

Received March 18, 2019, accepted April 1, 2019, date of publication April 10, 2019, date of current version April 17, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2909398

Deep Learning-Based Spectrum Prediction Collision Avoidance for Hybrid Wireless Environments

RUBEN MENNES^{ID1}, (Student Member, IEEE), MAXIM CLAEYS¹, (Member, IEEE),

FELIPE A. P. DE FIGUEIREDO², IRFAN JABANDŽIĆ², (Student Member, IEEE),

INGRID MOERMAN^{ID2}, (Member, IEEE), AND STEVEN LATRÉ¹, (Member, IEEE)

¹Department of Mathematics and Computer Science, University of Antwerp - imec, 2000 Antwerpen, Belgium

²Department of Information Technology, Ghent University - imec, 9000 Ghent, Belgium

Corresponding author: Ruben Mennes (ruben.mennes@uantwerpen.be)

This work was supported in part by the Research Foundation Flanders (FWO), in part by the Flemish Government, in part by the European Union's Horizon 2020 Research and Innovation Program under Grant 732174 (ORCA Project), and in part by the SMILE-IT Project, funded by the IWT.

ABSTRACT With a growing number of connected devices relying on the Industrial, Scientific, and Medical radio bands for communication, spectrum scarcity is one of the most important challenges currently and in the future. The existing collision avoidance techniques either apply a random back-off when spectrum collision is detected or assume that the knowledge about other nodes' spectrum occupation is known. While these solutions have shown to perform reasonably well in intra-Radio Access Technology environments, they can fail if they are deployed in dense multi-technology environments as they are unable to address the inter-Radio Access Technology interference. In this paper, we present Spectrum Prediction Collision Avoidance (SPCA): an algorithm that can predict the behavior of other surrounding networks, by using supervised deep learning; and adapt its behavior to increase the overall throughput of both its own Multiple Frequencies Time Division Multiple Access network as well as that of the other surrounding networks. We use Convolutional Neural Network (CNN) that predicts the spectrum usage of the other neighboring networks. Through extensive simulations, we show that the SPCA is able to reduce the number of collisions from 50% to 11%, which is 4.5 times lower than the regular Multiple Frequencies Time Division Multiple Access (MF-TDMA) approach. In comparison with an Exponentially Weighted Moving Average (EWMA) scheduler, SPCA reduces the number of collisions from 29% to 11%, which is a factor 2.5 lower.

INDEX TERMS Collaborative wireless networks, deep learning, machine learning, wireless MAC.

I. INTRODUCTION

Spectrum scarcity is a phenomenon that is getting increasingly serious with the growing use of the unlicensed Industrial, Scientific and Medical (ISM) radio bands. Cisco expects that the number of mobile devices will grow from 8 billion devices in 2016 to 11.5 billion devices in 2021. In this time frame, data traffic to and from these devices is expected to increase from 7.2 to 49.0 exabytes per month. Half of this data is exchanged via Wi-Fi [1]. At the same time, the number of technologies that use these ISM bands is also growing. This indicates that cross-technology interference (e.g. Wi-Fi conflicting with Bluetooth) is a major and

The associate editor coordinating the review of this manuscript and approving it for publication was Yin Zhang.

growing issue. Similarly, also in the unlicensed band, new Long Term Evolution (LTE) versions (e.g., Long Term Evolution-Unlicensed (LTE-U)) are making use of the spectrum in these bands as well.

Most network technologies in the ISM radio bands optimize their own network performance without considering that access to the spectrum is shared with other technologies. Several Medium Access Control (MAC) protocols with different collision avoidance techniques have been proposed and are being used in different wireless technologies. We can divide the collision avoidance techniques in three different classes [2]. (i) Random Access protocols: Nodes can access the spectrum at any time to transmit packets. Most of the time these protocols use a Carrier Sensing Multiple Access (CSMA) collision avoiding technique to reduce collisions.

(ii) Time-slotted protocols: The spectrum is divided in fixed slots, and nodes can only use the spectrum during slots defined by a scheduling algorithm. (iii) Hybrid protocols: Some protocols use a partially slotted transmission. Control signaling can make use of synchronized time slots, while data transmission may use random access protocols without time synchronization. All these techniques focus on optimizing their own performance. To the best of our knowledge, there is no protocol that is able to react on cross-technology behavior to avoid cross-technology interference.

As existing MAC protocols, described above, are simple inter-Radio Access Technology (RAT) collision avoiding techniques or do no inter-RAT collision avoiding at all, these algorithms are not able to perform well in multi-technology environments. The problem of optimizing inter-RAT spectrum is complex because of the diversity of technologies, the wide variety of different applications, and the mobility of the nodes. This makes the environment very dynamic and the problem hard to solve with naive approaches. Complex patterns could still be detected and used to avoid collisions and optimize throughput, but need a more advanced solution.

To find a technique to optimize the spectrum usage, and increase the overall throughput, we need a strategy combining the advantages of a Multiple Frequencies Time Division Multiple Access (MF-TDMA) network and a CSMA network. The former achieves good coordination within the network while the latter senses the spectrum to avoid collisions even with unknown sources. In this paper, we present Spectrum Prediction Collision Avoidance (SPCA), an approach to predict the behavior of the surrounding networks and environment by using deep learning. Based on this prediction, an MF-TDMA scheduler can select slots that will avoid collisions in a smart way and optimize its own traffic and the traffic of other networks. This will lead to fewer collisions on all networks, increase the overall throughput, and optimize the usage of the scarce spectrum. The key component of our approach is an intelligent spectrum sharing mechanism that first predicts the behavior of the surrounding spectrum in the near future and then uses that prediction to avoid collisions. By using a deep learning approach, it is possible to detect complex patterns and use these patterns to optimize the spectrum usage. The deep learning approach makes the algorithm technology agnostic, which makes it an ideal approach to work in multi-technology environments where there is no a priori knowledge about the used technologies in the neighborhood. With SPCA it is also possible to tune the aggressiveness and assertiveness of the protocol. This gives the opportunity to configure the network differently for specific use cases or users.

By using a deep learning module, we are able to train our algorithm on real data captured from different unknown sources with different network technologies. By creating this dataset, we are able to train our approach offline and evaluate the algorithm in a more realistic simulated environment. This paper focuses on the prediction of the environment and other network technologies by using a pre-trained

Convolutional Neural Network (CNN). The CNN learns to produce a probability matrix describing the probability that a slot (channel-time pair) in the next frame will be used. This probability matrix can then be used in a centralized or distributed MF-TDMA scheduler. There are three main contributions in this work. 1) The architecture of the approach, where we are using a CNN to predict the upcoming spectrum usage. 2) We used supervised learning to train our proposed models in a supervised offline way. 3) We are able to train our proposed models on real case data captured during the Defense Advanced Research Projects Agency (DARPA) Spectrum Collaboration Challenge (SC2) [3]. All the captured data is generated in the spectrum by other teams in the competition. Other teams are using different kind of radio technologies and have their own strategy and footprint on the spectrum which is unknown for our network. While the objective of the manuscript is to reduce the number of collisions when sharing the same portion of the spectrum and therefore, to increase the overall throughputs of all the networks.

The remainder of this paper is structured as follows: section II discusses related works on collision avoidance techniques, MF-TDMA networks, Cognitive Radio (CR) and Machine Learning (ML) for wireless network management. In section III, we give a formal description of the problem, while in section IV the architecture and operation of SPCA is described. In section V, we show how the simulations are executed. The results of these simulations are presented in section VI. Finally, section VII concludes the paper and summarizes the key findings and future work.

II. RELATED WORK

In this section, the state of the art with regard to collision avoidance techniques, MF-TDMA networks, and ML techniques for wireless network management is discussed. Table 1 summarizes the most important related work described in this section. The work presented in this paper builds on top of the proof-of-concept *Neural-Network-based MF-TDMA MAC Scheduler for Collaborative Wireless Networks* presented in our previous work [4]. While our previous work was focused on using ML to optimize the own network, this work optimizes on collaboration of our system to use the spectrum in a smarter way, providing higher throughput while providing more opportunities to other technologies to enhance their own performance. Additionally, this work is evaluated with realistic neighboring networks. While, in [4] only a jammer with programmed jamming distributions was used.

A. COLLISION AVOIDING TECHNIQUES

As described previously, there are three main groups to describe collision avoiding techniques. (i) Random Access protocols: One of the major known and used collision avoidance techniques is CSMA. CSMA techniques try to verify the absence of other traffic before using the spectrum. Most of the time this is done by starting to listen to the medium upfront. Only if the medium is free from the transmitter

TABLE 1. Collision avoiding algorithms summary.

Algorithm	Inter-RAT avoiding	Optimize	Multi-Channel	Learning method
CSMA with Listen-Before-Talk (LBT)	Yes	Own link	No	None
MF-TDMA	No	Own network	Yes	None
ALOHA-QIR, Chu et al. [5]	Yes	Own link	No	Q-Learning (online) [6]
Win-Stay Lose-Shift, Phung et al. [7]	Yes	Own link	Yes	None
CR Q-learning, Syed et al. [8]	Yes	Avoiding Primary User (PU)	Yes	Q-learning (online) [6]
Deep RL, Wang et al. [9]	Yes	Optimize node	Yes	Deep RL (online)
SPCA, proposed algorithm	Yes	Optimize overall throughput	Yes	Deep supervised learning (offline)

point of view, transmission starts. This mechanism is known as Listen-Before-Talk (LBT). It will send the packet at a random time, starting from the moment it does not notice any activity on the medium. This can be extended with a RTS/CTS (Request To Send/Clear To Send) flow control mechanism. This method tries to avoid collisions by creating a coordinated access to the medium using control signals between nodes. RTS/CTS communication cannot be used over different networks or across network technologies, while carrier sensing strategies will also listen to other network technologies. It is known that some CSMA networks (such as Wi-Fi) can block in a listening state if there are other network technologies using the same spectrum [10]. (ii) Time-slotted protocols: Another way of using the spectrum is by defining fixed time slots. This is also called Time Division Multiple Access (TDMA), or MF-TDMA if we also split the available bandwidth into different channels. These MAC protocols need work-wide synchronization so that every node can start listening and receiving exactly at the correct time, as defined in the MF-TDMA schedule. Over the last years, we have seen more and more technologies using TDMA schemes, especially in sensor networks [11] and Vehicular Ad hoc Networks (VANETs) (such as STDMA, SOFTMAC, TC-MAC, ect) [12]. Also, MF-TDMA MAC protocols are used more frequently in newer technologies such as 6TiSCH (6 Time Synchronized Channel Hopping) [13]. (iii) Hybrid protocols: Some protocols use a partially slotted transmission. Control signaling can make use of synchronized time slots, while data transmission may use random access protocols without time synchronization. The objective of all techniques described above, is optimizing the own network. In this work, we extend and use these principles to optimize the overall throughput of all networks. In a lot of modern situations were different network technologies are deployed at the same location and used at all the same time, it is important to use the spectrum as efficient as possible and avoid collisions between different neighboring networks.

B. MF-TDMA NETWORKS

MF-TDMA networks and scheduling in Wireless Sensor Networks (WSNs) are well-studied topics in literature. We can divide the TDMA and MF-TDMA schedulers into two big groups: centralized and decentralized schedulers.

Centralized schedulers are deployed on a centralized (master) node in the wireless network. All other nodes use control frames to send their knowledge about topology,

routing, queue lengths and application-specific parameters to the unique master node. This master node runs the scheduling algorithm and broadcasts the schedule to all the nodes in the network [14]. It is clear that this approach will result in an in-network collision free schedule as nodes cannot decide on their own when to send, and the centralized scheduler is “omniscient”. The major downside of this approach is the overhead in control communication to setup and maintain the MF-TDMA schedule and the latency.

Decentralized algorithms, on the other hand, are more flexible. There is a lower communication overhead at the cost of potential collisions scheduled inside the network itself. One of the first proposals for a decentralized 6TiSCH scheduler in WSNs was made by Tinka et al. [15]. Even though the work of Tinka et al. was never implemented, it was the inspiration of a variety for other decentralized MF-TDMA schedulers. Some schedulers use a Resource Reservation Protocol (RSVP)-like approach [16]. In this flow-based approach, each flow will reserve messages in the MF-TDMA schedule on the route of the flow. This will create a chain of reserve messages from source to destination before transmission can start. This approach is powerful if the main data is flow-based, but produces a lot of overhead if the main data is send in bursts or if routes change often. The DeBras algorithm [17] can handle bursty data more efficiently. In this algorithm, a node decides on its own to allocate a slot to one of its neighbors. Routes are allocated dynamically. This means that every node needs to have additional knowledge about routing. A node can use the On-the-Fly bandwidth reservation algorithm [18] to define how many slots it needs to transfer its traffic dynamically based on the current queue length.

In this paper, we do not focus on the scheduler. Our approach will provide additional data that can be used in any kind of scheduler so it will have more information about the environment and other networks in the neighborhood of the nodes. In the remainder of the paper, we use a centralized scheduler in our own network because this guarantee us that there will be no collisions scheduled inside the own network. This lead to a cleaner observation and evaluation.

C. COGNITIVE RADIO NETWORKS

Another approach to solving the spectrum scarcity problem is to allow users to use other frequency bands as long as they are free. This approach is called Cognitive Radio (CR) networks [19]. In the known CR network, there typically exist two types of users. (i) The Primary Users (PU) are the actual

users of the frequency band, while (ii) the Secondary Users (SUs) are allowed to use the frequency band only if they will not interact with the PUs. In this approach, all the SUs need to be able to know if a PU is using the channel and if they will introduce collisions should they start sending.

While some different techniques, such as Spectrum Decision, Spectrum Sharing and Spectrum Mobility, have been proposed, most algorithms start with Spectrum Sensing. Generally, Spectrum Sensing techniques can be classified into three groups: primary transmitter detection, primary receiver detection, and interference temperature management. The basic idea of all these spectrum sensing techniques is that a SU senses the spectrum and decides if the spectrum was free and could be used for its own traffic.

The main difference between the CR and our approach is the fact that there is no PU, nor SU in our approach. Every network can be equally important. Instead of using unused spectrum, as is the goal of CR, the goal of this paper is to optimize the already used spectrum.

D. MACHINE LEARNING FOR WIRELESS NETWORK MANAGEMENT

ML is a term used to describe several algorithms to optimize decision-making tasks. In the last years, this field made a huge progress in the domain of computer vision [20], natural language processing [21] and strategies tasks [22]. By extending this principle, researchers were able to play vintage Atari games better than a human where an artificial agent learns the game by using deep Reinforcement Learning (RL) [23]. In the context of collision avoiding ML strategies, a few algorithms are already proposed and make use of RL. In this context, the nodes act in the environment and receive a reward or penalty based on the outcome of the action/transaction. This reward is used to update values that represent a policy. Chu et al. [5] proposed ALOHA-QIR, an algorithm that uses Q-Learning to improve Slotted ALOHA where they achieve twice the maximum throughput of Slotted ALOHA while being more energy efficient. In the ALOHA-QIR algorithm, every node creates a frame that contains N slots. Each slot in the grid represents a score in the Q-Table. Based on the Q-Learning algorithm, the reward used in the Q-Learning algorithm is based on whether the transmission was successful or resulted in a collision. The advantage of this approach is the low complexity of both ALOHA and the Q-Learning algorithm. This combination makes the ALOHA-QIR light-weight and a fitting protocol for WSNs. However, one of the disadvantages of this approach is scalability. As the number of nodes and the amount of data grows, the ALOHA-QIR algorithm does not scale because it can only use a single channel. Phung et al. [7] proposed another strategy for an MF-TDMA network. In this approach, each node stay in the configuration that is active at the moment, as long communication is successful. However, once a packet was not successfully delivered, the node will pick a random slot until again reaching a stable situation. It turns out that a network can find a stable situation, but it

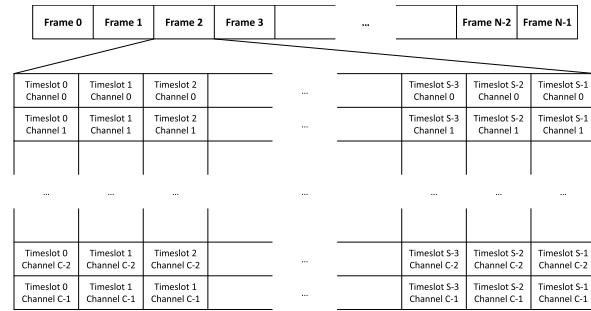


FIGURE 1. MF-TDMA frame of the own network.

is clear that if another network is also using the spectrum simultaneously, trying to find a stable configuration by randomly selecting slots, is not very efficient and could lead to a cascade of failures. The authors in [8] used an Universal Software Radio Peripheral (USRP) and GNU radio units to implement Q-learning in a multi-hop cognitive radio network. This approach was successful and have an implementation in CRs. Unfortunately, this approach used RL, which makes the algorithm slower to adapt during the bootstrapping phase, additionally the work focused on completely avoiding PUs, while in this work we focus on collaborative spectrum sharing where we want to optimize the overall network throughput. Wang et al. proposed to use Deep RL to select a free channel [9]. The difference with our approach is that we predict the spectrum on every node, and this could scale easily which is not the case in the approach of Wang et al. because of the multi-agent nature of the problem. Because the number of actions in an MF-TDMA network can be huge and as it is hard to evaluate actions in wireless networks, we choose to use function approximation with Neural Networks (NNs) instead of an RL technique. This allowed pre-training a NN, in our case a deep CNN, on smaller devices that are not able to keep training the model itself. The system avoid the learning phase that is typical for a RL framework. Using ML techniques only for prediction allows us to train the NN on measurable data.

III. FORMAL PROBLEM STATEMENT

In this section, we state the problem in a formal and mathematical way. Used symbols are described in Table 2 and defined later in this section.

In this work, we assume that the network consists of N MF-TDMA nodes called the own network, and a second (unknown) set of networks later called the Interfering Networks Cluster (INC), which may also include noise from other sources (e.g. microwaves). As we focus on the unlicensed bands, we assume that both networks will have equal priority and rights to use the spectrum. Because the INC is unknown, the own network has no knowledge about used technologies, traffic type and so on.

All N nodes in the own MF-TDMA network can communicate with each other on C channels. Each channel is divided into time slots. For each frame in the MF-TDMA schedule, there are S time slots and C channels available, as illustrated in Figure 1. We assume that all nodes in the own

TABLE 2. Overview of used symbols.

Symbol	Description
C	Number of channels used in the MF-TDMA super frame. $C \in \mathbb{N}$
S	Number of timeslots used in the MF-TDMA super frame $S \in \mathbb{N}$
N	Set of nodes in the own network.
F	Set of superframes in the overall execution.
c	$c \in [1, C]$.
s	$s \in [1, S]$
n	Node in the own network. $n \in N$
f	A super frame in the execution. $f \in F$
$A_{s,c}^{f,n}$	Action of node n at slot s, c in superframe f .
$TX_{s,c}^{f,n}$	1 if $A_{s,c}^{f,n} = TX$. 0 otherwise.
$RX_{s,c}^{f,n}$	1 if $A_{s,c}^{f,n} = RX$. 0 otherwise.
$P_{s,c}^f$	1 if the INC used channel c at slot s of superframe f .
$\gamma_{s,c}^f$	1 if $\sum_{n \in N} TX_{s,c}^{f,n} = 1 \wedge P_{s,c}^f = 0$. 0 otherwise.
$\gamma_{s,c}^{ff}$	1 if $\sum_{n \in N} TX_{s,c}^{f,n} = 0 \wedge P_{s,c}^f = 1$. 0 otherwise.
$\omega_{s,c}^f$	1 if there was a collision, 0 otherwise.
$\hat{\Gamma}^f$	The maximum overall throughput of the own network at superframe f .
$\hat{\Gamma}^f$	The maximum overall throughput of the INC at superframe f .
Ω^f	The total amount of collisions in superframe f .
$O_{s,c}^{f,n}$	The observation of energy at slot s, c of superframe f at node n .
$F(\alpha)$	Overall throughput of both the own and the INC relative to α .
$p_{s,c}^{f,n}$	Predicted probability energy will be detected by node n on slot s from frame f at channel c .
Δ	Relative distance between INC sender and receiver in simulation topology defined in Section V.
t	Threshold that defines the minimal certainty of a predicted slot $p_{s,c}^{f,n}$ before it can be scheduled by the scheduler.

network will be synchronized and have the same notion of time. This is a general property of every MF-TDMA network. Based on this property, we can state that all nodes in the own network execute the same slot s at exactly the same time for all frames $f \in F$. Every node n will execute an action $A_{s,c}^{f,n} \in \{\text{Idle}, TX, RX\}$ at time (f, s) for every channel c . Furthermore, we denote $TX_{s,c}^{f,n}$ and $RX_{s,c}^{f,n}$ as:

$$TX_{s,c}^{f,n} = \begin{cases} 1 & \text{if } A_{s,c}^{f,n} = TX \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$RX_{s,c}^{f,n} = \begin{cases} 1 & \text{if } A_{s,c}^{f,n} = RX \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

As a constraint, a node can only execute at most one action $\neq \text{Idle}$ at each time:

$$\forall n \in N, \quad \forall (f, s) \in F \times [1, S] : \sum_{c \in [1, C]} TX_{s,c}^{f,n} + RX_{s,c}^{f,n} \leq 1 \quad (3)$$

In this model we assume there will be an unknown number of nodes in the INC. We will only know if one of the nodes used the channel at a given time in the past. We assume that $P_{s,c}^f = 1$ if and only if the INC used channel c at time (f, s) . Note that for the INC there is no constraint on the number of transmissions for a single timestamp. This is

because the number of interfering networks inside the INC and the number of nodes they each contain, along with any characteristics of applied protocols are all unknown.

As described before, the goal of this work is not to design the scheduling algorithm itself but rather to improve the available information as a means to create a better and smarter schedule. Because of this, we use a simple centralized scheduler based on the work of Palattella et al. [24]. We assume that before every frame, all N nodes of the own network have exchanged all predicted information about the next frame with the master node. Based on this information, it constructs a schedule for the next frame. The next frame will only start once all the nodes received the new schedule.

Communication between two different nodes is successful if and only if at a given time (f, s) a node n is in an RX state and only received a message from exactly one other node on a given channel c . Because we use a centralized scheduler, we assume that the own network cannot schedule two TX actions for two different nodes at the same time using the same channel. We also assume that the INC is powerful enough that its transmissions can collide with transmissions of our own node. Based on these assumptions, we can define that communication of the own network at time (f, s) on channel c is successful if and only if one own node is transmitting at the time in that channel and no interfering node is transmitting. We use $\gamma_{s,c}^f$ to determine if an own packet could be successfully delivered or not. Note that there is always a chance that communication will fail because of the environment (e.g. reflection, fading, etc.).

$$\gamma_{s,c}^f = \begin{cases} 1 & \text{if } \sum_{n \in N} TX_{s,c}^{f,n} = 1 \wedge P_{s,c}^f = 0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Because there is no way for our own nodes to know if a transmission interferes with a packet of the INC, we can assume that communication in the INC is successful if $\gamma_{s,c}^{ff} = 1$, where:

$$\gamma_{s,c}^{ff} = \begin{cases} 1 & \text{if } \sum_{n \in N} TX_{s,c}^{f,n} = 0 \wedge P_{s,c}^f = 1 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

We define $\omega_{s,c}$ to be 1 if there is a collision. Note that this $\omega_{s,c}$ will denote collisions between the own and the INC, because we assume that the own network will be collision-free.

$$\omega_{s,c}^f = \begin{cases} 0 & \text{if } \sum_{n \in N} TX_{s,c}^{f,n} + P_{s,c}^f \leq 1 \\ 1 & \text{otherwise} \end{cases} \quad (6)$$

The maximum overall throughput of our own network for each frame f , $\hat{\Gamma}^f$, depends on the number of RX/TX slots for each node and the number of collisions among the network. In that way, we can define the throughput $\hat{\Gamma}^f$ as the maximal number of successful packets in frame f for the own network:

$$\hat{\Gamma}^f = \sum_{s \in [0, S]} \sum_{c \in [0, C]} \gamma_{s,c}^f \quad (7)$$

In the same way, we define the maximal throughput $\hat{\Gamma}^f$ for each frame f in the INC:

$$\hat{\Gamma}^f = \sum_{s \in [0, S)} \sum_{c \in [0, C)} \gamma_{s,c}^f \quad (8)$$

We define Γ^f and $\hat{\Gamma}^f$ as the actual relative throughput for the own network and INC respectively. This means that Γ^f is the number of arrived packets divided by the number of used slots in frame f .

The goal of this algorithm is to maximize the throughput of all networks by choosing the actions A as optimally as possible for all channel-time pairs for all the nodes n in the own network. We will define the objective function as follows:

$$\text{maximize } F^f(\alpha) = \alpha \Gamma^f + (1 - \alpha) \hat{\Gamma}^f \text{ with } \alpha \in [0, 1] \quad (9)$$

Note that to increase throughput Γ^f and $\hat{\Gamma}^f$, we can either (i) increase the number of TX/RX cells, (ii) generate more data, or (iii) decrease the number of collisions. Because we can only control our own network and we assumed that we have no control over the data generation, we can only increase the throughput for all networks by decreasing the number of collisions. This means that we can maximize both Γ^f and $\hat{\Gamma}^f$ by reducing the number of collisions Ω^f where:

$$\Omega^f = \sum_{s \in [0, S)} \sum_{c \in [0, C)} \omega_{s,c}^f \quad (10)$$

We assume that every node n can measure energy on all C channels. This is a fair assumption as new radio hardware technologies (such as Software Defined Radios (SDRs)) support simultaneous energy measurement over a broad bandwidth. We define $O_{s,c}^{f,n} \in [0, 1]$ as the observation made at time (f, s) on channel c by node n , where $O_{s,c}^{f,n} = 1$ if and only if all the bandwidth of the slot was used during the entire duration of the slot by the INC and $O_{s,c}^{f,n} = 0$ if and only if the slot is not used at all.

IV. SPECTRUM PREDICTION COLLISION AVOIDANCE

A. ARCHITECTURE

As illustrated in Figure 2, the proposed approach consists of 5 main components: the spectrum monitor, prepossessing unit, predictor unit, probability matrix on each node and the overall (centralized or decentralized) scheduler. In general, each node captures the energy on the overall spectrum. These energy measurements are forwarded to the preprocessing unit on the node. The preprocessing unit of node n will create the correct observation values $O_{s,c}^{f,n}$ for each slot. Once the prepossessing step is done for the entire frame, we can use a predictor unit. The predictor unit produces a probability matrix that describes the predicted usage of a slot in the next upcoming frame. The probability matrices from all nodes are used in the scheduler to select slots that are expected to be free during the next frame(s). Note that the scheduler could be centralized or decentralized. As long the scheduler itself uses the probability matrix to choose 'free' slots. In this paper,

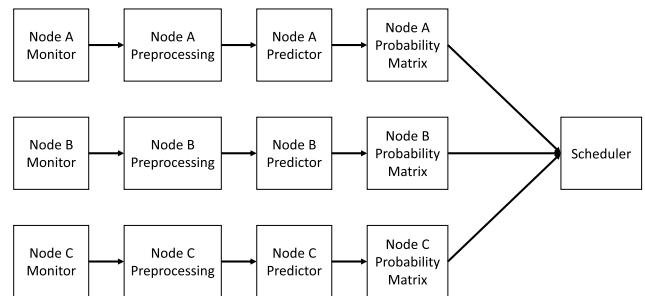


FIGURE 2. Overall algorithm architecture for the own network of three nodes.

we will focus on the predictor unit and use a centralized scheduler. In this way, we can ignore collisions because of bad scheduling due to the decentralization. By using a decentralized scheduler, this architecture scale easily, as there is no additional message exchange.

In Algorithm 1 the overall working of the SPCA algorithm is expressed. If a reschedule of the slots need to be executed each node will execute the function *Reschedule* as illustrated in Algorithm 1. Four steps will be executed. 1) Each node will construct an observation based on monitored history of h superframes. 2) A preprocessor will make sure that the observation is used by the predictor unit. In practice this implies normalizing the data and reshaping it in the correct way. 3) Pass the preprocessed data to the predictor unit (used in the current setup) at each node. Based on the internal working of the predictor unit (could be CNN, Neural Weighted Moving Average (NWMA) or something else) a prediction is computed. 4) These prediction are forwarded to the scheduler (centralized or decentralized) so it will select slots where the receiving node predicted that the slot will be free.

B. PREDICTOR UNITS

As mentioned above, a key component of our approach is the predictor unit. This unit predicts how the spectrum will be in the upcoming frame(s) by using historic data of the spectrum and a (trained) model. The output of the predictor unit is a frame with probabilities of a slot being used by another network. With this approach, it is possible to build a model which can detect concepts such as RTS/CTS in Wi-Fi by training the model on examples of this behavior. Without having any notion of the RTS/CTS concept, the model will learn that after two short packets there is a high probability that there will be a large packet on that channel in the near future. We define $p_{s,c}^{f,n} \in [0, 1]$ as the outcome of a predictor for frame f on slot s in channel c at node n .

In this paper, we introduce two predictor units. (i) The CNN predictor unit, and (ii) a shallow NN called NWMA. We discuss both approaches in the remainder of this section.

The CNNs predictor unit can be used to find small patterns across different locations in an image in different layers of the network. In our case it finds patterns (e.g. RTS/CTS) in different convolutional layers. It will search for these patterns

Algorithm 1 Pseudo Code Of the SPCA Algorithm

```

function Reschedule
   $O \leftarrow \text{getObservation}(h)$                                  $\triangleright$  Executed on each node
   $O \leftarrow \text{doPreprocessing}(O)$                              $\triangleright$  Get the observation of the last  $h$  super frames
   $P \leftarrow \text{Predict}(O)$                                       $\triangleright$  Do preprocessing on observation (if necessary)
  return ForwardPredictionToScheduler( $P$ )                       $\triangleright$  Execute the predictor unit used at the node based on the observation
end function                                                  $\triangleright$  Send probability matrix to scheduler and return new schedule

```

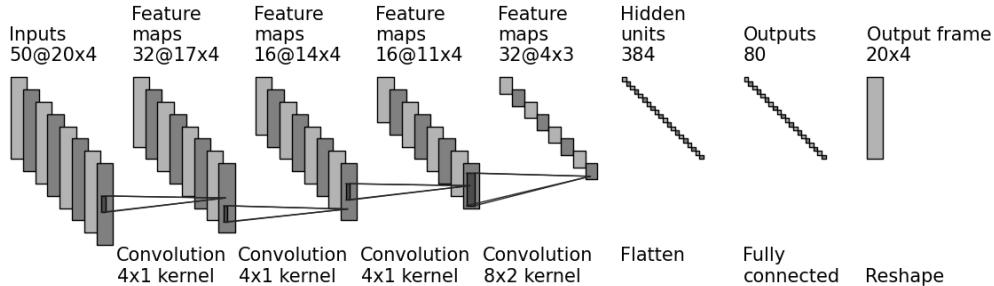


FIGURE 3. CNN architecture with 5 hidden layers: 4 convolutional layers followed by a fully connected layer. Using spectrum observation as input and outputs a probability matrix.

across the complete spectrum diagram, which can be represented as an image. Because each observation frame can be interpreted as a gray-scale image over time, CNNs are a more appropriate choice than traditional fully connected Deep Neural Network (DNN). In the CNN approach, we create a CNN with five hidden layers as illustrated in Figure 3. The CNN network consists of four convolution layers with kernel sizes of [4, 1], [4, 1], [4, 1] and [8, 2] respectively, followed by a fully connected layer. We use a ReLu activation function for all convolution layers. After the last fully connected layer a sigmoid activation function is applied. The input of the CNN is a 3D tensor that contains h observed superframes. The third dimension are the h different frames and will be used as filters in the first hidden convolution layer.

For the second predictor unit we used an NN with one hidden layer which we call NWMA. In this approach, we take the average over the last h frames, but the weights of each layer are learned during an off-line training phase. This approach is easier to deploy on constrained devices, as the number of weights to store is almost 1000 times smaller. In the simulations, NWMA contained only 50 weights, while the CNN we used had 48944 weights to store. But also the amount of operations is a factor 1 million smaller for the NWMA, making it a lot easier to use on a constrained embedded device.

The two introduced predictor units are interchangeable and can be used in the described architecture illustrated in Figure 2. Both predictor units use the preprocessed monitor information (O^n) of h history frames as an input and will return a probability matrix (p^n) which will be used by the scheduler.

We used a training set to train both the NWMA and the CNN network. For both networks, we used an Adam Optimization function [25] with a Learning Rate (LR) of

initially 10^{-3} together with a linear learning rate decay, reducing the LR to 10^{-7} after 2000 steps. The mean squared error is used as a cost function. We trained our NNs, both the CNN and NWMA, for 5000 episodes. No online training was executed during the simulations. Note that this will implies that the evaluation and training distribution of the NN should be similar. As a consequence the system cannot react on unknown behavior of the INC networks. Similar behavior (in the sense of similar state distributions) should be used for training as for execution of the system.

V. SIMULATION SETUP

To evaluate the predictor units, we used an MF-TDMA discrete event simulator written in Python based on the 6 Time Synchronized Channel Hopping (6TiSCH) simulator of Palatella et al. [18]. We changed the simulator to make it possible to have two separate networks where one of them does not use the MF-TDMA schedule, but only informs the simulator about its spectrum usage. In that way, we could implement both the INC and the own network into the simulator. The used topology is illustrated in Figure 4, where the gray nodes are the interfering nodes. The node in the middle sends packets to the node outside the ring.

To evaluate our approach and to train some of the predictor units, we captured and stored monitoring data during the first Preliminary Event (PE) of the SC2 competition organized by the DARPA [3]. In this competition, 19 different teams play different games against each other with the goal of optimizing both their own and the global throughput. Different games had different scenarios and topologies, which were unknown during the game. It is important to note that all teams created their own network stack from scratch and used different approaches. This makes the environment of the game highly realistic.

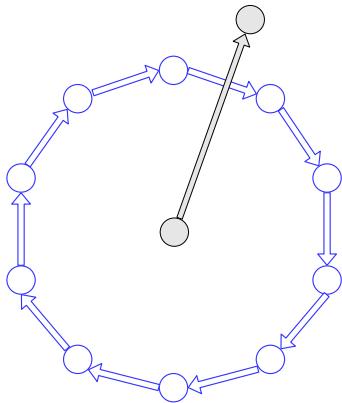


FIGURE 4. Topology and routing of the simulated test network with one unknown network (gray) and an surrounding MF-TDMA ring network (blue).

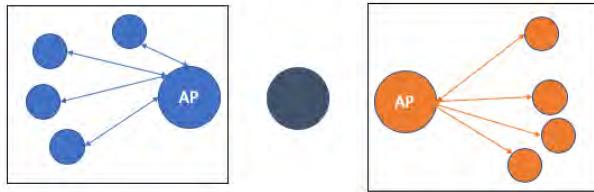


FIGURE 5. Topology of the simulated AP topology with two infrastructure networks and one “jamming” INC in the middle.

We divided all the captured games of all different scenarios in a training set (90%) and an evaluation set (10%). We reused the Pister-hack propagation model of the simulator to decide whether or not a packet of the INC will arrive correctly in the evaluation phase.

In this work, two realistic topologies are used as a simulation setup. The first topology is used for the majority of the simulations. In this topology, ten of our own nodes are located in a ring centered around a single interfering node. Our own nodes generate traffic with a random probability and always send their traffic to the next node in the ring, as illustrated with the blue arrows in Figure 4. Each node in the own network captures the energy generated by the INC and runs a predictor unit. The interfering receiver is located outside of the circle, at a distance Δ (relative to the circle’s radius) away from the interfering sender. The second topology is used to illustrate that the algorithm can also be used in other kind of topologies. As illustrated in Figure 5, two networks are operating in infrastructure mode. Both networks have a centralized Access Point (AP). All nodes will only communicate with the AP. We suppose that both networks can transmit data on every channel at every timeslot. The networks itself are not interfering with each other but APs interfere with the INC.

In this work, In this paper, we used a centralized scheduler that can reschedule the next frame after the execution of the last slot of the last frame. The main advantage of such a centralized scheduler is the fact that we can avoid collisions in our own network. In that way we could use the collision as a metric to evaluate our approach. Unfortunately, this cannot

be guaranteed by using a distributed scheduler. The used centralized scheduler assigns slots and selects the best slot for each node in the own network in a round-robin way. This is a valid assumption because this can easily be done by a centralized scheduler. We introduce an additional threshold $t \in (0, 1]$, and only if the probability value of a predicted slot $p_{s,c}^{f,n}$ is greater than t a slot can be selected by the scheduler. For example: if $t = 0.5$, the MF-TDMA scheduler only select slots if the chance that a slot will be free is larger than 50%.

The used prediction unit can easily be replaced by different methods, which we describe in subsection IV-B. This is possible because a predictor unit is actually a function that maps an observation matrix $O^{f,n}$ to a probability matrix $p^{f,n}$.

VI. RESULTS DESCRIPTION

A. METRICS

For each combination of parameters, we ran the simulation 25 times with different random seeds and data generator. We calculate the objective function $F^f(\alpha)$ of each frame f . Because we are especially interested in the overall network performance, we will redefine our objective function to $F(\alpha)$ as defined in Equation 11.

$$F(\alpha) = \frac{\sum_{f \in F} F^f(\alpha)}{|F|} \quad (11)$$

We can consider $F(\alpha)$ to be the weighted average throughput of all networks, where α defines the weight between our own throughput and the throughput of the others. If α is close to zero, we focus on the throughput of the interfering networks, while if α is close to one, we focus on the throughput of our own network.

Because we want to optimize the spectrum usage, we introduce two other metrics. (i) Collision Rate is the relative number of collisions between our own and the interfering networks. (ii) Missed Opportunities is the average number of free slots that could have been used for transmission by at least one node in our own MF-TDMA network. This means that there was a slot that was not used by the interfering network, and a transmit node and receiver node of the own MF-TDMA network that executed the Idle action while the transmitter had data available.

B. BASELINE PREDICTORS

To evaluate the predictor units described in subsection IV-B, we compared the simulations with two other state of the art techniques and with an approach where our own network does not send anything at all. This represents the base line of the interfering network, called as *Keep Silent* approach. The two other techniques are: (i) A regular MF-TDMA scheduler, called *Regular*. After every frame, a regular centralized scheduler can be applied. It will try to optimize the schedule of our own network without looking to the interfering network at all. (ii) An Exponentially Weighted Moving Average (EWMA) approach where the slot quality is measured by an EWMA over the most recent frames as shown in Equation 12. This is a typical extension to include slot quality to an

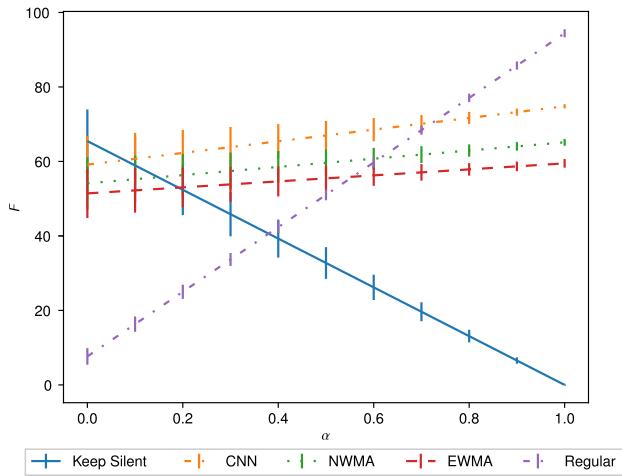


FIGURE 6. $F(\alpha)$ function for the simulations with all the different predictor units. For $\Delta = 1.55$ and $t = 0.5$.

IEEE 802.15 network [26]. An offline evaluation phase showed that, for the EWMA approach $a = 0.05$ gives the best results.

$$p_{s,c}^{f,n} = \frac{\sum_{i \in [1,h]} (1-a)^{i-1} O_{s,c}^{f-i,n}}{\sum_{i \in [1,h]} (1-a)^{i-1}} \quad (12)$$

C. GENERAL RESULTS

In Figure 6, we show the weighted average throughput function $F(\alpha)$ for the different predictor units. First of all, we see that with a *Keep Silent* strategy the overall throughput F will decrease if α grows as our own MF-TDMA network does not use any slot. On the other hand with a *Regular* strategy, the overall throughput F increased if α grows, although the regular scheduler produces more collisions with higher α , it also uses more free slots. It is clear that using any kind of quality measuring strategy (EWMA, NWMA, CNN) can increase our objective function from the moment we are interested in the throughput of more than one network. The results clearly show that if α is bigger than 0.1 and equal or smaller than 0.7, the CNN predictor unit performs even better than the most extreme approaches such, as not sending at all or using the complete spectrum in the regular approach. As it has the largest range where the weighted average throughput is the highest, the deep CNN is the best approach. The CNN approach is always outperforms the EWMA and the NWMA strategy, and only if we care about the throughput of only our network or only the interfering network, it is more appropriate to use a naive *Regular* or *Keep Silent* approach.

Figure 7 shows the relative number of collisions for all different strategies in comparison with the total number of used slots of the interfering network. The collision ratio of the CNN approach is the lowest of all (sending) approaches. We can reduce the number of collisions with 75% in comparison to the collisions in the *Regular* scenario which represents the current state of the art. In Figure 8 the number of missed

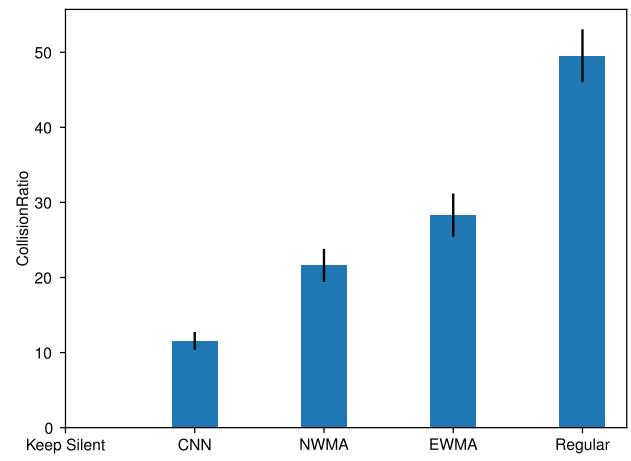


FIGURE 7. Average number of collision for the different predictor units. For $\Delta = 1.55$ and $t = 0.5$.

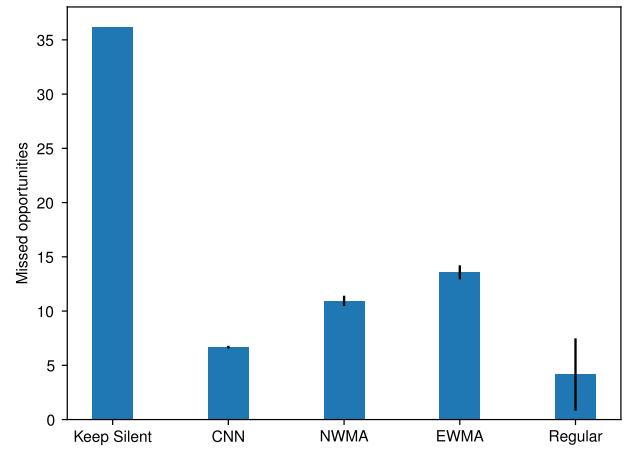
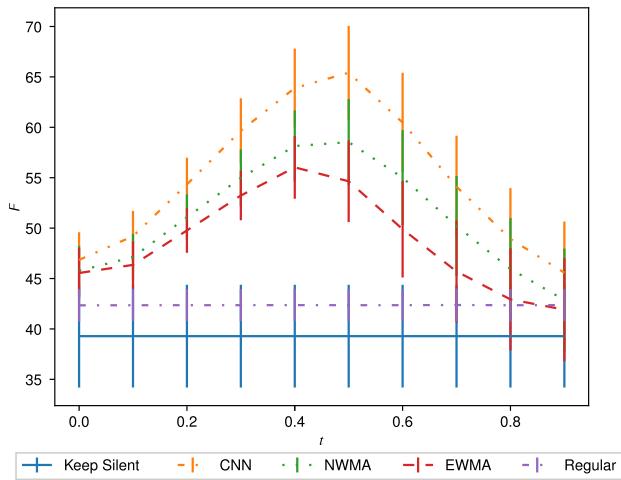


FIGURE 8. Missed opportunities for the different predictor units. For $\Delta = 1.55$ and $t = 0.5$.

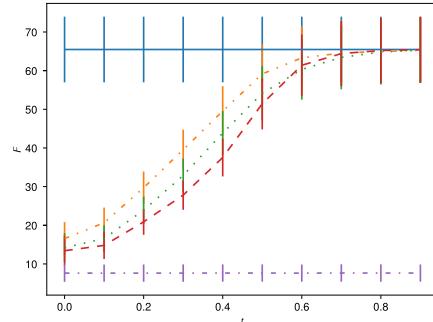
opportunities is visualized for all strategies. We see that the number of missed opportunities is very low for the CNN approach. Only the *Regular* approach does slightly better on average, which make is unsurprising as it will try to use as many slots as possible. For all strategies the nodes could save at most $h = 50$ frames in their memory.

D. IMPACT OF PARAMETERS

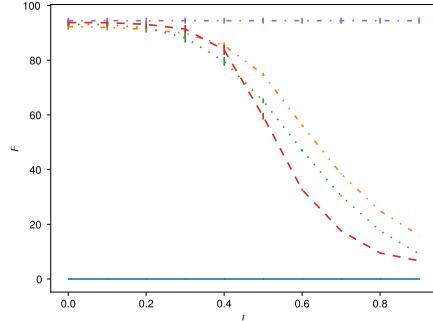
Figure 9 illustrates the $F(\alpha)$ function if we vary the threshold t from 0 to 1 in steps of 0.1. If we only want to optimize either the interfering or the own network it makes sense to keep t very low or high respectively, as shown in Figure. 9b and Figure. 9c. However, if we want to optimize the throughput of both networks we can see that a threshold value t around 0.5 will give us the most optimal solution. Here the scheduler is at least 50% certain a slot will be free and will leave the slots he is (almost) certain will be used by the interfering network untouched. This makes the own network less aggressive. It is important to note that it could be the case that not enough slots will be selected. It could be useful to



(a)



(b)

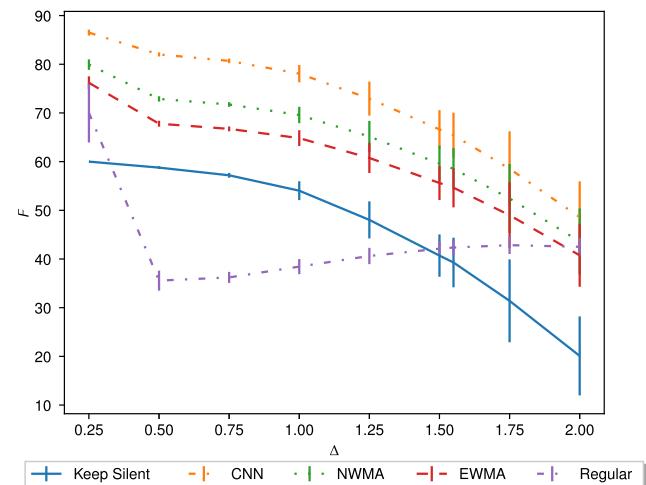


(c)

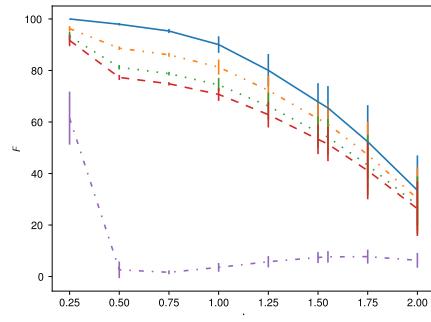
FIGURE 9. $F(\alpha)$ for the simulations with all the different predictor units over various thresholds t . With $\Delta = 1.55$. (a) $\alpha = 0.4$. (b) $\alpha = 0$. (c) $\alpha = 1$.

have a backup mechanism to overcome starvation of the own network.

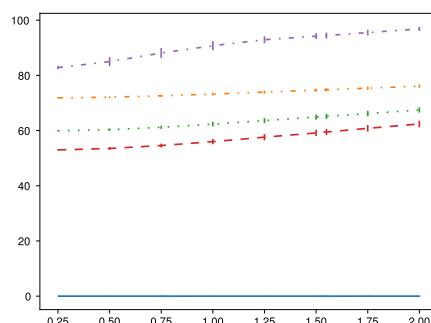
In Figure 10, we showed the impact of the relative distance of the interfering receiver Δ on the objective function $F(\alpha)$. If the interfering receiver is located further away there will be more influence of the nodes in the own network. Note that the influence of the interfering network will be the highest if the receiver is close to the own node, (i.e. with Δ between 0.5 and 1). If $\Delta < 0.5$, the interfering receiver is close to the sender, so it is easier to hear the sender. Otherwise, as Δ grows larger than 1, fewer nodes can be heard by the interfering receiver, but the interfering sender's transmissions can also



(a)



(b)



(c)

FIGURE 10. $F(\alpha)$ for the simulations with all the different predictor units over various interfering receiver distance Δ , with $t = 0.5$. (a) $\alpha = 0.4$. (b) $\alpha = 0$. (c) $\alpha = 1$.

fade out. We show in Figure 10a that if we want to optimize the overall throughput ($\alpha = 0.4$) our CNN approach will outperforms all the other approaches. We observe the expected behavior: the throughput (F) decreases if the distance in the interfering network (Δ) will increase.

If we increase the gain of the interfering network, the influence of the own network on the interfering network is minimal because the messages of the own network will always be “softer” and the interfering sender will over shout the own transmissions. This is visible in Figure 11, where we show the success ratio for $\alpha = 0$ and $\alpha = 0.4$ for different Δ .

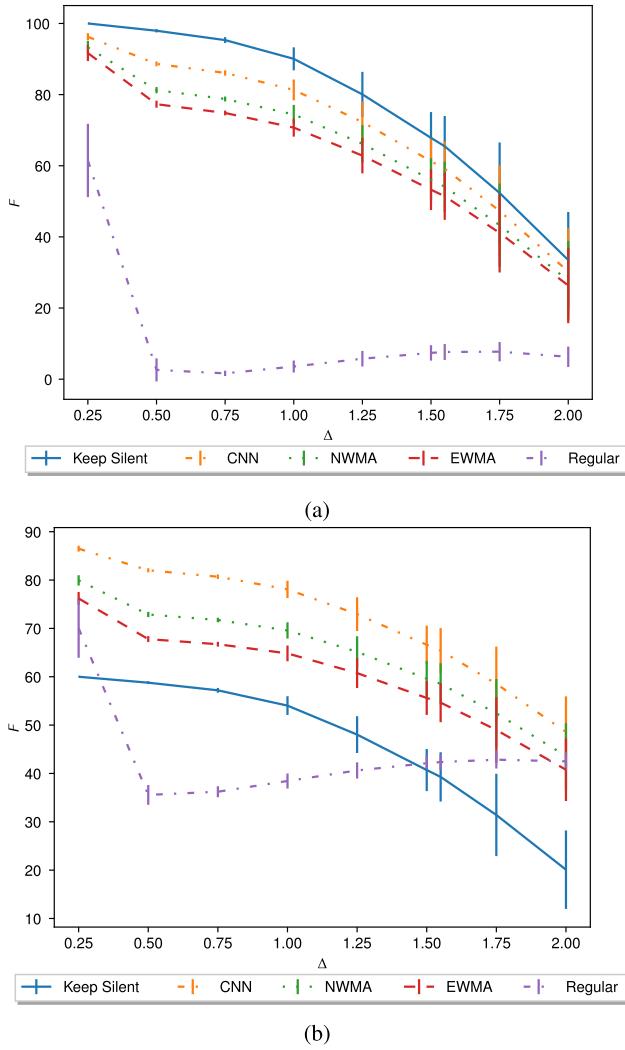


FIGURE 11. $F(\alpha)$ for the simulations with all the different predictor units over various interfering receiver distance d . With $t = 0.5$. Where interfering network use higher gain. (a) $\alpha = 0.0$. (b) $\alpha = 0.4$.

Figure 11a shows that the CNN approach is very close to the optimal if we want to optimize the throughput of the interfering network ($\alpha = 0$), even when our own network is also sending. On the other hand, if we want to optimize the overall throughput (with $\alpha = 0.4$) we can see that the CNN approach again outperforms the other approaches and can even provide 2.5 times more overall throughput than the constant sending approach, as is currently often deployed.

E. INFRASTRUCTURE MODE

As described in section V, we included a topology in infrastructure mode as well. Two different networks are interfered by a common INC as shown in Figure 5.

In Figure 12 we show the weighted average throughput function $F(\alpha)$ for the different predictor units in the AP simulation setup. At a first glance it is clear to see that the results of the AP simulation is similar to the previous described results in subsection VI-C. Because in this simulation setup it was

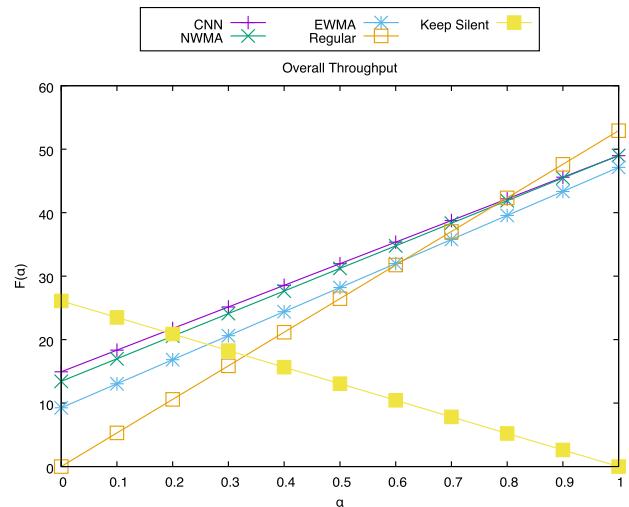


FIGURE 12. $F(\alpha)$ for the AP simulation with all the different predictor units.

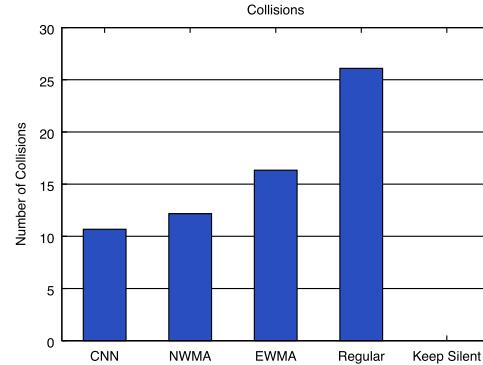


FIGURE 13. Average number of collisions for the different predictor units in the AP simulation setup.

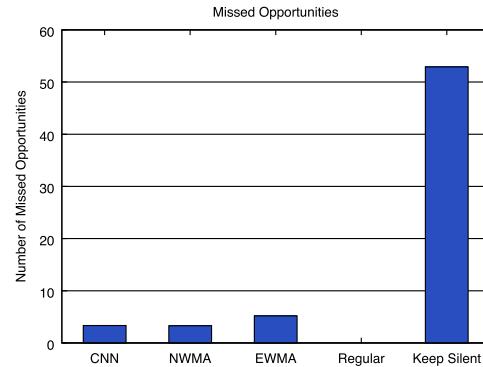


FIGURE 14. Average number of missed opportunities for the different predictor units in the AP simulation setup.

harder for the INC to transfer a packet successfully, we could see that the overall throughput with the CNN predictor unit and the NWMA predictor unit is almost equal. But, as shown in Figure 13 and Figure 15 we could see that less transmit opportunities are used which results in a higher success ratio and less collisions.

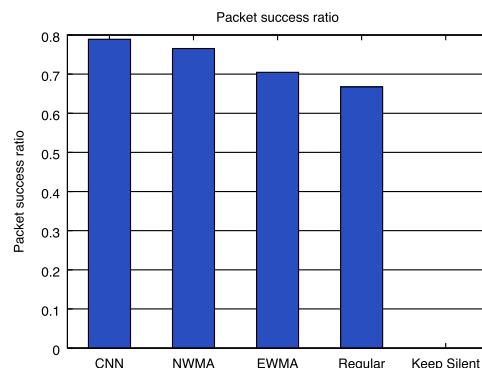


FIGURE 15. Average success ratio of the AP networks for the different predictor units.

VII. CONCLUSIONS AND FUTURE WORK

In this paper we presented SPCA, an approach to improve the cross-technology wireless spectrum usage by reducing the number of collisions between different networks where we can only control one MF-TDMA network. The goal is to optimize the throughput of all networks instead of only the own throughput. In the own MF-TDMA network, all nodes can measure the spectrum and use a predictor unit to predict the next frame based on captured history. We showed, by using a discrete event simulator, that a CNN predictor unit can be used to help the MF-TDMA scheduler to increase the combined throughput of our network and the INC. We also showed that by using the pre-trained CNN predictor unit it was possible to decrease the number of collisions with a factor 4.5, from 50% to 11%, in comparison to a regular MF-TDMA approach and decrease the number of collisions with a factor 2.6, from 29% to 11%, in comparison to a typical EWMA strategy. We showed that it is important to leave slots open by introducing the threshold parameter t . The scheduler will not use slots if the predictor unit predicts with high certainty that slots will be used in the surrounding networks.

Future research is necessary to fine-tune the predictor unit. In this paper we first observed the environment to train our model but in real case scenarios it would be useful to tweak or retrain the model based on the environment while simultaneously acting in the environment. Future work is also necessary to predict more frames in advance so that a scheduler can design a schedule that can hold for a longer period of time.

REFERENCES

- [1] Index, Cisco Visual Networking, “Global mobile data traffic forecast update, 2016–2021 white paper,” Cisco: San Jose, CA, USA, 2017.
- [2] C. Cormio and K. R. Chowdhury, “A survey on MAC protocols for cognitive radio networks,” *Ad Hoc Netw.*, vol. 7, no. 7, pp. 1315–1329, Sep. 2009.
- [3] DARPA SC2. *DARPA Spectrum Collaboration Challenge*. Accessed: Apr. 2018. [Online]. Available: <https://spectrumcollaborationchallenge.com/>
- [4] R. Mennes, M. Camelo, M. Claeys, and S. Latré, “A neural-network-based MF-TDMA MAC scheduler for collaborative wireless networks,” in *Proc. IEEE Wireless Commun. Netw. Conf.*, Barcelona, Spain, Apr. 2018, pp. 1–6.
- [5] Y. Chu, P. D. Mitchell, and D. Grace, “ALOHA and Q-learning based medium access control for wireless sensor networks,” in *Proc. Int. Symp. Wireless Commun. Syst.*, Aug. 2012, pp. 511–515.
- [6] C. J. C. H. Watkins and P. Dayan, “Q-learning,” *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, May 1992.
- [7] K.-H. Phung, B. Lemmens, M. Mihaylov, L. Tran, and K. Steenhaut, “Adaptive learning based scheduling in multichannel protocol for energy-efficient data-gathering wireless sensor networks,” *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 2, Feb. 2013, Art. no. 345821.
- [8] A. R. Syed, K. L. A. Yau, H. Mohamad, N. Ramli, and W. Hashim, “Channel selection in multi-hop cognitive radio network using reinforcement learning: An experimental study,” in *Proc. Int. Conf. Frontiers Commun., Netw. Appl.*, Nov. 2014, pp. 1–6.
- [9] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, “Deep reinforcement learning for dynamic multichannel access in wireless networks,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 2, pp. 257–265, Jun. 2018.
- [10] A. M. Cavalcante et al., “Performance evaluation of LTE and Wi-Fi coexistence in unlicensed bands,” in *Proc. IEEE 77th Veh. Technol. Conf.*, Jun. 2013, pp. 1–6.
- [11] I. Demirkol, C. Ersoy, and F. Alagoz, “MAC protocols for wireless sensor networks: A survey,” *IEEE Commun. Mag.*, vol. 44, no. 4, pp. 115–121, Apr. 2006.
- [12] M. Hadded, P. Muhlethaler, A. Laouiti, R. Zagrouba, and L. A. Saidane, “TDMA-based MAC protocols for vehicular ad hoc networks: A survey, qualitative analysis, and open research issues,” *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2461–2492, 4th Quart., 2015.
- [13] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert, “6TiSCH: Deterministic IP-enabled industrial Internet (of Things),” *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 36–41, Dec. 2014.
- [14] Á. A. Farfás and D. Dujovne, “A queue-based scheduling algorithm for PCE-enabled industrial Internet of Things networks,” in *Proc. 6th Argentine Conf. Embedded Syst.*, Aug. 2015, pp. 31–36.
- [15] A. Tinka, T. Watteyne, K. Pister, and A. M. Bayen, “A decentralized scheduling algorithm for time synchronized channel hopping,” in *Proc. Int. Conf. Ad Hoc Netw.* Berlin, Germany: Springer, 2010, pp. 201–216.
- [16] A. Morell, X. Vilajosana, J. L. Vicario, and T. Watteyne, “Label switching over IEEE802.15.4e networks,” *Trans. Emerg. Telecommun. Technol.*, vol. 24, no. 5, pp. 458–475, Aug. 2013.
- [17] M. Esteban, S. Kathleen, and L. Steven, “A distributed density optimized scheduling function for IEEE 802.15.4e TSCH networks,” *Trans. Emerg. Telecommun. Technol.*, vol. 29, no. 7, Jul. 2018, Art. no. e3420. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3420>
- [18] M. R. Palattella et al., “On-the-fly bandwidth reservation for 6TiSCH wireless industrial networks,” *IEEE Sensors J.*, vol. 16, no. 2, pp. 550–560, Jan. 2016.
- [19] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, “A survey on spectrum management in cognitive radio networks,” *IEEE Commun. Mag.*, vol. 46, no. 4, pp. 40–48, Apr. 2008.
- [20] A. Ioannidou, E. Chatzilari, S. Nikolopoulos, and I. Kompatiari, “Deep learning advances in computer vision with 3D data: A survey,” *ACM Comput. Surv. (CSUR)*, vol. 50, no. 2, Jun. 2017, Art. no. 20.
- [21] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in deep learning based natural language processing,” *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018.
- [22] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, p. 436, 2015.
- [23] V. Mnih et al., “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, p. 529, 2015.
- [24] M. R. Palattella, N. Accettura, M. Dohler, L. A. Grieco, and G. Boggia, “Traffic aware scheduling algorithm for reliable low-power multi-hop IEEE 802.15.4e networks,” in *Proc. IEEE 23rd Int. Symp. Pers., Indoor Mobile Radio Commun.*, Sep. 2012, pp. 327–332.
- [25] D. P. Kingma and J. Ba. (Dec. 2014). “Adam: A method for stochastic optimization.” [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [26] S. Datta, I. Seskar, M. Demirhan, S.-C. Mau, and D. Raychaudhuri, “Ad-hoc extensions to the 802.15.3 MAC protocol,” in *Proc. 6th IEEE Int. Symp. World Wireless Mobile Multimedia Netw.*, Jun. 2005, pp. 293–298.



RUBEN MENNES received the master's degree in computer science from the University of Antwerp, Belgium, in 2016. In 2016, he joined the Department of Mathematics and Computer Science, University of Antwerp and imec, where he is currently pursuing the Ph.D. degree. The focus of his research is mainly on collaborative wireless networks and smart wireless networks, the cross-domain between wireless network technologies, and machine learning.



INGRID MOERMAN received the degree in electrical engineering and the Ph.D. degree from Ghent University, in 1987 and 1992, respectively, where she became a part-time professor, in 2000. She is currently a staff member of IDLab, a core research group of imec with research activities embedded in Ghent University and University of Antwerp. She is coordinating the research activities on mobile and wireless networking, and is leading a research team of about 30 members at Ghent University.

Her main research interests include collaborative and cooperative networks, intelligent cognitive radio networks, real-time software-defined radio, flexible hardware/software architectures for runtime radio/network control and management, the Internet of Things, low power wide area networks (LPWAN), high-density wireless access networks, next generation wireless networks, and experimentally-supported research. She has a longstanding experience in running and coordinating national and EU research funded projects. At the European level, she is, in particular, very active in the area of Future Internet Research Experimentation and Future Connectivity Systems, where she has coordinated and is coordinating several FP7/H2020 projects (CREW, WiSHFUL, eWINE, and ORCA). She is the Leader of the team SCATTER in the DARPA spectrum collaboration challenge, which is one of the prizewinners in phase 1 of the challenge (<https://www.darpa.mil/news-events/2017-12-21a>). She has authored or coauthored over 700 publications in international journals or conference proceedings.



MAXIM CLAEYS received the master's degree in computer science from Ghent University, Belgium, in 2012. He is currently pursuing the Ph.D. degree in the field of Quality of Experience optimization for multimedia delivery, Ghent University, where he joined the Department of Information Technology, in 2012. After obtaining his Ph.D. degree with his dissertation titled "Quality-Driven Management of Video Streaming Services in Segment-Based Cache Networks", in 2017, he joined the IDLab Research Group, University of Antwerp, where he currently holds a postdoctoral position.



FELIPE A. P. DE FIGUEIREDO received the B.S. and M.S. degrees in telecommunications from the Instituto Nacional de Telecomunicações (INATEL), Minas Gerais, Brazil, in 2004 and 2011, respectively. He is currently pursuing the Ph.D. degree with the Internet Technology and Data Science Laboratory, Ghent University, Ghent, Belgium. He has been working with the R&D of telecommunications systems for over 10 years. His research interests include digital signal processing, digital communications, mobile communications, MIMO, multicarrier modulations, and FPGA development.



IRFAN JABANDŽIĆ received the master's degree in automatic control and electronics from the Faculty of Electrical Engineering Sarajevo, Bosnia and Herzegovina, in 2014. After working in area of embedded engineering for two years, in 2016, he joined the Internet Technology and Data Science Laboratory, Ghent University, Ghent, Belgium, where he is currently pursuing the Ph.D. degree. His main research interests are in the fields of wireless networks and wireless sensor networks, focusing on flexible and adaptive MAC architectures and heterogeneous wireless technologies coexistence.



STEVEN LATRÉ received the M.Sc. degree in computer science from Ghent University, Belgium, in 2006, and the Ph.D. degree in computer science engineering from Ghent University, in 2011, on autonomic network management. He is leading the IDLab Research Group, University of Antwerp, consisting of over 80 researchers. He is an Assistant Professor with the University of Antwerp and imec, Belgium. His personal research expertise focuses on deep reinforcement learning and wireless network management. He has authored or coauthored over 70 papers published in international journals or in the proceedings of international conferences. He is a member of the Young Academy Belgium. He is a recipient of the IEEE COMSOC Award for best the Ph.D. in network and service management 2012, the IEEE NOMS Young Professional Award 2014, and the IEEE COMSOC Young Professional Award 2015.