Wireless Technology Identification Using Deep Convolutional Neural Networks

Naim Bitar, Siraj Muhammad, Hazem H. Refai.
School of Electrical and Computer Engineering
The University of Oklahoma
Tulsa, USA.
Email: {naimbitar, sirajmuhammad, hazem}@ou.edu

***Abstract*—With the proliferation of wireless technologies and the ever-increasing growth in Internet of Things (IoT) devices operating the license-free Industrial, Scientific, and Medical (ISM) band, intelligent access systems capable of coexisting in crowded spectrum regions are of vital importance.**

In this work we study the adaptation of Convolutional Neural Networks (CNNs) to the problem of identifying coexisting wireless devices. We develop a machine learning conduit to facilitate the detection and identification of frequency domain signatures for 802.x standard compliant technologies. Spectrum scans across the entire ISM region (80-MHz) are recorded and a data-driven training process for a wide range of Signal-to-Noise Ratios (SNRs) is completed. Model accuracy is compared to that attained using standard feature based classification methods. Results indicate CNN models outperform their counterpart methods in terms of classification accuracy, connoting them to be highly effective tools for detecting and identifying coexisting devices despite acute overlap and interference presence. The proposed approach aims to advance cognitive wireless awareness by enhancing automatic detection and identification accuracy.

***Index Terms*—CNN, Machine Learning, Neural Networks, Cognitive Radio, Spectrum Sensing, Wireless Identification.**

I. Introduction

Wireless communication networks have become ubiquitous in today's modern world. Technologies such as (WiFi, ZigBee, Bluetooth,...) share the 2.4 GHz ISM band mainly because it is unlicensed and free. These various technologies compete for resources and attempt to coexist and operate harmoniously. With limited spectrum resources and exposure to different levels of interference among competing technologies, successful coexistence is imperative for adequate wireless communication.

Studying spectrum occupancy is required to identify vacant channels. Characterizing occupancy and identifying operating technologies in a frequency band improves the probability of successful communication. Moreover, identifying technologies utilizing specific channels aids in modeling interference encountered when attempting access to the medium. Primarily, intelligent Cognitive Radio (CR) systems require correct spectrum usability assessment, and situational awareness prior to accessing licensed frequency bands. As the probability of interference increases in heterogeneous coexisting networks [13], identification of wireless technologies operating a specific spectrum band provides a cognitive node with an estimate of potential interference. Furthermore, wireless operators anticipating LTE-U deployments alongside 802.x standard compliant technologies in the unlicensed band, medical devices utilizing transceivers working under stringent interference requirements, and next generation Local Area Networks (LANs) employing Self-Organizing Network (SON) engines can all benefit from enhanced spectral situational awareness.

In this domain, efforts have been mostly constrained to expert crafted features and decision criterion as in [3]–[5], [8], [10]. Many of these solutions are relatively specific and lack the universality needed to cope with an increasing number of transmitter types and the wide variety of complex propagating environments. However, recent trends in machine learning applied to classification problems have proved to be extremently successful in distinguishing random and complex patterns. In particular, Convolutinal Neural Networks (CNNs) in computer vision have offered a compeling alternative to automatic learning of problem-specific features [19] exploting the universal approximation characteristics of neural networks [1].

In this work we investigate the adaptation of CNNs to the problem of identifying coexisting wireless devices. We demonstrate improved classification accuracy against current day approaches.

The balance of this paper is organized as follows. Section II presents related work. Section III discuses data collection and feature extraction. Section VI delineates the adapted deep learning framework. Section V presents and discusses the results attained. Finally, section VI concludes the paper.

II. Related Work

In this section, we briefly review the current research progress and present a concise summary of approaches used to identify wireless technologies as found in literature.

978-1-5386-3531-5/17/$31.00 © 2017 IEEE

TABLE I
LITERATURE SEARCH SURVEY

Method/Ref	[13]	[3]	[4]	[5]	[9]	[10]	[12]	[8]	[18]	[17]	[14]	[11]	[7]	[2]
MAC Layer	✓	✓				✓	✓							
PHY Layer		✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓
Learning technique	NB - KNN	Distribution Analysis	HMM	RF fingerprinting	RF fingerprinting	Distribution Analysis	SVM	FL	CNN - DNN	CNN	Neural Network	Neural Network	CRP Clustering	Heirarchical clustering + KNN

Cognitive radios rely on information gathered during spectral sensing to foster communication between protocol-disparate devices. Clearly, an essential requirement is accurate service discovery. To this end, several approaches have been explored spanning processes like matched filter, cyclostationary and energy detection techniques. Methods can also be grouped into one of two types based on the information utilized in the analysis: 1)- MAC layer characteristic information or 2) - PHY Layer characteristic information. Table I summarizes the methods and the learning techniques applied per reference.

A. MAC Layer methods

Authors in [12] utilized power and time features to distinguish between four MAC protocols (TDMA, CSMA/CA, pure ALOHA, and slotted ALOHA) using Support Vector Machines (SVMs). Similary, authors in [13] captured MAC layer temporal characteristics of 802.11 b/g/n homogeneous and heterogeneous networks and used a K-Nearest Neighbor (KNN) and a Naive Bayes (NB) Classifier to distinguish between all three. Maximum accuracy of 85% was achevied with the NB classifier. Authors in [10] analyzed the distribution of inter-arrival times of packets as a parameter for identical 802.11 device identification. While much can be gained by analyzing the MAC layer, it may be more advantageous to consider raw PHY layer information as the data processing inequality means there is more information contained in a PHY layer waveform [3].

B. PHY Layer methods

Accordingly, authors in [3] using narrow-band sampling exploited protocol-specific differentiators consisting of time burst duration, bandwidth and periodicity of transmissions to identify distinct WiFi and Bluetooth networks and devices. Authors in [4] classified second-order cyclic Orthogonal Frequency Division Multiplexing (OFDM) features using hidden Markov Models (HMMs) to differentiate between 802.11 a/g signals. In contrast, researchers in [5] attempted to distinguish between identical 802.11 Network Interface Cards (NICs) leveraging the Radio-Frequency (RF) fingerprint of repeated patterns of minute imperfections of the transmitter hardware. Similarly, researchers in [9] utilized transient features that are device specific of wireless transmitters when turned on and off to differentiate between identical transmitters of 802.11b. Authors in [8] used Power Spectral Density (PSD) measurements as sensing information to label signals. They extracted time and frequency behavior features along with carrier frequency and bandwidth and used a fuzzy logic (FL) signal classifier to identify 802.11 WLAN, Bluetooth and two other proprietary networks. Their method utilized pre-defined features and a fixed decision process that yielded good results for sufficiently high duty cycle transmissions as reported, experiencing classification errors with frequency-hopping systems such as Bluetooth in narrow-band analysis. Authors in [18] were the first to utilize Deep Neural Networks (DNNs) and CNNs to identify 11 different modulation techniques. They were able to distinguish extracted time domain activity signatures with an accuracy of roughly 87%. Authors in [11], [14] utilized neural networks and power frequency measurements to predict spectral holes (inactivity periods) present in the band. Authors in [17] proposed a CNN architecture to improve this work and reported an enhanced accuracy of above 95%. Authors in [7] identified spectrum usage in a heterogeneous radio environment of WiFi and ZigBee transmitters. They captured two features; center frequency and frequency spread. Using the Chinese Restaurant Process the researchers performed unsupervised clustering. They achieved results ranging between 80% - 90% accuracy for clustering WiFi and ZigBee. Authors in [2] used single-link hierarchical clustering to cluster Bandwidth (BW), Center Frequency (CF) and temporal width (TW) of 802.11b,g and Bluetooth, then using Nearest Neighbor classifier matched input signals feature triplets (BW, TW, FC) to the existing cluster centroids.

The majority of the aforementioned methods require sophisticated signal processing to enable extraction of the salient features, and require powerful radios equipped with high sampling ADCs to trace fast hopping systems. Furthermore, they require carefully crafted technology specific features to be hard-coded into the detection algorithm. These features while suitable for distinguishing some activities or signatures may perform poorly when a new system is added to the environment. Therefore, one would have to manually re-design handcraft features for specific applications.

The paradigm of cognitive systems emphasizes a connectionist classifier that requires the fewest assumptions and is most applicable to a problem in which a priori information might not be available. Consequently, machine learning is a potential candidate for such a solution.

In this work we attempt to apply Convolutional Neu-

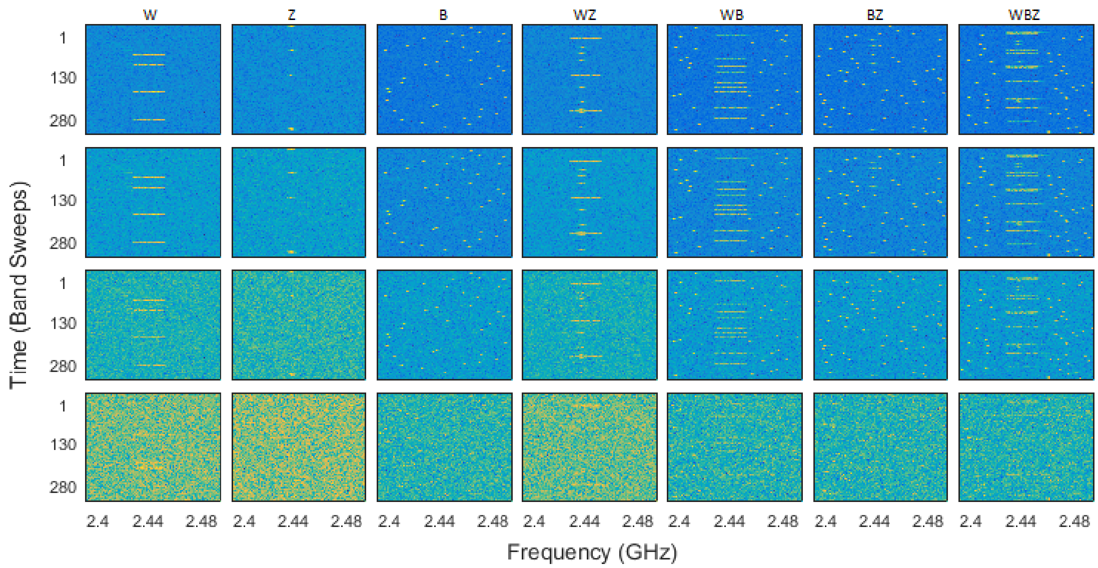

Fig. 1. Spectogram of Wireless Technologies. 1^{st} row SNR = 30 dB, 2^{nd} row SNR = 20 dB, 3^{rd} row SNR = 10 dB and 4^{th} row SNR = 0 dB

ral Networks to distinguish between a heterogeneous WLAN network composed of Bluetooth, ZigBee and 802.11n. The proposed solution aims to enable an automatic feature extracting system capable of identifying operating standards without the need to craft technology specific features in the model. In this work we show that despite acute overlap and interference, CNNs are capable of achieving considerable improvement over traditional machine learning techniques.

III. Data Collection and Feature Extraction

Data collection was performed in a semi-anechoic chamber to isolate unwanted external noise. Power measurements across the entire 80 MHz spectrum were recorded. A resolution frequency of approximately 285 KHz was chosen. This required 280 adjacent scans to sweep the entire band. The choice was made to simulate the capabilities of lower-end spectral scanners opposed to high-end radios with powerful Analog-to-Digital Converters (ADCs) capable of sampling the entire band at once. The device used was the National Instruments (NI) PXIe-1075 chassis equipped with a PXIe-5663 vector network analyzer. The sweep time was around 5.3 ms. The network under study was composed of a heterogeneous deployment of WiFi 802.11n, ZigBee and Bluetooth. WiFi was generated using two Mikrotik Router boards (RB953GS) with R11e-2HPnD radio card boards on channel 6 with center frequency of 2.437GHz. ZigBee traffic was generated using the CC2530 development kit board over channel 17 of center frequency 2.435 GHz overlapping WiFi's transmissions. Two laptops were used to send files over Bluetooth 4.0 technology. Communication transmissions pertaining to different classes were captured with an SNR range of 0 - 30 dB. Homogeneous cases of: WiFi (W) ZigBee (Z), Bluetooth (B) and Heterogeneous co-interfering cases of: Wifi-ZigBee (WZ), Wifi-Bluetooth (WB), Bluetooth-ZigBee (BZ) and WiFi-Bluetooth-ZigBee (WBZ) were all recorded. Power frequency spectograms visualizing captured transmissions are shown in figure 1. The first row depicts all seven classes at 30 dB. For brevity, transmissions at 20, 10 and 0 dB are only shown in rows 2, 3 and 4 consecutively. It is observed at 0 dB, noise is predominant, rendering transmissions imperceptible. The time-frequency power values of all SNRs captured were then fed into the CNN classifier as one large training set, described in the following section, as features for training. In total, around 4.3 million sweeps were taken generating 15400 samples each containing 78400 features and a single aforementioned label.

IV. Learning Architecture

A CNN architecture was developed to train and classify recorded signals. Power-frequency values obtained from the In-Phase and Quadrature (I & Q) components extracted were fed as inputs into the convolutional neural network taking each sample consisting of 78400 features as a 2D matrix of size 280x280. Pre-processing normalization and scaling was performed on the input data. This step was crucial as training and testing were not done for each SNR seperately, but rather the entire SNR range of samples were shuffled and input as one large set for training. This enabled testing classification accuracy for any random SNR input which mimics SNR fluctuations observed in real wireless environments. Furthermore, normalization was key to alleviate the affect of heterogeneous SNR values between technologies. Dropout was used to prevent over-fitting. Training was conducted

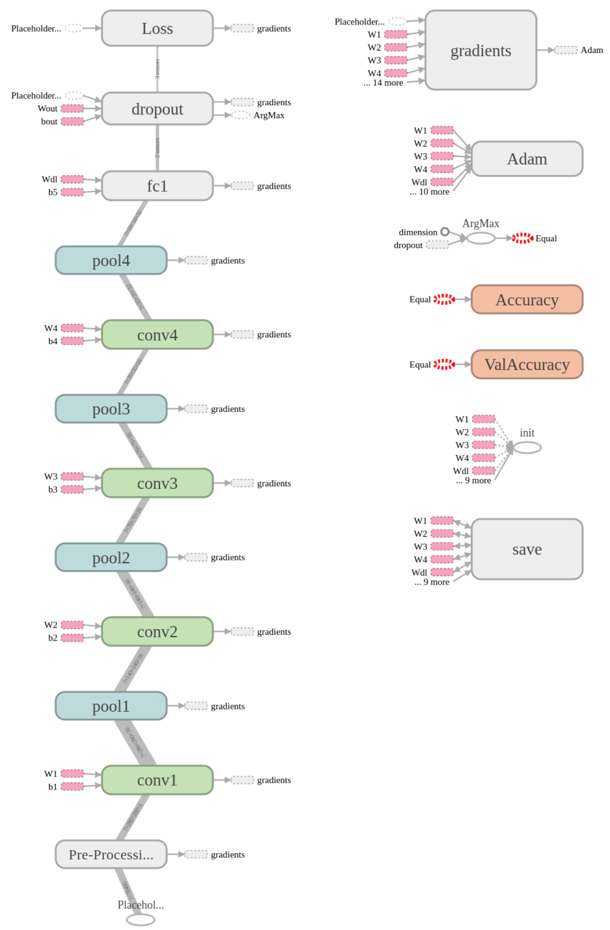

Fig. 2. Convolutional Neural Network Structure.

using a categorical cross entropy loss function along with an Adam optimizer [15]. Implementation, training and prediction were done in the open source tensorflow software library [16]. Training was run on an NVIDIA Cuda [6] enabled GeForce GTX 770.

A. Network Structure

The developed CNN structure consists of 5 transformation layers begining with four consecutive convolutional and pooling layers followed by one dense fully connected layer. Figure 2 depicts the complete network structure using tensorboard [16]. The figure illustrates input and output array sizes along with the pipeline of operation. Table II details the model structure and filter sizes.

The model input is a matrix generated from the power per frequency values containing time-frequency characteristics of each technology occupying the band. The model input can be written as:

$$X_s = [x_{i,j}, ..., x_{N,M}],\\ i \in [1, N], j \in [1, M], s \in [1, S], \tag{1}$$

where s is the sample index. S is the total number of samples captured (15400). N is the total number of

TABLE II
CNN STRUCTURE

Layer Type	Input Size	Filter Size	Activation Function
Convolutional Layer (1)	280 x 280	3 x 3 x16	Relu
Convolutional Layer (2)	140 x 140	5 x 5 x16	Relu
Convolutional Layer(3)	70 x 70	7 x 7 x16	Relu
Convolutional Layer(4)	35 x 35	7 x 7 x 32	Relu
Fully Connected Layer	10368 x 1	1024 neurons	Softmax

sweeps in one sample (280), and M is the total number of resolution channels in the captured band (280). The extraction of features is the combination of convolutional and pooling layers, and is the core part of the CNN model. The pooling procedure is indicated here as a function named "pool". Accordingly, we write the output of the first convolution and pooling layer as:

$$O_1 = pool\Big(\sigma\big(W_1^v * X_s + b_1^v\big)\Big), v \in [1, c_1], \tag{2}$$

where c_1 denotes the number of convolutional filters in the 1_{st} layer, σ denotes the activation function and W, b are weight and bias parameters utilized per layer. The activation function transforms the output to a manageable and scaled data range which is beneficial to model training. Furthermore, the combination of the activation function through layers can mimic very complex and non-linear functions. For this purpose, the ReLu activation function was adopted, defined as follows:

$$\sigma(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & otherwise. \end{cases} \tag{3}$$

Therefore, assuming $O_0 = X_s$, the output in the l_{th} convolutional and pooling layers can be written as:

$$O_l = pool\Big(\sigma\big(W_l^v * O_{l-1} + b_l^v\big)\Big), v \in [1, c_l], l \in [1, L-1], \tag{4}$$

where L is the total length of our proposed CNN architecture. Characteristics of the feature extraction process are two properties: (a) Convolution and pooling learn the frequency-time relationships in terms of the prediction task in model training; (b) the features learnt and output by the convolutional layers are concatenated into a dense vector that contains the final and most high-level features of the input. The dense vector can be written as:

$$O^f = f(O_{L-1}), \tag{5}$$

where f is a flattening function designating the aforementioned concatenation procedure. Finally, the vector is transformed into model outputs through a fully connected layer. The model output can thus be written as:

$$\hat{y_s} = W_{fu} * \Bigg(f\bigg(pool\Big(\sigma\big(W_l^v * O_{l-1} + b_l^v\big)\Big)\bigg)\Bigg) + b_{fu}\\ s \in [1, ..S], \tag{6}$$

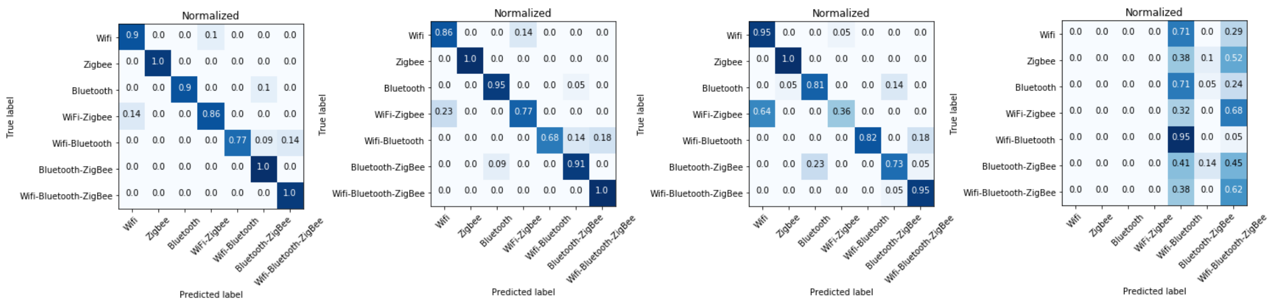

Fig. 3. Confusion Matrix Comparison at SNRs = 30, 20, 10 and 0 dB Consecutively

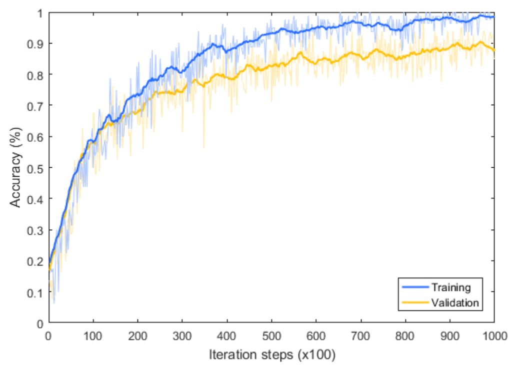

Fig. 4. Training and Validation Accuracy.

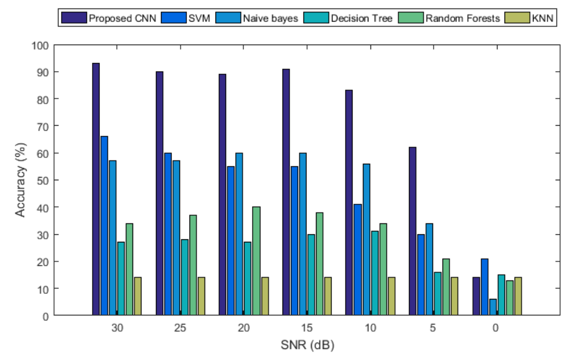

Fig. 5. Testing Accuracy Comparison

where W_{fu}, b_{fu} are weight and bias parameters of the fully connected layer and $\hat{y_s}$ is the one-hot predicted technology score per sample s. The predictions of the CNN model are the Identification (ID) of each technology operating the band.

B. CNN Optimization

The categorical cross entropy loss function was employed on the one-hot final output layer to measure the difference between predictions and the actual class. Thus, minimizing the loss is taken as the training goal of the CNN.

$$\text{Total Loss} = \frac{1}{S}\sum_{s=1}^{S}\Big(-log\Big(\Psi_i(\hat{y}_s)\Big)\Big), \tag{7}$$

$$\Psi_i(\hat{y}) = \frac{e^{\hat{y}_i}}{\sum_k e^{\hat{y}_k}},$$

where Ψ_i is the softmax function. $\hat{y}_i$ represents the score value of the correct one-hot encoded class.

$$\Theta = \underset{\Theta}{\text{argmin}}\Bigg(\frac{1}{S}\sum_{s=1}^{S}\Bigg(-log\Big(\frac{e^{\hat{y}_{s\,i}}}{\sum_k e^{\hat{y}_{s\,k}}}\Big)\Bigg)\Bigg). \tag{8}$$

Conclusively, training was done on a batch size of 80 samples, and the learning rate was set to 0.001.

V. Results

A. Training Results

Figure 4 illustrates the accuracy attained during the training phase which consisted of 100,000 iteration steps. Peak training accuracy achieved was 100% for training, and 96.2% for validation. This model was saved and utilized for testing. Row one in table III illustrates accuracy attained at SNRs (30, 25, 20, 15, 10, 5 and 0) dB for the proposed CNN model. A confusion matrix illustrating prediction error per class is shown in figure 3. We observe CNN achieved very good results for most cases. ZigBee was found to be the main culprit for misclassification error due to co-channel operation with WiFi. This was evident in the confusion matrices illustrated indicating errors in (WiFi) vs. (WiFi-ZigBee), and (WiFi-Bluetooth) vs. (WiFi-Bluetooth-ZigBee). Moreover, (ZigBee-Bluetooth) and (Bluetooth) at low SNRs reported classification errors due to Bluetooth's hopping sequence overlap with ZigBee, which at low SNR, made ZigBee's 5-Mhz signature indistinguishable to Bluetooth's 1-MHz bandwidth.

B. Comparison Analysis & Discussion

To compare the results attained from the proposed classifier with other machine learning methods, we train and classify the same input features on 5 popular classification methods. We perform different optimizations per method in order to find the best parameters i.e. (kernel, degree of polynomial, distance metric, weight, ... etc) that yields the highest accuracy result. Table III and figure 5 depict the results attained. It is observed that the CNN architecture outperformed all other methods with a large margin in terms of classification accuracy. It achieved a testing accuracy of 93% on the highest SNR

TABLE III
CLASSIFICATION RESULT COMPARISON

Method	30dB	25dB	20dB	15dB	10dB	5dB	0dB
Proposed CNN	93%	91%	90%	91%	83%	62%	14%
SVM	66%	66%	65%	65%	64%	30%	21%
Naive Bayes	57%	57%	60%	60%	56%	34%	6%
Random Forests	34%	37%	40%	38%	34%	21%	13%
Decision Tree	27%	28%	27%	30%	31%	16%	15%
KNN	14%	14%	14%	14%	14%	14%	14%

of 30 dB, and a comparable result of around 91% for SNRs between 15-25 dB. Accuracy began to deteriorate around 10 dB, and eventually reached blind guessing at 0 dB. This was caused by the high levels of noise present at this SNR which completely obscures all signal features in the spectogram as observed in figure 1, row 4. An SVM classifier with a polynomial kernel of degree 5 achieved the second best accuracy of around 66%. A Naive Bayes classifier with a multinomial distribution model came in third with a stable overall performance over the 30-10 dB range. Random forest, decision trees and KNN classifiers all performed below satisfactory and required considerable training time. The reason we believe this occurred is due to the large dimensionality of the 78400 features per sample. KNN was the least accurate method of all, achieving results kin of blind guessing around 14%. The proposed CNN architecture and SVM required the lengthiest training time compared to all other classification methods; furthermore all methods have a near instantaneous testing time. However, the proposed CNN outperformed all in terms of accuracy. Finally, the developed 5-layer CNN architecture was the result of an extensive testing examination of many architecture depths and layer filter sizes. It was found that a deeper network did not correlate with a higher classification accuracy. Thus, higher complexity does not achieve improved accuracy in this case. Finding the optimal depth and filter sizes remain an open research problem in deep learning and is beyond the scope of this paper.

VI. CONCLUSION

With the continuing growth of devices operating the ISM license-free bands, reliable coexistence management and adequate wireless identification is vital. In this work we proposed the adaptation of Convolutional Neural Networks to the problem of identifying 802.x standard compliant technologies operating the ISM band. Testing showed CNN was capable of learning power (time-frequency) features during an extensive data-driven training process with an accuracy of 93% using a proposed 5-layer CNN architecture. Furthermore, the CNN model exhibits improved and promising results across a large range of SNR values compared to traditional machine learning techniques. The proposed approach enables automatic feature detection of operating devices and allows low-end spectral scanners to achieve good prediction accuracy despite acute co-interference present. Future work will investigate throughput and occupancy levels and their effect on the CNN model. Finally, further CNN optimization will be investigated to possibly improve results.

REFERENCES

[1] Balázs Csanád Csáji, *Approximation with Artificial Neural Networks*. Faculty of Sciences; Eötvös Loránd University, Hungary, 2001.

[2] A. N. Mody et al, *Recent advances in cognitive communications*. IEEE Wireless IEEE Communications Magazine, vol. 45, no. 10, pp. 54-61, October 2007.

[3] R. Miller, W. Xu, P. Kamat, W. Trappe, *Service Discovery and Device Identification in Cognitive Radio Networks*. 2007 2nd IEEE Workshop on Networking Technologies for Software Define Radio Networks, pp. 40-47, 2007.

[4] K. Kim, C. M. Spooner, I. Akbar, J. H. Reed, *Specific Emitter Identification for Cognitive Radio with Application to IEEE 802.11*. EEE GLOBECOM 2008–2008 IEEE Global Telecommunications Conference, pp. 1-5, 2008.

[5] V. Brik, S. Banerjee, M. Gruteser, S. Oh, *Wireless device identification with radiometric signatures*. Proceedings of the 14th ACM international conference on Mobile computing and networking - MobiCom '08, pp. 116, 2008.

[6] John Nickolls, Ian Buck, Michael Garland, Kevin Skadron, *Scalable Parallel Programming with CUDA*. ACM Queue, vol. 6 no. 2, March/April 2008, Pages 40-53.

[7] N. Shetty, S. Pollin and P. Pawelczak, *Identifying Spectrum Usage by Unknown Systems using Experiments in Machine Learning*. IEEE Wireless Communications and Networking Conference, Budapest, 2009, pp. 1-6.

[8] K. Ahmad, G. Shresta, U. Meier and H. Kwasnicka, *Neuro-fuzzy signal classifier (NFSC) for standard wireless technologies.* 7th International Symposium on Wireless Communication Systems, York, 2010, pp. 616-620.

[9] Z. Shi, M. Liu, L. Huang, *Transient-based identification of 802.11b wireless device*. International Conference on Wireless Communications and Signal Processing (WCSP), pp. 1-5, 2011.

[10] C. Neumann, O. Heen, S. Onno,, *An Empirical Study of Passive 802.11 Device Fingerprinting*. 32nd International Conference on Distributed Computing Systems Workshops, pp. 593-602, 2012.

[11] S. Pattanayak, R. Nandi and P. Venkateswaran, *Identification of spectrum holes using ANN model in TV bands with AWGN*. IEEE Asia Pacific Conference on Wireless and Mobile, Bali, 2014, pp. 188-193, 2014.

[12] S. Hu, Y. Yao, Z. Yang, *MAC protocol identification using support vector machines for cognitive radio networks*. IEEE Wirel. Commun., vol. 21, no. 1, pp. 52-60, Feb. 2014.

[13] S. A. Rajab, W. Balid, M. O. Al Kalaa and H. H. Refai, *Energy detection and machine learning for the identification of wireless MAC technologies*. 2015 International Wireless Communications and Mobile Computing Conference (IWCMC), Dubrovnik, 2015, pp. 1440-1446.

[14] S. Bai, X. Zhou and F. Xu, *Spectrum prediction based on improved-back-propagation neural networks*. 11th International Conference on Natural Computation (ICNC), Zhangjiajie, 2015, pp. 1006-1011.

[15] Diederik P. Kingma, Jimmy Ba, *Adam: A Method for Stochastic Optimization*. 3rd International Conference for Learning Representations, San Diego, 2015.

[16] M. Abadi, A. Agarwal, et al., *Tensorflow: large-scale machine learning on heterogeneous systems, 2015*. Software available from tensorflow.org

[17] Benjamin Neel, Samuel North, Marc Messier, Birsen Sirkeci-Mergen, *Spectrum Sensing using USRP SDRs and Convolutional Neural Networks*. 15th Int'l Conf on Wireless Networks ICWN'16, 2016.

[18] O'Shea T.J., Corgan J., Clancy T.C., *Convolutional Radio Modulation Recognition Networks*. Jayne C., Iliadis L. (eds) Engineering Applications of Neural Networks. EANN 2016. Communications in Computer and Information Science, vol 629. Springer, Cham

[19] Suraj Srinivas et al, *A Taxonomy of Deep Convolutional Neural Nets for Computer Vision*. Frontiers in Robotics and AI 2(36), January 2016.