Blind Traffic Classification in Wireless Networks

Enrico Testi, Lorenzo Pucci, Elia Favarelli, and Andrea Giorgetti
DEI, University of Bologna
Via dell'Università 50, 47522 Cesena, Italy
e-mail: {enrico.testi4, lorenzo.pucci3, elia.favarelli2, andrea.giorgetti}@unibo.it

***Abstract*—In this paper, we propose a non-collaborative radio-frequency (RF) sensor network that, observing the radio spectrum generated by the users of a wireless network, can separate and classify their activities. Numerical results demonstrate that using blind source separation (BSS) and some well-known classifiers, over-the-air user traffic identification is possible. Moreover, we demonstrate that the performance of the proposed methodology is remarkably good in the presence of multiple classes and rather robust when channel impairments (i.e., presence of shadowing) degrade BSS. For example, we show that using a neural network (NN) outstanding classification performance can be achieved even using a relatively low number of RF sensors with very short observation windows (i.e., 30 ms).**

I. Introduction

With the advent of the internet of things (IoT), there is a rapidly growing demand for radio services by billions of devices, making the radio spectrum an increasingly valuable resource [1]–[4]. In such scenarios, in-depth knowledge of the composition of traffic, as well as the identification of trends in application usage, may help cognitive radios (CRs) improving network design and provisioning. Moreover, it seems very important, if not mandatory, that user traffic classification is carried out without the need to be a part of the network or without increasing its overhead sharing additional information. There are many approaches and methodologies for traffic classification proposed in the literature [5], [6]. Such methodologies can be grouped into three main categories [7]. *Port-based classification* is used when the protocols are assigned to a well-known transport-layer port (i.e., TCP, HTTP). The main issue with this method is that many applications use dynamic port-negotiation mechanisms to guarantee user privacy. *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 to perform classification through *learning* algorithms [5]. This methodology can be applied to encrypted traffic because the content of packets is never exploited, and it is lightweight in terms of sensing, but it can be less accurate than payload-based classifiers.

This work was supported by MIUR under the program "Dipartimenti di Eccellenza (2018-2022) - Precise-CPS," and the CoACh project funded by the POR FESR 2014-2020 program.

While traffic classification in wired networks has been extensively investigated, very few works address the problem in wireless systems, although the emergence of CR technology makes this aspect rather important [7]. In fact, more in-depth knowledge of how a network uses the wireless medium and, thus, the classification of its users' activities may contribute to the development of much effective spectrum sharing strategies. In this context, it is desirable to automatically recognize the user-level application that has generated a given stream of packets from direct observation of the radio-frequency (RF) scene [2]–[4]. This work proposes a machine learning (ML)-based approach for traffic classification in wireless networks using low-cost RF sensors, where such sensors do not need to be part of the network to perform classification. In particular, the main contributions are the following:

- We propose the idea of using a sensor network that exploiting only the received signal strength (RSS) traces collected by the sensors is capable of classifying traffic patterns of the users of a wireless network.
- We design a specific blind source separation (BSS) method to reconstruct the transmitted power profiles of each node of the network.
- We compare the performance of classifiers, such as support vector machines (SVMs) and neural networks (NNs), and statistical tools, such as principal component analysis (PCA) and kernel principal component analysis (KPCA), to assess their ability to classify traffic patterns.
- We provide an in-depth analysis of the classifiers' performance as a function of the number of RF sensors, the acquisition time window, and the shadowing intensity.

The numerical results, based on simulated waveforms, reveal that over-the-air user traffic classification after BSS is possible, using a relatively low number of RF sensors and short acquisition time window.

Throughout the paper, capital boldface letters denote matrices, lowercase bold letters denote vectors, $(\cdot)^{\mathsf{T}}$ stands for transposition, $\odot$ stands for the element-wise product, $\mathbf{I}_N$ indicates the $N \times N$ identity matrix, $\mathbb{E}\{\cdot\}$ is the expectation operator, and $||\cdot||_2$ is the ℓ_2-norm operator. The remainder of this paper is organized as follows. In Section II, we introduce the scenario and system model. In Section III, the BSS method is proposed, while in Section IV, the classification algorithms are presented. Numerical results are given in Section V. Conclusions are drawn in Section VI.

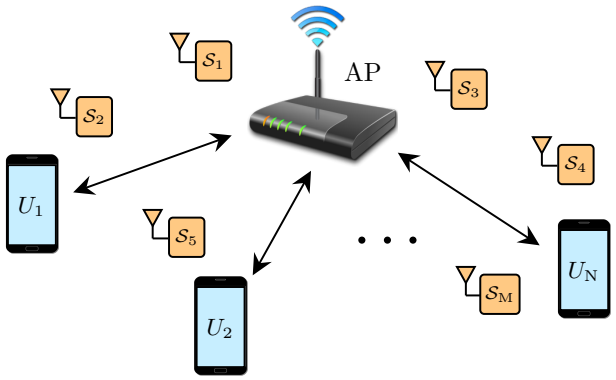

Fig. 1. System overview: the network of sensors monitoring the traffic patterns among the nodes of the wireless network.

II. System Overview and Problem Setup

Let us consider a scenario with a wireless network composed of N users monitored by M RF sensors (non collaborating with the network) all randomly placed on a two-dimensional landscape (see Fig. 1). We assume that the technical specifications of the network (i.e., wireless routing protocol, physical layer structure) are unknown, and each sensor in the landscape can measure only the instantaneous received power over short time intervals. The goal is to design an automatic traffic classification tool that exploits only features observable by the packet flows' temporal evolution. Therefore, all the subsequent tasks are performed without demodulating the received signals, so that a simple energy detector (ED) sensor suffices [8], [9].[1]

However, the extraction of the desired packet flows is quite challenging. Such patterns are masked by multiple access interference, packet collisions, as well as by physical layer impairments like propagation and noise. The most common propagation issues (e.g., shadowing) and the effects of thermal noise at the sensors have to be accounted in this scenario, and their influence on the accuracy of the algorithms needs to be assessed. Without loss of generality, the effect of shadowing between the nodes and the sensors is modeled by a log-normal distributed random matrix $\mathbf{S}$ whose elements are

$$s_{m,n} = \exp(\sigma_{\mathrm{S}} G_{m,n}) \quad m = 1, \ldots, M, \;\; n = 1, \ldots, N$$

where $G_{m,n}$ are independent, identically distributed (i.i.d.) zero mean Gaussian random variables (r.v.s) with unit variance and shadowing parameter σ_{S}.[2] Furthermore, assuming conventional energy detection, the thermal noise at the output of the integrator can be modeled as a constant ν that can be added to each sample received.[3] The profiles of the packets transmitted by each user are time series arranged as rows of the matrix $\mathbf{P} \in \mathbb{R}^{N \times K}$ where K is the number of power samples collected; each one calculated over contiguous short intervals of duration T_{i}. The channel is represented by the mixing matrix $\mathbf{H} \in \mathbb{R}^{M \times N}$, whose elements are the $h_{m,n}$ channel gains between sensor m and node n. Without loss of generality, the channel model adopted for $h_{m,n}$ is free-space path-loss. Therefore, the matrix of the received power profile $\mathbf{X} \in \mathbb{R}^{M \times K}$ can be written as

$$\mathbf{X} = (\mathbf{S} \odot \mathbf{H})\mathbf{P} + \nu \mathbf{1}_{M,K} \tag{1}$$

where $\mathbf{1}_{M,K}$ is an $M \times K$ matrix of all ones.

III. Blind Source Separation

For the traffic classification purpose, accurate knowledge of the transmitted power profiles of the users is mandatory. Assuming that the packet streams generated by each user are unknown, as well as the parameters of the propagation channel, BSS is a way to unmix the signals from the mixture (1) observed by the RF sensors. BSS consists of separating the source signals taking advantage of their statistical properties (i.e., statistical independence) when the mixing matrix is unknown [10]. For this purpose, we need to estimate the number of sources N. The computation of independent components can be made simpler and better conditioned if the mixture is pre-processed before the separation. In this section, we present a method for BSS based on whitening, dimensionality reduction, and independent component analysis (ICA). The proposed method works only in the case of $M \geq N$.

A. Whitening and estimation of the number of nodes

Whitening consists of applying a linear transformation to the observation matrix so that its components are uncorrelated and have unit variance. This process is accomplished through the eigenvalue decomposition of $\mathbf{X}$, and selecting only the most relevant components of the observation matrix, discarding the others. This is achieved by setting a threshold on the eigenvalues. The number of components selected coincides with the number of sources that we want to separate [10]. As a result, we estimate the number of nodes N of the network and obtain a whitened observation matrix $\tilde{\mathbf{X}} \in \mathbb{R}^{N \times K}$. Therefore, we have $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^{\mathrm{T}} = \mathbf{I}_N$, where $\tilde{\mathbf{X}}$ is the whitened version of $\mathbf{X}$ with lower dimension. The whitened vectors (i.e., the rows of $\tilde{\mathbf{X}}$) are orthogonal to each other but not necessarily independent. For this reason, ICA is now used to unmix the source signals.

B. Non-negative ICA

Given the white signal $\tilde{\mathbf{X}}$, we aim to reconstruct the original positive semi-definite matrix $\mathbf{P}$ with good accuracy. ICA is

[1] Such limitation in the sensing capability has a bright side: the privacy of the users is preserved, data can be encrypted, and the sensors can be cheap.

[2] The shadowing parameter is usually expressed as the standard deviation of the received power in deciBel, i.e., $\sigma_{\mathrm{S}}(\mathrm{dB}) = \frac{10}{\ln 10}\sigma_{\mathrm{S}}$.

[3] The noise term in the ED is a central chi-squared r.v. with a number of degrees of freedom, N_{DOF}, proportional to the time-bandwidth product. When N_{DOF} is large the noise term can be considered constant. As detailed in Section V we consider a system with bandwidth $W = 20\,\mathrm{MHz}$ (i.e., WiFi channel) and an integration time $T_{\mathrm{i}} = 10\,\mu\mathrm{s}$, hence $N_{\mathrm{DOF}} = 2WT_{\mathrm{i}} = 400$.

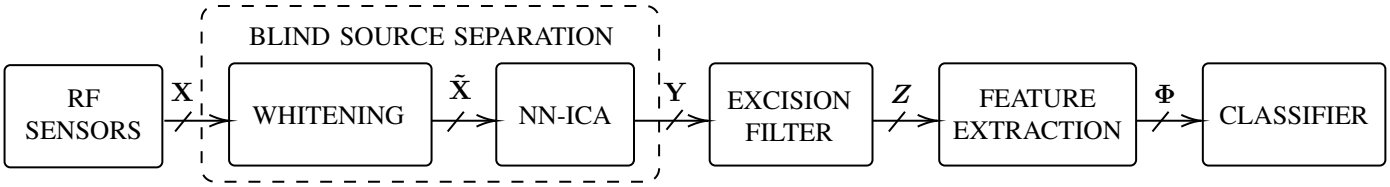

Fig. 2. Block diagram of the data processing chain for blind traffic classification of wireless users.

a processing method that finds statistically independent and non-Gaussian components from data. ICA variants differ from each other for the objective function and the optimization algorithm used. In general, the chosen function determines the properties of ICA, such as consistency and robustness, while the optimization algorithm impacts the convergence speed, memory occupation, and stability. The output of ICA is an unmixing matrix $\mathbf{W}$ such that

$$\mathbf{Y} = \mathbf{W}^{\mathrm{T}}\tilde{\mathbf{X}} \tag{2}$$

where $\mathbf{Y}$ contains the reconstructed signals. We adopted the iterative method for ICA under the hypothesis of non-negative signals as sources, with kurtosis as a measure of non-Gaussianity, and decorrelation based on the Gram-Schmidt orthonormalization, proposed in [10].

C. Excision filter

Signal unmixing is not perfect, because of the presence of noise and shadowing, so the output $\mathbf{Y}$ has residual crosstalk that has to be removed, e.g., by an excision filter.

To isolate the profiles of packets transmitted by each node, the signals in $\mathbf{Y}$ have been processed to obtain sequences of 0s and 1s arranged in a matrix $\mathbf{Z}$. In particular, the element $z_{n,k}$ contains 1 if node n is sending a packet at the time sample k and 0 otherwise. To do so, we use the received samples to detect the event "packet sent" by the conventional binary hypothesis test. The threshold η is set as a fraction $\alpha \in [0, 1]$ of the maximum of $\mathbf{Y}$, i.e.,

$$\eta = \alpha \max_{n,k}\{y_{n,k}\}. \tag{3}$$

The complete block scheme of the data processing chain is depicted in Fig. 2.

IV. User Traffic Classification

A. Features extraction

For the traffic classification problem, three different traffic profiles have been simulated, in particular, *video streaming*, *chat*, and *web navigation*. Video streaming traffic can be considered a dense stream of packets containing a relatively large volume of data, while chat traffic can be seen as sparse groups of packets representing the messages sent and received by the user. Web navigation, instead, produces a more variable traffic profile with respect to the other activities. An example of packet flows for these traffic types is reported in Fig. 3 There are four relevant features which characterize the statistic of packets' inter-arrival time:

- *Sample mean*
$$\mathsf{M}_\tau = \frac{1}{n}\sum_{k=1}^{n}\tau_k. \tag{4}$$
- *Sample variance*
$$\mathsf{V}_\tau = \frac{1}{n-1}\sum_{k=1}^{n}(\tau_k - \mathsf{M}_\tau)^2. \tag{5}$$
- *Kurtosis*
$$\mathsf{K}_\tau = \frac{\mathsf{m}_4}{\mathsf{m}_2^2} \tag{6}$$
where m_4 and m_2 are respectively the 4th and the 2nd order moments, estimated from samples as
$$\mathsf{m}_q = \frac{1}{n}\sum_{k=1}^{n}(\tau_k - \mathsf{M}_\tau)^q. \tag{7}$$
- *Rate of packets*, R_p, i.e., number of packet arrivals per second.

B. Survey of the classifiers

Let us define the *feature matrix* $\mathbf{\Phi} \in \mathbb{R}^{F\times D}$ where D is the number of points while F is the number of features extrapolated for each point, i.e., $F = 4$ according to Section IV-A.

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); $C = 3$ in the current setting. The element t_{dc} of $\mathbf{t}$ is 1 when the dth observation belongs to the cth class, otherwise its value is set to -1.

We now briefly review the algorithms adopted for over-the-air traffic classification: PCA, KPCA, SVM, and NN.

1) PCA: PCA is a widely known algorithm in exploratory data analysis. Considering the cth class, given the centered training set $\mathbf{\Phi}_c$, the algorithm remaps the training data from the feature space $\mathbb{R}^F$ in a subspace $\mathbb{R}^P$ (where $P < F$ is the number of principal components selected) that minimizes the information loss between the projected data and the original ones. The best subspace over which to project the data depends on the training set distribution and the number of components selected P. Iterating this process for all the C classes, we obtain a set of subspaces, one for each class, where to project the data. For the multi-class classification purpose, we seek to find which of the subspaces gives a better representation of the test data. For this reason, firstly, we project the test data set over all the subspaces already found. Then, data are remapped to their original space, and the Euclidean distance between the original data and the remapped ones is calculated

for each subspace of the set. The class corresponding to the subspace that gives the minimum Euclidean distance is the output of the classifier.

2) KPCA: This approach takes inspiration by the standard PCA and overcomes the limitation of the linear mapping that corresponds to finding linear boundaries in the original feature space. In many applications, this constraint represents a severe limitation and can sharply decrease the classification accuracy. KPCA firstly maps the data with a non-linear function, after which applies the standard PCA to find a linear boundary in the new feature space. Such boundary becomes non-linear, going back to the original feature space. A crucial point in KPCA is the selection of a non-linear function that leads to linearly separable data in the new feature space. In the literature, when the data distribution is unknown, the Radial basis function (RBF) kernel is often proposed as the right candidate to accomplish this task [11]. Suppose we have a generic point $\boldsymbol{\phi}_d$ that corresponds to a vector of length F, we can apply the RBF as follows

$$K_{\phi_{f,d}} = e^{-\gamma||\boldsymbol{\phi}_d - \phi_{f,d}||_2^2} \quad \text{with} \quad f = 1, 2, \ldots, F \tag{8}$$

where γ is a kernel parameter (inversely proportional to the width of the Gaussian function) that must be appropriately set, and $K_{\phi_{f,d}}$ is the fth component of the point $\boldsymbol{\phi}_d$ in the kernel space. Overall the starting vector $\boldsymbol{\phi}_d$ is mapped in a vector $\mathbf{K}_{\boldsymbol{\phi}_d}$ of length F. Applying now the PCA to the new data set obtained remapping all the training points, it is possible to find non-linear boundaries in the starting feature space for a better classification. It is good practice to center the points mapped with the RBF because the mapping in the new feature space could be non-zero mean.

3) Support vector machine: The SVM constructs a set of hyperplanes in high-dimensional space that can be used for tasks like classification or regression [12], [13]. Hence, it is a parametric learning algorithm whose error function includes a regularization term as follows:

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

where $\mathbf{w}$ is the vector of the weights of the SVM model, and λ is the regularization parameter.

4) NN: Considering the case study of this work, the groups of points of the three classes are not linearly separable; therefore, a shallow NN has been chosen as a fourth classification algorithm [14], [15]. In particular, we adopt a 2-hidden-layer feed-forward NN. The well-known *k-fold cross-validation* method has been chosen to avoid overfitting. During the training phase, the network tracks the function described by the matrix of the features and finds the classification region's boundary. Once the boundaries have been found it is possible to classify new points according to their position on the hyperplane.

V. 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 performance. As a figure of merit, we define the *accuracy* as the number of traffic streams correctly classified over all the tested ones. Since the number of test points for each class is the same, this figure of merit perfectly suits this case study. The RF sensors are uniformly distributed in the landscape of the network, represented by a squared area of side 10 m. The wireless network under test is based on the IEEE802.11n standard and is composed of an access point (AP) and $N = 3$ devices, all randomly placed in the area. The number of classes considered is $C = 3$, corresponding to three different pattern of activities generated by the users: *web navigation*, *video stream* and *chat*. The results presented in this section are obtained from NS-3 simulations of 100 different scenarios. More precisely, in each scenario, the position of the nodes, the position of the sensors, and the shadowing are chosen randomly, while each user generates its traffic profile according to one of the three patterns (an example is reported in Fig. 3). The sampling frequency at the RF sensors is set to $f_s = 10$ kS/s leading to a time resolution of 100 μs. The excision filter threshold η is set with $\alpha = 0.9$. For the PCA algorithm the number of components is set to $P = 1$, while for KPCA is $P = 3$ and $\gamma = 30$. The SVM parameter λ is set to 0.1. The NN has 40 nodes in the first hidden layer and 20 in the second one. All the layers are fully connected, and the activation functions are ReLU for the hidden layers and softmax for the output layer. The network is trained with 1500 points for 2500 epochs (iterations of the stochastic gradient descent algorithm) with an initial learning rate of 10^{-4}. The learning rate decreases by a factor 10 after 2000 epochs.

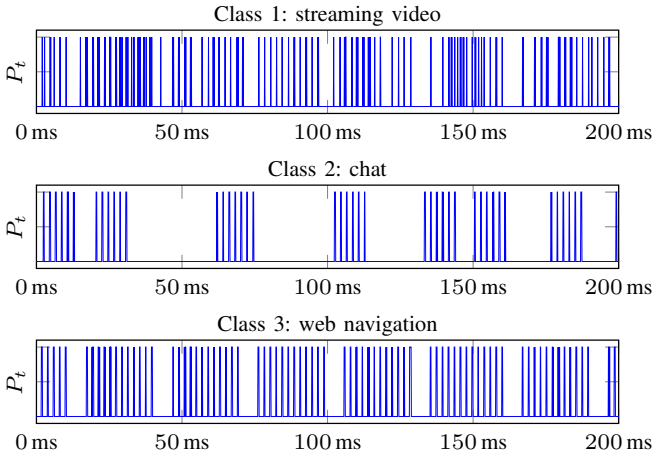

Fig. 3. An example of packet profiles transmitted by the users for the three activities considered.

A. Accuracy vs. number of RF sensors

In this test, we studied the performance of the classifiers as a function of the number of RF sensors distributed in the landscape. The shadowing parameter is set to $\sigma_S = 2$ dB and $\sigma_S = 5$ dB, respectively. As expected, the reconstruction error

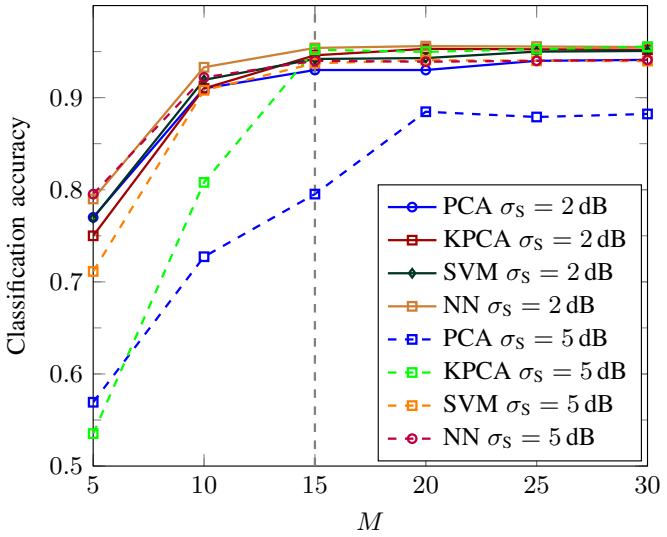

Fig. 4. Accuracy varying the number of sensors M with an observation window of 1 s.

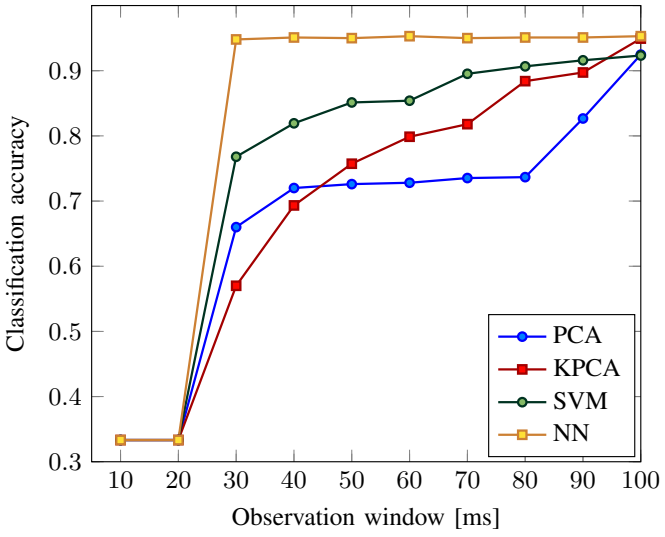

Fig. 5. Accuracy as a function of the observation window duration for $M = 15$ sensors.

of the BSS increases when a low number of sensors is used. In Fig. 4, it is shown how the quality of the classification of the algorithms falls as M drops below 15 sensors. Note that, the performance of the PCA results less effective than the other algorithms. In particular, Fig. 4 shows that the NN is the most suitable classifier in this scenario. For such tests, the observation window is set to 1 s, according to the next section.

B. Accuracy vs. observation window

This test aims to find a proper acquisition window duration to guarantee that the algorithm reaches the maximum achievable accuracy. Now, the number of sensors is $M = 15$ and the shadowing parameter is $\sigma_S = 2$ dB. With this aim, Fig. 5 shows how the accuracy of the classification algorithms depends on the window width. As expected, if the capture window is too short (i.e., 20 ms), the accuracy degrades significantly. This behavior is related to the time scale for which the features selected are meaningful. Moreover, the figure shows that the NN outperforms the other algorithms even with a short observation window (i.e., 30 ms). This is probably due to the different training procedures.

VI. Conclusion

In this work, we proposed a user traffic classification framework for wireless networks, based on RF measurements. We showed that, after the BSS, the NN outperforms the other classifiers achieving remarkable performance also in case of propagation impairments (e.g., shadowing), and with a short observation window (30 ms). The analysis of the proposed solution revealed that the number of RF sensors strongly impacts the performance of the algorithms. This is because traffic classification is affected by imperfect power profile reconstruction of the transmitted signals at the nodes.

References

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

[2] E. Testi, E. Favarelli, and A. Giorgetti, "Machine learning for user traffic classification in wireless systems," in *Europ. Sig. Proc. Conf. (EUSIPCO)*, Rome, Italy, Sep. 2018, pp. 2040–2044.

[3] E. Favarelli, E. Testi, L. Pucci, and A. Giorgetti, "Anomaly detection using WiFi signal of opportunity," in *IEEE International Conference on Signal Processing and Communication Systems (ICSPCS)*, Surfers Paradise, Gold Coast, Australia, Dec. 2019.

[4] E. Testi, E. Favarelli, L. Pucci, and A. Giorgetti, "Machine learning for wireless network topology inference," in *IEEE Int. Conf. on Signal Proc. and Comm. Systems (ICSPCS)*, Gold Coast, Australia, Dec. 2019.

[5] S. Valenti, D. Rossi, A. Dainotti, A. Pescape, A. Finamore, and M. Mellia, "Reviewing traffic classification," *Lect. Notes in Comp. Science*, vol. 7754, pp. 123–147, Feb. 2013.

[6] T. De Schepper, M. Camelo, J. Famaey, and S. Latré, "Traffic classification at the radio spectrum level using deep learning models trained with synthetic data," *International Journal of Network Management*, 2020.

[7] 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, pp. 56–68, Feb. 2017.

[8] 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.

[9] A. Mariani, A. Giorgetti, and M. Chiani, "Wideband spectrum sensing by model order selection," *IEEE Trans. Wireless Commun.*, vol. 14, no. 12, pp. 6710–6721, Dec. 2015.

[10] X. Yu, D. Hu, and J. Xu, *Blind Source Separation: Theory and Applications*, 1st ed. Wiley Publishing, 2014.

[11] B. Schölkopf, A. Smola, E. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

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

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

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

[15] E. Favarelli, E. Testi, and A. Giorgetti, "One class classifier neural network for anomaly detection in low dimensional feature spaces," in *IEEE Int. Conf. on Signal Proc. and Comm. Systems (ICSPCS)*, Gold Coast, Australia, Dec. 2019.