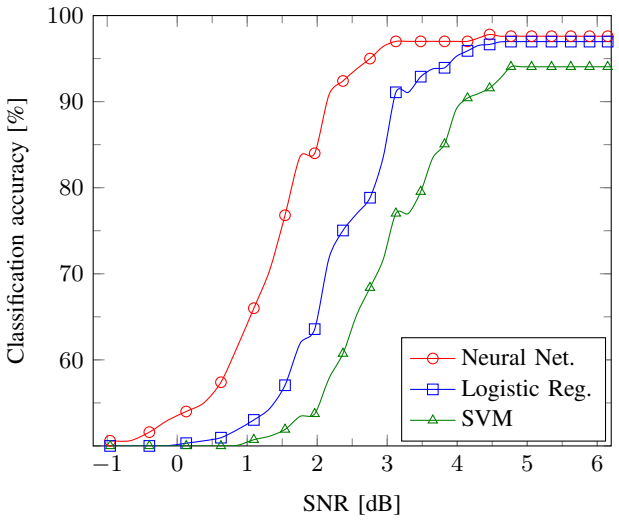

Fig. 6. Classification accuracy as a function of the SNR for the three algorithms proposed.

C. Neural network cross-validation

As explained in Section III-C, a crucial aspect in NNs is the choice of the cardinality of the basis, M. Indeed, a high M might lead to *overfitting*. The cross-validation process is necessary to determine how many basis elements, or neurons, have to be used to correctly train the NN. The well-known *k-fold cross-validation* method has been chosen for this purpose, and $n_{\mathrm{T}} = n_{\mathrm{O}} = 50$ points per class have been used for both training and testing. In Fig. 5 the results of the k-fold cross-validation are shown, where the error function (12) is plotted as a function of the number of basis elements M. It is possible to notice that if $M > 5$ the error function of the test subset increases substantially, so $M = 5$ appears to be the best choice for this setting.

D. Impact of SNR

To evaluate the robustness of the classification algorithm, the RF sensor has been moved along the red line shown in Fig. 1. As expected, an increase of the distance with respect to the AP-User link causes a decrease of the received power and of the SNR. A reduction of the SNR degrades the ED performance, which cause both missed packet detection and inaccurate ToA estimation. These aspects then reflects on the quality of the feature extracted and finally on the performance of the classifier. On this point, it is interesting to understand the accuracy of the classification algorithms at different SNR regimes.

For this test, 60 points of training per-class, corresponding to 60 acquisition windows of length 5 s, have been used. The classification performance varying the SNR is reported in Figure 6. Note that the SHLNN provides superior performance with respect to SVM and LR. Furthermore, LR outperforms SVM and reaches almost the same performance of the NN for SNR above 4 dB.

V. Conclusion

In this work we proposed and studied a RF-based automatic traffic recognition exploiting machine learning techniques. The numerical results based on real waveforms, collected by a RF sensor, demonstrated that user traffic classification is possible, that it does not require expensive devices, and that its accuracy can be larger than 90% even at relatively low SNRs. Using a NN such very good classification performance can be achieved also at SNR around 2 dB. Future research directions include the use of multiple sensors to provide superior performance.

Acknowledgement

This work was supported in part by the European Commission under the H2020 EuroCPS project (grant no. 644090).

References

[1] K. Sithamparanathan and A. Giorgetti, *Cognitive Radio Techniques: Spectrum Sensing, Interference Mitigation and Localization*. Boston, USA: Artech House Publishers, Nov. 2012.

[2] P. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Pearson, May 2005.

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

[4] P. de Kerret, D. Gesbert, and M. Filippone, "Decentralized deep scheduling for interference channels," *CoRR*, vol. abs/1711.00625, 2017. [Online]. Available: http://arxiv.org/abs/1711.00625

[5] S. Valenti, D. Rossi, A. Dainotti, A. Pescape, A. Finamore, and M. Mellia, "Reviewing traffic classification," *Lect. Notes in Comp. Science*, vol. 7754, 2013.

[6] J. Kornycky, O. Abdul-Hameed, A. Kondoz, and B. Barber, "Radio frequency traffic classification over wlan," *IEEE/ACM Trans. on Netw.*, vol. 25, no. 1, Feb. 2017.

[7] A. Mariani, A. Giorgetti, and M. Chiani, "Effects of noise power estimation on energy detection for cognitive radio applications," *IEEE Trans. Commun.*, vol. 59, no. 12, pp. 3410–3420, Dec. 2011.

[8] ——, "Wideband spectrum sensing by model order selection," *IEEE Trans. Wireless Commun.*, vol. 14, no. 12, pp. 6710–6721, Dec. 2015.

[9] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer Verlag, Aug. 2006.

[10] J. Watt, R. Borhani, and A. K. Katsaggelos, *Machine Learning Refined*. Cambridge University Press, 2016.