Towards Low-Complexity Wireless Technology Classification Across Multiple Environments

Jaron Fontaine[1], Erika Fonseca[2], Adnan Shahid[1], Maicon Kist[2], Luiz A. DaSilva[2], Ingrid Moerman[1], Eli De Poorter[1]

Abstract

To cope with the increasing number of co-existing wireless standards, complex machine learning techniques have been proposed for wireless technology classification. However, machine learning techniques in the scientific literature suffer from some shortcomings, namely: (*i*) they are often trained using data from only a single measurement location, and as such the results do not necessarily generalise and (*ii*) they typically do not evaluate complexity/accuracy trade-offs of the proposed solutions.

To remedy these shortcomings, this paper investigates which resource-friendly approaches are suitable across multiple heterogeneous environments. To this end, the paper designs and evaluates classifiers for LTE, Wi-Fi and DVB-T technologies using multiple datasets to investigate the complexity/accuracy trade-offs between manual feature extraction and automatic feature learning techniques.

Our wireless technology classification reaches an accuracy up to 99%.

[*]The research leading to these results has received funding from the Fund for Scientific Research Flanders (Belgium, FWO-Vlaanderen, FWO-SB grant no. 1SB7619N and FWO EOS grant no. 30452698 (MUSE_WINET)), from the European Horizon 2020 Programme (under grant agreement no. 688116 eWINE Project [www.ewine-project.eu] and 732174 ORCA Project [https://www.orca-project.eu/]), and from the Science Foundation Ireland (under grant no. 13/RC/2077 (CONNECT)).

[1]J. Fontaine (corresponding author, SB PhD fellow at FWO), A. Shahid, E. De Poorter and I. Moerman with the IDLab, Department of Information Technology, Ghent University - imec, iGent Tower, Technologiepark-Zwijnaarde 15, B-9052 Ghent, Belgium. (email: Jaron.Fontaine@Ugent.be)

[2]E. Fonseca, M. Kist and L. A. DaSilva are with the CONNECT Research Centre for Future Networks and Communications, Trinity College Dublin, 2 Dublin, Ireland.

[3]Manuscript received December 20, 2018; revised —.

Moreover, we propose the use of data augmentation techniques to extend these results to unseen environments at the cost of only 2% reduction in accuracy. When concerning generalisation capabilities, complex automatic learning techniques surpass simple manual feature extraction approaches. Finally, the complexity of these automatic learning techniques can be significantly reduced by using computationally less intensive received signal strength indicator data while reaching acceptable accuracies in unseen environments (92% vs 97%).

Keywords: manual feature extraction, automatic feature learning, wireless technology classification, machine learning, CNN

---

1. Introduction

With the advent of multimedia-enriched mobile phone applications, traffic demand from wireless users is increasing substantially. Furthermore, the number of wireless Internet of Things (IoT) devices is growing at an unprecedented rate: it is predicted that by 2020 there will be around 20 billion wireless devices around the globe [1].

In this context, machine learning, which offers the ability to learn without being explicitly programmed, shows enormous potential to better manage the limited resources of a wireless network and enable the delivery of a new generation of services.

Due to limited licensed bands and the growing traffic demands, the mobile communication industry is striving for offloading traffic from licensed to unlicensed bands. In Releases 13 and 14 of Long Term Evolution (LTE), the 3rd Generation Partnership Project (3GPP) has proposed Licensed-Assisted Access (LAA), in which LTE can operate on both licensed and unlicensed bands via carrier aggregation [2]. This approach, however, raises questions on its effect on the performance of legacy IEEE 802.11 (Wi-Fi) [3]. In such a co-existence environment, it is necessary to make intelligent decisions for maintaining the Quality of Service (QoS) requirements of both technologies. On the other hand, it is predicted that the 5th generation (5G) network will provide 1000 times the capacity as compared to the current system [4]. Offloading licensed traffic to unlicensed bands is beneficial, nevertheless it cannot solely fulfill the extensive capacity requirement. In this regard, an efficient sharing of licensed bands is a promising solution [5]. Various standardization bodies, European Telecommunications Standards Institute (ETSI) and the 3GPP are currently focusing on various licensed spectrum sharing models such as to apply cognitive radio techniques by radio environment maps (REM)s [6] and radio access network (RAN) sharing [7], respectively.

A first step towards achieving this objective is for wireless systems to be able to identify what other wireless technologies are present in the same band and what their characteristics of operation are. In this paper, we design and analyse machine learning techniques for technology classification in shared spectrum. In our evaluation of those techniques, we consider three technologies: Wi-Fi, LTE and Digital Video Broadcasting Terrestrial (DVB-T). These technologies are likely to operate in shared spectrum in the near future. Due to the 3GPP LAA proposals, LTE and Wi-Fi will operate and compete with each other in unlicensed bands [2]. Moreover, the reuse factor used in licensed DVB-T systems leads to significant amounts of unused spectrum at a given location [8, 9, 10]. In order to efficiently utilise the licensed spectrum, secondary users can use it without creating any harmful impact on the primary network. This spectrum sharing model was used by the Federal Communications Commission (FCC) for television bands and is termed as white space reuse [10].

To operate in shared spectrum, it is crucial that a wireless system is able to identify other technologies present in its vicinity, for interference avoidance and management, as well as for the detection of systems that may be operating in violation of the spectrum regime agreed upon for the band. The use of machine learning for wireless technology classification allows unprecedented technology classification accuracy using a wide range of signal features. However, a number of research issues still remain open:

- **Extensibility of results to different environments.** In theory, machine learning allows scalability by building a generalised model using a broad set of signals, collected in multiple environments. However, when using small datasets, as if often the case in scientific research, this generalisation remains a challenging problem [11].
- **Selection of the input features.** It is currently still an open research question on how to best engineer input features to enable efficient machine learning [12]. Manual feature selection limits the number of required input features to only the ones deemed most effective, but it requires extensive domain expert knowledge and can limit the performance due to the inability to extract hidden or underlying features. On the other hand, automatic feature learning enables faster development of models and applications while also trying to improve the representa-

tion of data by discovering previously unknown features, at the risk of making the models more complex. To the best of our knowledge, the efficiency gains of both approaches for wireless technology classification have not yet been quantified and compared.

The main contributions of our work are the following:

- **Quantitative comparison of the efficiency of machine learning techniques using manual feature extraction versus automatic feature learning for wireless technology classification.** Specifically, we compare these two approaches by using multiple machine learning techniques, including decision trees, neural networks, convolutional neural networks (CNN) and image classification techniques. In addition, we evaluate the impact of different input features, including Received Signal Strength Indicator (RSSI) data (suitable, for example, to embedded devices) as well as more complex input features such as In-phase and quadrature (IQ) samples and Fast Fourier Transform (FFT) of the IQ samples that generates spectrogram images, to explore how well automatic deep learning can exploit features in more complex data.
- **Analysis of the generalisability and robustness to noise of wireless technology classification using machine learning.** More specifically, we test generalisability using data collected in different unseen environments, to exploit the model's flexibility. Furthermore, the robustness of the models is explored by inducing noise into the datasets. This allows the assessment of the classification accuracy for multiple Signal to Noise (SNR) levels.
- **Trade-off and complexity analysis of machine learning techniques.** We compare the previously mentioned techniques by analysing their complexity in terms of trainable parameters, memory footprint and training time. We also discuss the trade-offs concerning the complexity of the proposed techniques.

The remainder of the paper is organised as follows. Section 2 discusses related work. Next, various feature learning techniques are presented, together with a dataset description, in section 3. In section 4, manual feature extraction techniques based on RSSI distributions are introduced, together with a detailed description of the decision trees and a fully connected neural network (FNN) that we used. Next, automatic feature learning techniques based on IQ samples and RSSI values, along with the CNN designs adopted, are introduced in section 5. In section 6, results of the aforementioned approaches are presented and compared in terms of accuracy, generalisation, robustness and complexity. The paper ends with conclusions in section 7.

2. Related work

Machine learning techniques are increasingly popular and widely adopted at different layers of the network protocol stack. Table 1 lists recent papers in the domain of wireless technology classification with their classification goals, input data, machine learning approaches and compares their contributions in terms of generalisation to multiple (unseen) locations, robustness to SNR and complexity trade-offs.

- The authors in [13] used CNNs for classifying 802.11 b/g, 802.15.4 and 802.15.1, all of which operate in unlicensed bands. Their accuracy exceeds 95% with a signal-to-noise ratio greater than -5dB.
- The authors of [14] classify the presence of radar signals, even with simultaneous transmissions of LTE and Wi-Fi systems.
- The authors of [15] target the same technologies as our paper. However, instead of machine learning, [15] uses fixed algorithms (heuristics) in an attempt to classify Wi-Fi, LTE and DVB-T, and the paper does not validate the results using different datasets.
- Besides technology classification, it is also possible to classify modulation techniques, for example using k-nearest neighbors (k-NN), Support Vector Machines (SVM) and Naive Bayes algorithms [16] or CNN based machine learning [17].
- Paper [18] identified eight kinds of signals: binary phase shift keying (Barker codes modulation), linear frequency modulation, Costas codes, Frank code and polytime codes (T1, T2, T3 and T4). This paper used image-based CNNs, which train on spectrogram images instead of RSSI or IQ data.
- In [19], the authors propose an end-to-end learning technique using spectrum data. Their goal is to identify modulation techniques and detect wireless interference with automatic feature learning. Three CNNs are trained with different kinds of data: IQ samples, amplitude/phase data and frequency domain data. Their experiments show that amplitude/phase data can outperform IQ and frequency domain data in modulation classification, while the frequency domain achieves the highest accuracy for interference detection.

Table 1: Overview of related work in the field of wireless technology classification

Paper	[13]	[14]	[15]	[16]	[17]	[18]	[19]	This
Classification goal	802.11, 802.15.4, 802.15.1	Radar	802.11, DVB-T, LTE	Modulation	Modulation	Modulation	Modulation, interference	802.11, DVB-T, LTE
Input Data	IQ	Spectrogram	RSSI	IQ	FFT	Spectrogram	IQ	RSSI, IQ, FFT, spectrogram
Approach	CNN	CNN	Fixed algortihm	k-NN, SVM, Naive Bayes	LSTM, DNN	CNN	CNN	Rforest decision trees, FNN, CNN
Generalisation locations			+-					✓
Robustness to SNR	✓	✓		✓	✓	✓	✓	✓
Complexity trade-offs			+-	✓	✓			✓

Most of the above mentioned papers used IQ samples in the frequency-domain as training input, with some using additional data such as phase, amplitude and average magnitude FFT. These samples are used as an input for the machine learning techniques. However, IQ samples require complex sensing methods and such capability is not available on most resource-constrained wireless devices. Only [17] and [15] (although [15] does not discuss complexity trade-offs) adopt a more resource-friendly solution using, respectively, average magnitude FFT data or RSSI data that contains less information compared to IQ samples but is easier to collect, while [16] discusses complexity trade-offs off multiple classifiers, with complex IQ data. Most papers do validate robustness to noise with multiple SNR levels, an important metric to validate classification performance. Unfortunately, only [15] uses training data from multiple locations, but none of the above papers evaluate the performance of its proposed machine learning techniques using multiple independent and unseen datasets from different locations. Thus, in this paper we propose and discuss which models are best suited to increase accuracy, robustness and generalisability while trying to minimise complexity. To this end, we (*i*) evaluate more types of input data than prior work (manual features from RSSI, RSSI, raw IQ, FFT IQ image-based), (*ii*) evaluate more machine learning techniques than prior work (Decision Tree, FNN and CNN) and (*iii*) analyse the impact of using two separate datasets from different locations.

On another layer of the network stack, above signal and technology recognition, sits traffic recognition. Likewise, traffic recognition is an active research topic in many performance optimisation and monitoring areas. These include mobile, anonymity and encrypted traffic classification that enable profiling and allow management tools to enhance network performance [20, 21, 22]. However, the main difference is that our work focuses on robustness and generalisation towards multiple environments that can have various channel conditions. Moreover, these works targeting traffic recognition target manual and statistical feature extraction, while the models presented in this paper favor raw signals to automatically extract features using CNNs. However, when considering manual feature extraction, C4.5 decision trees and random forests, the proposed models achieved good results comparable to the traffic recognition papers.

3. System description

In this section, we propose a spectrum manager framework which makes use of the models in this paper and assists operators for fine tuning their spectrum decisions. As mentioned above, one of our goals is to assess the generalisability of the proposed machine learning techniques for technology classification to systems deployed in different locations and under different conditions. Hence, we describe the datasets we collected and used in our study. These datasets are restricted to perform single-label classification. Hence, no overlapping signals were allowed. Finally, an overview of the evaluated technology classification approaches gives an overview of the models.

3.1. Spectrum manager framework

The proposed spectrum manager is shown in Figure 1 and performs the following three tasks (*i*) fetch IQ samples, (*ii*) results from the trained models, and (*iii*) spectrum decisions. The heart of the spectrum manager is a classification module, which we design by using machine learning approaches that do not require domain expertise. In the first task, IQ samples/RSSI values are fetched from Universal Software Defined Radio (USRP) which is part of the spectrum manager and is in a close proximity of the operators. In the second task, the trained models are used for getting identification of the technologies from the IQ samples fetched in the first task. Finally, the

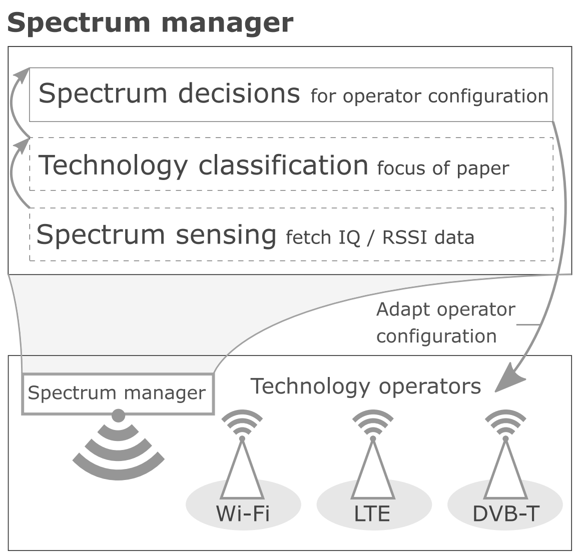

Figure 1: A spectrum manager can make decision based on technology classification models proposed in this paper to optimise usage of the wireless spectrum by different operators.

third task, makes spectrum policies and builds interference maps. This information can be conveyed by the spectrum manager to the operators for fine tuning their spectrum decisions so that they can fairly coexist with each other.

3.2. Data acquisition

To train technology classification models, we have utilised seven datasets: the 6 datasets were captured at multiple locations in Ghent, Belgium and the second one in Dublin, Ireland. We have made all datasets publicly available for future research comparisons. [4] [5] The objective of utilising datasets captured at multiple and different locations is to investigate how well the model can generalise for unseen environments. More precisely, the results in this paper evaluate the performance of our models, trained on Ghent's dataset and validated on Dublin's dataset. For the remainder of the paper, we refer to training dataset as a *seen* dataset and the validation dataset as

[4] The dataset captured in Ghent is available at https://github.com/ewine-project/Technology-classification-dataset

[5] The dataset captured in Dublin is available at https://github.com/ewine-project/lte-wifi-iq-samples

an *unseen* one.

- The seen dataset consists of IQ samples of LTE, Wi-Fi and DVB-T captured in 6 various locations in Ghent [6].
- The unseen dataset consists of IQ samples for LTE and Wi-Fi. These samples were collected[7] in the CONNECT building in Dublin city centre [23].

In both locations, IQ samples where captured, from which the RSSI was calculated[8] using (1) for $N = 16$.

$$RSSI = 10 * log_{10}(\frac{1}{N}\sum_{k=1}^{N}(I_k^2 + Q_k^2)) \tag{1}$$

where N and k correspond to the number of IQ samples per RSSI and the index of IQ samples, respectively.

Figures 2 and 3 show the time domain and spectrogram representation of the IQ samples of the seen (two locations in Ghent are shown) and unseen dataset, respectively. The figures show clear similarities but also have differences in terms of background noise, sending intervals and signal strength. These environmental and antenna-related differences are needed to enable and verify generalisation capabilities of the trained models in section 6.

3.3. Evaluated technology classification approaches

Table 2 provides an overview of the proposed approaches for wireless technology classification and the machine learning techniques adopted, together

[6]An Anritsu MS 2690A spectrum analyser was used to capture samples of each of the aforementioned signal types [15]. The Wi-Fi signal, captured in various office locations in Ghent, and contains traces at 5540 MHz and at 2412 MHz. The LTE signal was obtained from a base station nearby, operating at 806 MHz. Lastly, DVB-T signals were captured from a local TV broadcasting station that operates at 482 MHz. The IQ samples were collected at the rate of 10 MHz for a duration of 1.1 seconds.

[7]As a capture device, we used a B210 USRP software defined radio. From the dataset, we used 14 measurements, each of 2 sec, which consist of 125,000 RSSI or 2 million IQ samples, which translate to a total number of 1.75 million RSSI values or 28 million IQ samples.

[8]In total 68,750 RSSI values or 1.1 million I/Q samples were computed for each measurement of 1.1 seconds. We down sampled the measurements to a rate of 1 MHz to reduce the dataset footprint. 163 measurements were performed, which translate to 11,206,250 RSSI values or 179,300,000 IQ samples in total.

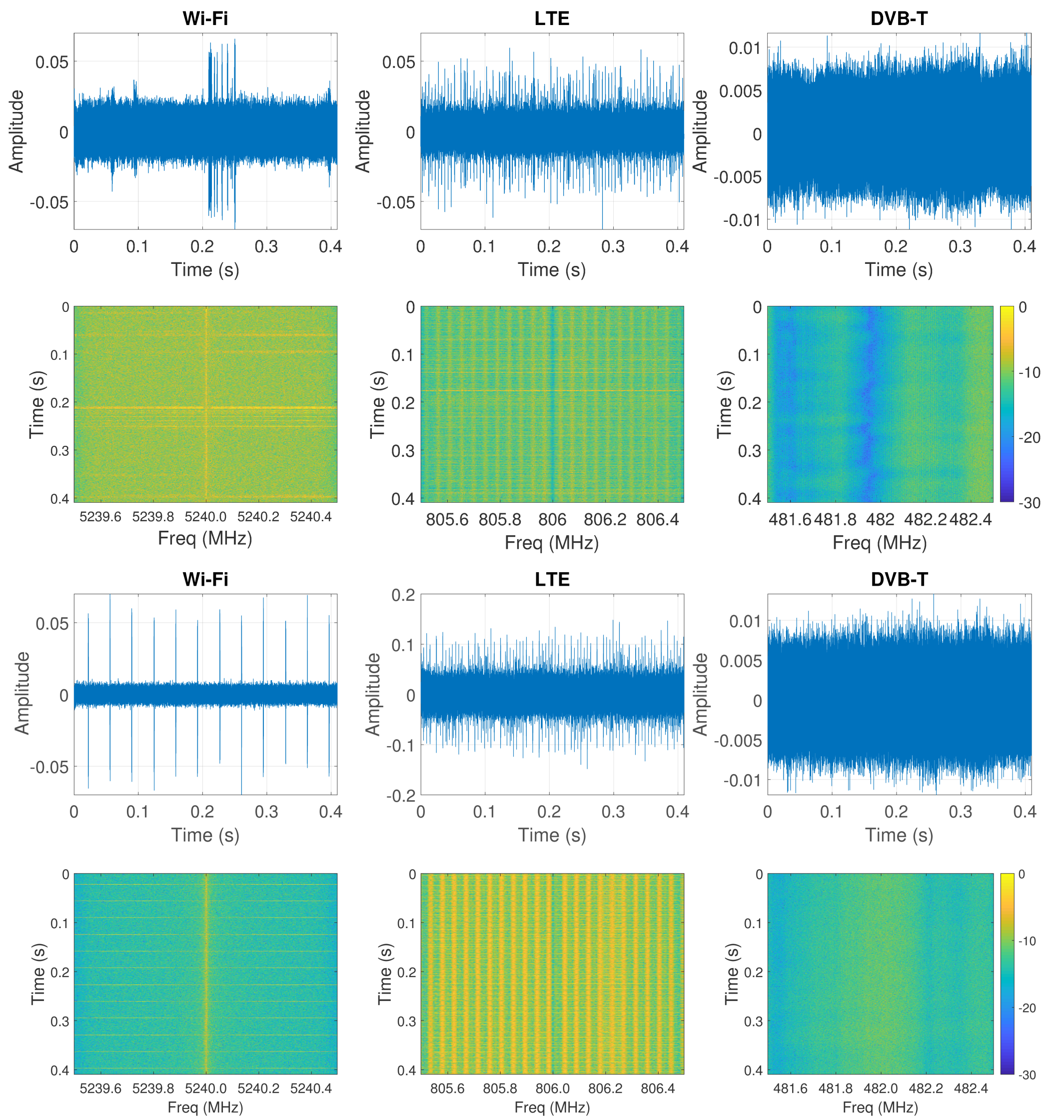

Figure 2: Time domain and and spectrogram representation of the seen dataset showing different characteristics for each technology. Two locations in Ghent with different environmental characteristics are shown, which can boost generalisation to multiple locations.

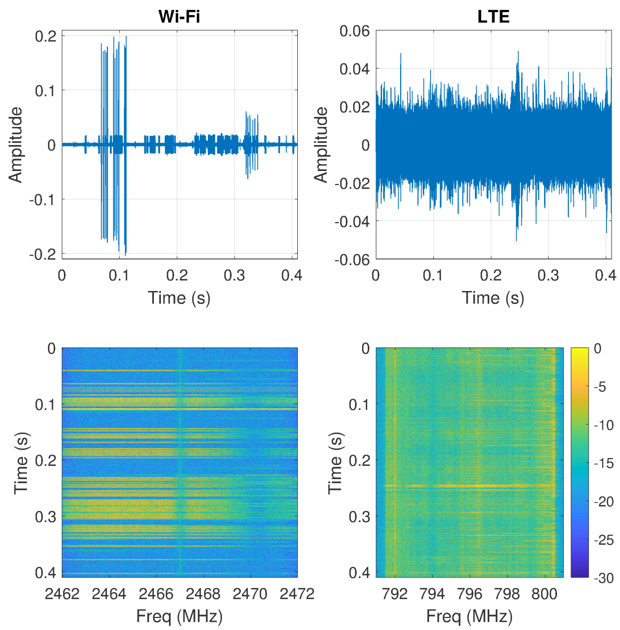

Figure 3: Time domain and and spectrogram representation of the unseen dataset captured at Dublin showing different characteristics compared to the seen dataset.

with their training data format. In the final column, we refer to the section where we discuss each approach in detail.

Table 2: Machine learning techniques and feature extraction approaches for technology classification proposed in this paper

Approach	**ML technique**	**Data**	**Section**
Man. feat.	Fully connected neural networks	RSSI	4.3
Man. feat.	Decision trees and random forests	RSSI	4.4
Auto. feat.	Conv. neural networks	RSSI	5.2
Auto. feat.	Conv. neural networks	IQ	5.2
Auto. feat.	Conv. neural networks	Spectrogram	5.2

Figure 4 draws an overview of the steps taken to achieve manual and autonomous feature extraction. Every training process starts with RAW IQ sample datasets collected at our various locations. Depending on the approach, samples will need to be recomputed to other formats. RSSI values, computed as discussed in 3.2, are used to manually extract features. In this scenario, discussed in 4.2, the model receives only the optimal selected subset of features. IQ samples can be processed with FFT and visualised with a spectrogram or directly used as raw input to the model. Once the data is processed, the corresponding model is trained.

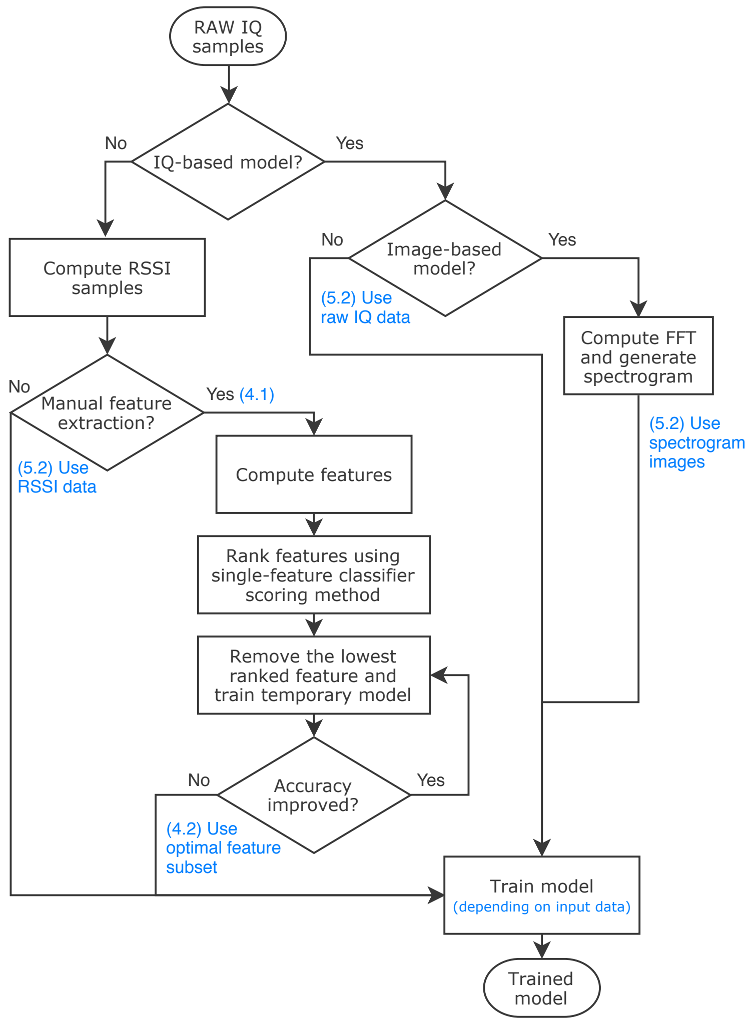

Figure 4: Overview of the steps taken to achieve manual and autonomous feature extraction, including their sections.

4. Manual feature extraction based on RSSI distributions

This section discusses the manual feature extraction and selection processes and describes briefly the machine learning techniques which were used to train classifiers and generate results.

4.1. Manual feature extraction

Before extracting features, we preprocessed and converted RSSI data into histograms that estimate the probability distributions of RSSI values. As a use-case, these histograms are calculated using 256 RSSI samples which corresponds to a sample duration of 4.096 ms. This method is based on [15], which shows that these distributions offer valuable features. However, here we extract and evaluate more and different features that are simple to calculate. The histograms are used as input for the feature extraction module. The output of this module is a feature vector X_i such that,

$$X_i = [\, r_0, r_1, ..., r_{19}, R_{min}, R_{max}, P_n, P_w, H_{std}, D_{mean}, D_{median}] \quad (2)$$

where:

- $\{r_0, r_1, ..., r_{19}\}$ is a set of 20 intervals selected from the input histogram. r_0 corresponds to the leftmost part of the histogram, while r_{19} represents the rightmost part. Each interval thus contains 5% of the histogram and its value indicates the frequency of RSSI values within the corresponding interval.
- R_{min} is the minimum RSSI value and thus the left boundary of the histogram.
- R_{max} is the maximum RSSI value and thus the right boundary of the histogram.
- P_n is the measured number of peaks in the histogram.
- P_w is the width of the highest peak.
- H_{std} is the standard deviation of the histogram values.
- D_{std} is the standard deviation of the RSSI values upon which the histogram is calculated.
- D_{mean} is the mean of the RSSI values upon which the histogram is calculated.
- D_{median} is the median of the RSSI values upon which the histogram is calculated.

Table 3: Number of selected features with their accuracy on a (un)seen dataset

# features	Training accuracy	Validation accuracy
28 (all)	88.4%	74.0%
15	88.9%	77.0%
11	88.8%	76.1%

4.2. Feature selection

One advantage of using manual feature extraction methods is the control over which features are used to train the model. [24] discusses feature selection as a method to improve the accuracy of the model. To allow optimal selection, each feature is ranked according to the score calculated by a ranking method. Several such methods have been proposed by [24]. In this work, we used the single-feature classifier method which gives the highest prediction accuracy compared to other methods such as entropy-based, correlation-based, etc. The single-feature classifier method takes each of the features, one-by-one, and calculates the resulting accuracy as a ranking metric for the corresponding feature.

In order to determine how many features we select from the ranked list, we start removing the lowest ranked feature and proceed up the list. Each time, the classifier uses the remaining features to train. Finally, we know which and how many are the most optimal features to select. The following fifteen features were selected: $r_1, r_2, r_3, r_4, r_8, r_9, r_{10}, r_{11}, r_{12}, r_{19}, R_{min}, r_{max}, P_w, D_{std}$, D_{mean}. Table 3 illustrates a higher accuracy when selecting a subset of fifteen features compared to all 28 features, which confirms the findings of [24] are also valid for wireless technologies, where using too many features can complicate the model. In addition, the results show that removing too many features results in a lower accuracy score. The model losing valuable information to learn classifying wireless technologies explains this behaviour. For further results, the fifteen highest scoring features, according to the single-feature classifier ranking method, were used to compare the performance of the classifier against competing approaches.

4.3. Fully connected neural network

Table 4 provides an overview of the employed artificial neural network (ANN) architecture. This ANN is also known as a FNN because of its multiple (two) hidden layers having connections to all nodes of the previous and following layers. The input layer with a size of 29 neurons, or 15 after feature selection, receives the manually extracted feature vectors, $\boldsymbol{v}_n \in \mathbb{R}^{29}$, containing values as described in subsection 4.1. This layer is followed by two fully connected layers with 25 and 10 neurons respectively. Finally, an output layer classifies the wireless signal through three neurons for DVB-T, Wi-Fi and LTE. The first two layers use a radial basis activation function (3):

$$output = radbas(\,||\,\boldsymbol{w} \cdot \boldsymbol{p}\,||\,b), \tag{3}$$

where $\boldsymbol{w}$ and $\boldsymbol{p}$ are weight and input vectors respectively, b is the bias and $radbas(n)$ is

$$radbas(n) = e^{-n^2}. \tag{4}$$

The output of this activation function will be 1 when the difference between $\boldsymbol{w}$ and $\boldsymbol{p}$ is 0.

The last layer of the neural network uses a softmax activation function (5):

$$softmax(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{5}$$

where $j = 1, ..., \#classes$ and z_i is

$$z_i = \sum_k p_k W_{ki} \tag{6}$$

where i is the considered output neuron, $k = 1, ..., \#neuronsPreviousLayer$, p_k is the output of the previous layer's neuron and W_{ki} is the weight applied to p_k. In contrast to the models proposed in section 5, this neural network is much smaller. There is no need for feature learning in raw data using many deep and convolutional layers. Rather we designed a less complex FNN that can perform better given already extracted features [25], hence this design choice. The model learns by applying scaled conjugate gradient back-propagation each time it is given training data. This gradient is used to update the weights and bias values of the neural network. The training of such a network requires the inputs, weights and activation functions all to have derivative functions.

4.4. Decision tree and random forest

Compared to neural networks, decision trees offer insight into how classification is performed. Unlike neural networks, they are not considered black-boxes. Decision trees compare one of the features at each of their nodes. If

Table 4: FNN structure

Layer type	**Layer size**	**Activation function**
Input	15 neurons	radbas
Fully connected	25 neurons	radbas
Fully connected	10 neurons	radbas
Output	3 neurons	softmax

the value of the feature is smaller than the trained value, then the algorithm follows the left branch; if it is larger, then it follows the other direction. During the training phase of a decision tree, decisions are made upon which feature should be selected and what the value should be. This decision depends on the implementation, e.g., the C4.5 algorithm, which we used, splits the tree using normalised information gain, also called gain ratio (7) [26]:

$$Gainratio(Y, X) = \frac{H(Y) - H(Y|X)}{H(X)} \tag{7}$$

with

$$H(X) = -\sum_{i=1}^{n} P(x_i) \ln_2 P(x_i), \tag{8}$$

and

$$H(Y|X) = H(Y, X) - H(X), \tag{9}$$

where $P(x_i)$ is the probability of feature X having a value x_i out of all possible values. $H(X)$ thus represents uncertainty in X or the minimum bits needed to encode X [26]. $H(Y, X)$ is the joint entropy and $H(Y|X)$ is the conditional entropy between class Y and feature X.

The C4.5 algorithm for building decision trees is illustrated in Algorithm 1 [27]. In the algorithm, T represents the considered instances at each node. The chosen label at a leaf is set when only one class is present in the instances of a node or when there are no instances. In the last case, the chosen class is the most frequent one in the instances at the parent node. Another case is when only a few instances are present. Then, the class is set as the most frequent one, present in these instances. Note that these early stopping conditions try to prevent overfitting. Overfitting occurs when the model has high accuracy on the training data, but low accuracy on the validation data. Techniques such as pruning are further applied to prevent overfitting. Nodes

Algorithm 1: C4.5 Algorithm.

```
Input: Instances containing features X and classes Y.
Output: A classification decision tree.
ConstructTree(T):
if OneClass or FewCases then
|   return leaf;
else
|   create decision node N;
|   foreach attribute X do
|   |   ComputeGainRatio(Y,X);
|   end
|   N.test = AttributeWithHighestGain;
|   if N.test is continuous then
|   |   find threshold;
|   end
|   foreach splitted T' in T do
|   |   if T' is Empty then
|   |   |   child of N is leaf;
|   |   else
|   |   |   child of N = ConstructTree(T');
|   |   end
|   end
|   return N;
end
```

are replaced by one of their children nodes and the resulting accuracy with validation data is captured. Finally, the algorithm chooses the node which resulted in the most significant improvement on validation data. This method is called sub-tree replacement and is executed as long as the accuracy on validation data is increased [28]. As an alternative, C4.5 implementations do sometimes only use the largest subtree to replace its parent. We implemented pruning together with a maximum tree depth of 25 in order to maximise generalisation while reducing the tree size and thus minimising complexity.

Finally, to further improve the accuracy of decision trees, we have explored and used ensemble learning techniques such as random forests for our results to compare state-of-the-art decision tree methodologies. Random forests further prevent overfitting by generating multiple C4.5-generated decision trees, each trained with a random subset of features at each node to reduce correlation between the trees. Each tree votes for the predicted class. Finally, the most voted for class Y is chosen given input X [29].

5. Automatic feature learning based on raw IQ samples and image-based spectrograms

This section describes the automatic feature learning approaches that we have explored. Additionally, a description of CNNs, along with corresponding configuration details, is provided.

5.1. Feature learning

The approaches described in this chapter are based on supervised feature learning techniques which are heavily exploited in the computer vision domain. In this field, the manual feature extraction followed by dimensionality reduction (as in Section 4.1) is replaced by applying deep learning techniques directly on raw pixel intensities (e.g., the method proposed by the authors of [30]). Similarly, in our research, we apply FNN and CNNs on raw IQ values, their derived, simpler, RSSI samples and image-based spectrograms.

5.2. Convolutional neural networks

Table 5 provides an overview of the CNN architecture we adopted for the classification of wireless technologies. We started the design of our CNN architecture based on our previous work [31]. Next, we further improved generalisation to multiple locations and improved robustness to noise by experimentally fine tuning parameters as discussed further in this section. We implemented three types of CNNs based on their used data-type:

1. **RSSI-based CNN**: for training this CNN, we used RSSI samples, which are less complex than IQ samples. This CNN uses 256 RSSI samples as an input, which corresponds to 4.096 ms, similar to the sample length described in section 4.1.
2. **IQ-based CNN**: In this CNN, 4,096 raw IQ samples are used, which corresponds also corresponds to 4.096 ms. In Table 5 an input size of 8,192 is used because each IQ sample has two components.
3. **Image-based CNN**: The data used in this CNN are FFT IQ samples. Spectrograms are generated and saved as an image with dimensions 64 x 64 pixels. Again, this corresponds to 4.096 ms per input.

Compared to the FNN, described in section 4.3, the CNN includes many layers without typical neurons. These layers include functions to process

Table 5: CNN architectures with their corresponding data-type

RSSI-Based CNN		**IQ-Based CNN**		**Image-Based CNN**	
Layer	**Output dimensions**	**Layer**	**Output dimensions**	**Layer**	**Output dimensions**
Input	1 x 256	Input	2 x 4096	Input	64 x 64
Conv (64, 1x3), Relu	1 x 254 x 64	Conv (64, 1x2), Relu	2 x 4095 x 64	Conv (64, 2x2), Relu	63 x 63 x 64
Dropout	1 x 254 x 64	Dropout	2 x 4095 x 64	Max pooling (2,2)	31 x 31 x 64
Conv (16, 1x3), Relu	1 x 252x 16	Conv (32, 1x3), Relu	2 x 4093 x 32	Batch normalization	31 x 31 x 64
Max pooling (1,2)	1 x 126 x 16	Max pooling (1,2)	2 x 2046 x 32	Zero padding	35 x 35 x 64
Dropout	1 x 126 x 16	Dropout	2 x 2046 x 32	Conv (32, 1x3), Relu	35 x 33 x 32
Flatten	2016	Conv (16, 2x2), Relu	1 x 2045 x 16	Max pooling (2,2)	17 x 16 x 32
Dense (20), Relu	20	Max pooling (1,4)	1 x 511 x 16	Batch normalization	17 x 16 x 32
Dropout	20	Flatten	8176	Zero padding	21 x 20 x 32
Softmax	3	Dense(25), Relu	25	Conv (16, 1x3), Relu	21 x 18 x 16
		Dense(3)	3	Conv (4, 3x1), Relu	19 x 18 x 4
		Softmax	3	Conv (2, 2x2), Relu	18 x 17 x 2
				Flatten	612
				Dense (75)	75
				Dropout	75
				Dense (10), Relu	10
				Dropout	10
				Softmax	3

the output from previous layers. The first kind of processing layer is a convolutional layer. Such layers contain multiple learnable feature maps and calculate their values from various, but not all (such as in fully connected layers), previous neurons. They intend to have small receptive fields and decrease parameters by sharing filter weights [32]. The size of these feature maps varies in the convolutional layers, e.g., the first convolutional layer from the image-based CNN contains 64 feature maps with a size of 2 x 2, connected to neurons of the input layer. We experimented with increased stride sizes, which control the number of values the filter has to move. This is by default 1 by 1 so that each convolution connects all neighbours values within the convolutional filter size. However, increased stride sizes decreased the performance of the model. We believe this is due to features being more present locally and chronologically in our data. Increasing the stride size will decrease the number of local receptive fields, which results in lower performance.

Dropout is the next type of layer we used in our CNN. This layer produces more generalised models by preventing overfitting of training data, which we experimentally validated.

Pooling is another type of layer with the intent of reducing the total number of parameters to train on. This dimensionality reduction dramatically enhances training time and reduces the model's required memory footprint. In our case, we found optimal results with a max-pool size of $1X2$ and $2X2$. This pool will take 2 and 4 values, respectively, from the previous layer and output the maximum. Using max-pooling in the RSSI CNN, the number of trainable parameters decreased from 87,494 to 43,747.

The above mentioned layers are followed by a fully-connected dense layer. Each neuron of this layer is connected to all of the previous layer's neurons. This way, learned local features from previous convolutional layers get connected and are used to perform the final steps of classification.

The final layer contains three neurons, one for each class, and is activated with a softmax layer, as described in 4.3. In contrast to the FNN discussed in 4.3, each convolutional and dense layer is succeeded by a ReLU activation function. Here, this activation function performs slightly better than the radial basis activation function. The ReLU function, first proposed in [33], is defined in (10):

$$f(x) = max(0, x) \tag{10}$$

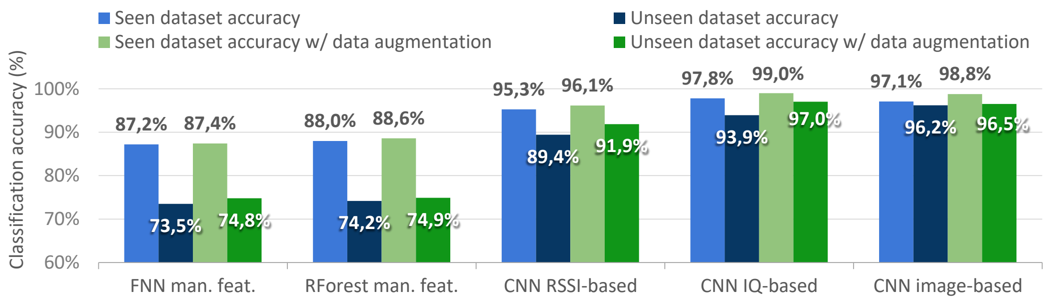

Figure 5: Results of manual and automatic feature learning approaches.

Finally, the models are trained for maximum 100 epochs using early stopping criteria (no loss improvement for 10 epochs) and a batch size of 256 samples.

6. Results and comparison

This section first presents results regarding accuracy in seen environments and afterwards generalisation towards unseen environments. Next, robustness towards additional noise levels is analysed, followed by a complexity analysis of the proposed approaches. All results are validated using 10-fold cross-validation to ensure there is no bias towards portions of the dataset and minimise variation of the results [34].

6.1. Accuracy

Results of the proposed approaches are presented in Figure 5. In this scenario, automatic feature learning with the CNN using raw IQ samples achieves the highest accuracy (97.8%), followed closely by the image-based CNN (97.1%) and the RSSI-based CNN (95.3%). Manual feature extraction methods achieve a slightly lower accuracy for both the FNN (87.2%) and the Random Forest (RForest) decision trees (88.0%). Figures 6a - 6e show the above results in more detail using confusion matrices. More specifically, accuracies for each correct classification and classification errors of Wi-Fi, LTE and DVB-T are shown. We observe classification errors to be the highest for Wi-Fi for manual RSSI based methods. Around 40% of Wi-Fi is identified as DVB-T. This leads to the conclusion that better features are needed to differentiate the two technologies. Despite these results, LTE classification seems to perform well across all models, even for less-complex manual feature extraction-based and raw RSSI-based methods. The IQ and image-based

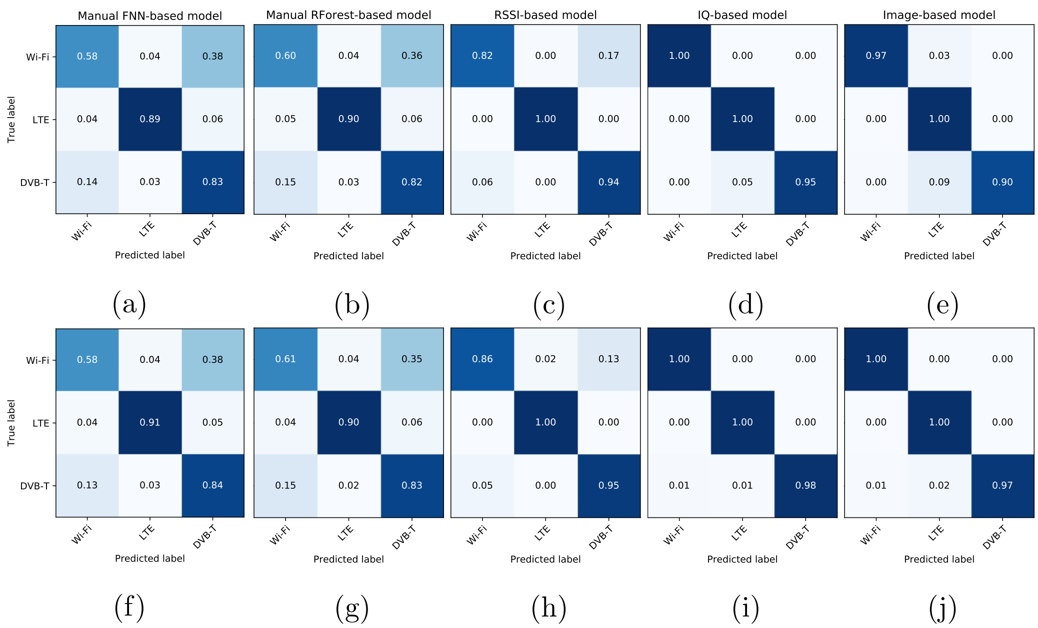

Figure 6: Above confusion matrices of all five approaches (a) Manual FFN model, (b) Manual RForest model, (c) RSSI CNN model (d) IQ CNN model (e) Image CNN model. Below confusion matrices with approaches using data augmentation including different SNR levels (f) Manual FFN model, (g) Manual RForest model, (h) RSSI CNN model (i) IQ CNN model (j) Image CNN model.

models clearly have superior performance as a result of the more complex models and feature-rich data.

6.2. Generalisability

The above results are only viable for environments that closely resemble those where the training data was collected. Therefore, we assess the generalisation of the models and validate the classification performance with a dataset from an unseen and different environment. Figure 5 shows for each approach lower accuracy on unseen datasets. This result is expected because the environment has other properties and captured signals are influenced in different ways. However, IQ- and image-based approaches still manage to achieve an accuracy above 93%, while the RSSI-based CNN achieves 89.4%. Manual feature extraction techniques struggle to generalise, exhibiting an accuracy just under 75%. This behaviour occurs because valuable information is lost through conversion of IQ samples to RSSI and further through manual extracted features.

To remedy this, we combined the techniques to improve generalisation and avoid overfitting discussed in 5.2, with additional data augmentation techniques. These techniques transform each sample of the dataset in various ways and add them to the original dataset. Specifically, we post-processed the seen dataset and included noise of different SNR levels, which is considered as a way of applying data augmentation techniques to IQ samples and RSSI values. Each sample is extended with noise, with SNR levels ranging from -15dB to +30dB with a step of 5dB. As a result, the original dataset size is increased by a factor of 10.

The results presented in Figure 5 illustrate accuracy improvements in all approaches through data augmentation, especially on the unseen dataset with the CNN using raw RSSI and IQ data (achieving and additional 2.5% - 3.1% generalisation increase). This leads to a very competitive scenario were RSSI, IQ and image-based CNN can be considered feasible for wireless technology classification. While manual feature extraction techniques show performance just under 90% in scenarios similar to those of the trained datasets, unseen scenarios keep struggling, with accuracies around 75%. These data augmentation techniques also show 1-7% improvement for single class classification accuracy on the seen dataset as shown in figures 6f - 6j.

6.3. Robustness

Next, we discuss the robustness of our proposed solutions against additional noise levels. Again, the models are trained with data containing SNR levels ranging from -15dB to +30dB. Validation results are collected for untrained samples in each SNR level. Figure 7 illustrates classification accuracy as a function of SNR. The image-based CNN achieves the highest accuracy overall, even in the low SNR scenario of -15dB. This is due to the fact that the image based CNN, which uses FFT of the IQ samples, is more immune to noise. As such the authors of [35] prove that such FFT frequency-based features surpass time-based features for wireless device identification in degraded SNR scenarios. Unsurprisingly, in high SNR scenarios it is clear that the automatic feature learning techniques outperform the manual feature extraction methods, which have limited features. Moreover, the similar and limited performance of the RForest and the FNN also hint to inferior feature extraction compared to their automatic extraction counterpart. Looking further at the results, the IQ-based CNN performs notably worse in low SNR scenarios. High sensitivity to noise by IQ samples is one possible underlying reason for this result. With these fluctuations in the dataset, the IQ-based

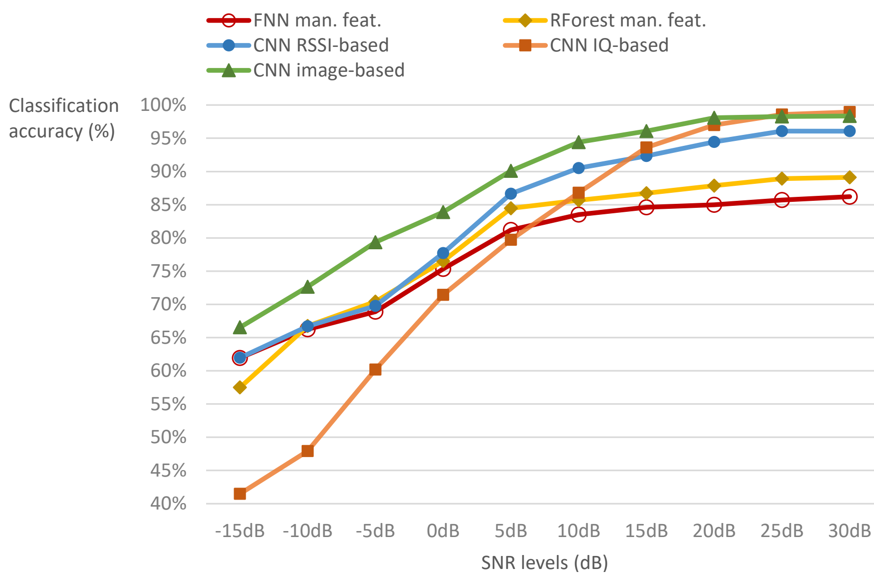

Figure 7: Classification accuracy as a function of SNR levels, for manual and automatic feature learning approaches.

CNN cannot learn to classify technologies in a reliable way. Results of the RSSI-based CNN further support this explanation because multiple IQ samples are averaged to become RSSI samples, as explained in equation 1 and are thus less susceptible to fluctuations due to added noise. As such, the input to the neural network has a much larger impact considering noise for IQ samples compared to RSSI samples. The RSSI-based CNN model achieves good performance, even in low SNR scenarios with an utmost difference of 10% compared to image-based CNN at -5dB, while performing only 3% less at high SNR scenarios compared to other automatic feature learning methods. These CNN-enabled methods prove to be robust from 10dB and upwards with accuracies ranging between 86% and 98%.

6.4. Complexity

Table 6 illustrates the complexity of the proposed approaches. Results are collected on a Windows computer with an Intel® Core^{TM} CPU i9-9900K @ 3.60GHz, NVIDIA® TITAN RTX^{TM} 24GB graphics card and 32GB of system memory. Manual feature extraction methods require less memory and are much faster in terms of training time. Moreover, the RSSI-based CNN achieves a much smaller memory footprint compared to the more complex

Table 6: Trainable weights, memory footprint and training time of the proposed approaches

Model	**Weights**	**Memory**	**Train time**
RForest man. feat.	6393	0.08GB	19s
FNN man. feat.	1018	0.12GB	51s
CNN RSSI-based	43747	0.81GB	100s
CNN IQ-based	212935	8.46GB	1500s
CNN Image-based	55430	2.61GB	950s

IQ- and image-based methods. One of the reasons is the 16 times smaller input size. The IQ- and image-based methods require high-end GPUs to train on. Furthermore, because of their high number of weights, layers and convolutions, they require more resource-heavy systems to deploy as wireless technology classification systems. Although IQ-based models require most resources, we want to highlight that these model require no pre-processing. This makes the model very interesting compared to image-based models which require computational-heavy FFT and image generation capabilities. This pre-processing can limit the feasibility when the model is deployed for wireless classification.

As a conclusion, manual feature extraction methods are very resource-friendly, but only perform well in known environments. Automatic feature learning methods perform better, especially in terms of generalisation. On the one hand the RSSI-based CNNs show great efficiency potential with their relative small memory footprint and high accuracy. On the other hand, IQ- and image-based methods achieve the highest prediction accuracies no matter their resource requirements.

7. Conclusions and future work

Machine learning techniques show enormous potential in many domains, including wireless technology classification. In this domain, due to increasing heterogeneity in wireless communications, often sharing the same spectrum band, sensing the environment and making intelligent decisions is crucial. Many of the previous works present deep learning approaches to successfully identify wireless technologies on the fly. However, many of the proposed methods target only resourceful devices and fail to address generalised and robust models for different environments with changing noise levels.

In this paper, we have proposed and evaluated techniques to allow wireless technology classification for resource-constrained devices, as well as for more resourceful devices. Furthermore, we have shown that data augmentation techniques add an additional boost to generalisation, next to various model design choices, for unknown environments up to 3.1%. We have demonstrated that applying FFT algorithms to IQ samples, to further create image-based spectrograms, enables high accuracy, even in lower SNR scenarios. Raw IQ files achieve the highest generalisation capabilities by achieving the highest accuracy in unseen environments. Finally, manual feature extraction proved to be inferior compared to automatic feature learning in terms of accuracy, but can still be useful in known environments, while requiring very low complexity. Moreover, the less complex RSSI-based model offers a good balance between complexity, accuracy, generalisation and robustness to noise. These results demonstrate the positive effect of choosing the correct machine learning technique and data format. As such, the outcome of this paper enables wireless domain experts to incorporate intelligence into wireless communications using machine learning techniques while targeting multiple environments and recommends multiple approaches for wireless technology classification.

We envision future research adding support for overlapping signals. This will enrich the models support for irregular signal behaviour and prevent misclassification for these kind of signals. Additionally, autoencoders can be used for semi-supervised learning, minimising the required amount of labelled data that is needed. This will further accelerate the adoption of new supported technologies in many environments. Furthermore, future work can make intelligent decisions for wireless technology operators based on the detected present technologies. Finally, models with even lower complexity should be developed with a small accuracy-complexity trade-off, reducing the operational costs of future intelligent devices.

[1] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of Things (IoT): A vision, architectural elements, and future directions, Future generation computer systems 29 (2013) 1645–1660.

[2] H.-J. Kwon, J. Jeon, A. Bhorkar, Q. Ye, H. Harada, Y. Jiang, L. Liu, S. Nagata, B. L. Ng, T. Novlan, et al., Licensed-Assisted Access to Unlicensed Spectrum in LTE Release 13, IEEE Communications Magazine 55 (2017) 201–207.

[3] R. Zhang, M. Wang, L. X. Cai, Z. Zheng, X. Shen, L.-L. Xie, LTE-unlicensed: the future of spectrum aggregation for cellular networks, IEEE Wireless Communications 22 (2015) 150–159.

[4] Incorporated, The 1000x data Challenge (2013).

[5] R. H. Tehrani, S. Vahid, D. Triantafyllopoulou, H. Lee, K. Moessner, Licensed spectrum sharing schemes for mobile operators: A survey and outlook, IEEE Communications Surveys and Tutorials 18 (2016) 2591–2623.

[6] ETSI, Building the future, work programme 20142015, Technical Report, 2014.

[7] 3GPP, Study on radio access network (RAN) sharing enhancements, Technical Report, 2014.

[8] Y. j. Choi, C. S. Kim, S. Bahk, Flexible Design of Frequency Reuse Factor in OFDMA Cellular Networks, in: IEEE International Conference on Communications, volume 4, pp. 1784–1788.

[9] J. Huschke, W. Rave, T. Kohler, Downlink capacity of UTRAN reusing frequencies of a DVB-T network with negligible influence on DVB-T performance, in: IEEE Vehicular Technology Conference, volume 3, pp. 1579–1583.

[10] B. Ellingster, H. Bezabih, J. Noll, T. Maseng, Using TV receiver information to increase cognitive white space spectrum, in: IEEE International Symposium on Dynamic Spectrum Access Networks, pp. 131–141.

[11] L. Zhou, S. Pan, J. Wang, A. V. Vasilakos, Machine learning on big data: Opportunities and challenges, Neurocomputing 237 (2017) 350 – 361.

[12] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (2013) 1798–1828.

[13] M. Schmidt, D. Block, U. Meier, Wireless Interference Identification with Convolutional Neural Networks, CoRR abs/1703.00737 (2017).

[14] A. Selim, F. Paisana, J. A. Arokkiam, Y. Zhang, L. Doyle, L. A. DaSilva, Spectrum Monitoring for Radar Bands Using Deep Convolutional Neural Networks, in: IEEE Global Communications Conference, pp. 1–6.

[15] W. Liu, M. Kulin, T. Kazaz, A. Shahid, I. Moerman, E. De Poorter, Wireless Technology Recognition Based on RSSI Distribution at Sub-Nyquist Sampling Rate for Constrained Devices, Sensors 17 (2017).

[16] T. J. O'Shea, J. Corgan, Convolutional Radio Modulation Recognition Networks, International Conference on Engineering Applications of Neural Networks abs/1602.04105 (2016).

[17] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, S. Pollin, Distributed Deep Learning Models for Wireless Signal Classification with Low-Cost Spectrum Sensors, IEEE Transactions on Cognitive Communications and Networking abs/1707.08908 (2017).

[18] M. Zhang, M. Diao, L. Guo, Convolutional neural networks for automatic cognitive radio waveform recognition, IEEE Access 5 (2017) 11074–11082.

[19] M. Kulin, T. Kazaz, I. Moerman, E. De Poorter, End-to-end learning from spectrum data : a deep learning approach for wireless signal identification in spectrum monitoring applications, IEEE Access 6 (2018) 18484–18501.

[20] G. Aceto, D. Ciuonzo, A. Montieri, A. Pescap, Multi-classification approaches for classifying mobile app traffic, Journal of Network and Computer Applications 103 (2018) 131 – 145.

[21] A. Pescape, A. Montieri, G. Aceto, D. Ciuonzo, Anonymity services tor, i2p, jondonym: Classifying in the dark (web), IEEE Transactions on Dependable and Secure Computing (2018) 1–1.

[22] G. Aceto, D. Ciuonzo, A. Montieri, A. Pescap, Mobile encrypted traffic classification using deep learning, in: 2018 Network Traffic Measurement and Analysis Conference (TMA), pp. 1–8.

[23] eWINE-project, Iq samples of lte and wifi, `https://github.com/ewine-project/lte-wifi-iq-samples`, 2017.

[24] I. Guyon, A. Elisseeff, An Introduction to Variable and Feature Selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.

[25] L. Jing, M. Zhao, P. Li, X. Xu, A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox, Measurement 111 (2017) 1 – 10.

[26] C. R. Shalizi, Methods and Techniques of Complex Systems Science: An Overview, Springer US, pp. 33–114.

[27] S. Ruggieri, Efficient C4.5 [classification algorithm], IEEE Transactions on Knowledge and Data Engineering 14 (2002) 438–444.

[28] Salvatore Ruggieri, Subtree Replacement in Decision Tree Simplification, in: SDM.

[29] L. Breiman, Random forests, Machine Learning 45 (2001) 5–32.

[30] J. Schmidhuber, Multi-column Deep Neural Networks for Image Classification, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Washington, DC, USA, 2012, pp. 3642–3649.

[31] V. Maglogiannis, A. Shahid, D. Naudts, E. De Poorter, I. Moerman, Enhancing the coexistence of lte and wi-fi in unlicensed spectrum through convolutional neural networks, IEEE Access 7 (2019) 28464–28477.

[32] S. Lawrence, C. L. Giles, A. C. Tsoi, A. D. Back, Face recognition: a convolutional neural-network approach, IEEE Trans. Neural Netw. 8 (1997) 98–113.

[33] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, H. S. Seung, Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit, Nature 405 (2000) 947.

[34] Y. Zhang, Y. Yang, Cross-validation for selecting a model selection procedure, Journal of Econometrics 187 (2015) 95 – 112.

[35] B. Danev, S. Capkun, Transient-based identification of wireless sensor nodes, in: 2009 International Conference on Information Processing in Sensor Networks, pp. 25–36.