Automatic Jamming Signal Classification in Cognitive UAV Radios

Ali Krayani, *Student Member, IEEE,* Atm S. Alam, *Member, IEEE,* Lucio Marcenaro, *Member, IEEE,* Arumugam Nallanathan, *Fellow, IEEE,* and Carlo Regazzoni, *Senior Member, IEEE*

Abstract—**The integration of Cognitive Radio (CR) with Unmanned Aerial Vehicles (UAVs) is an effective step towards relieving the spectrum scarcity and empowering the UAV with a high degree of intelligence. The dynamic nature of CR and the dominant line-of-sight links of UAVs poses serious security challenges and make the CR-UAV prone to a variety of attacks as malicious jamming. Joint jammer detection and automatic jammer classification is a powerful approach against the physical layer threats by identifying multiple jammers attacking the network that realize a crucial stage towards efficient inference management. This paper proposes a novel method for joint detection and automatic classification of multiple jammers attacking with different modulation schemes. The method is based on learning a representation of the radio environment encoded in a Generalized Dynamic Bayesian Networks (GDBN) whilst multiple GDBN models represent various jamming signals under different modulation schemes. The CR-UAV performs multiple predictions online in parallel and evaluates multiple abnormality measurements based on a Modified Markov Jump Particle Filter (M-MJPF) to select the best-fit model that explains the detected jammer and recognize the modulation scheme accordingly. The simulated results demonstrate that the proposed GDBN-based method outperforms both Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) in terms of classification accuracy and achieves a higher degree of explainability of its own decisions by interpreting causes and effects at hierarchical levels under the Bayesian learning and reasoning processes.**

Index Terms—**Cognitive Radio, Unmanned Aerial Vehicles, Dynamic Bayesian Network, Bayesian Filtering, Modulation Recognition.**

I. Introduction

The advent of the Unmanned Aerial Vehicles (UAVs) and its recent rapid growth in a myriad of applications have got plenty of interest to be leveraged in the fifth-generation (5G) technology [1]. Owing to the dynamic deployment flexibility, high mobility and strong Line-of-Sight (LoS) communication links of UAVs, they are regarded as an important complement to the terrestrial networks from the sky [2]. However, UAV-based communications will face several problems such as spectrum scarcity due to the explosively increasing number of connected UAVs [3], energy-efficiency due to the on-board limited battery lifetime [4], [5] and physical layer security (e.g. jamming attacks) due to the open nature of ground-to-air wireless channels and the dominant LoS propagation links [6]. Cognitive Radio (CR) is considered as one of the most promising solutions that can tackle the aforementioned problems due to its capability in pursuing its own goals autonomously, learning the radio environment, monitoring and predicting the environmental changes and infer the appropriate action that can be performed [7]. A series of recent studies have investigated the integration of CR and UAVs (i.e. Cognitive UAV Radios) for different aspects such as, communication capacity and Quality of Service (QoS) improvement [8], [9], collision avoidance stability [10], trajectory optimization [11], energy harvesting [12], spectrum scarcity [13], energy efficiency optimization [14], [15], interference coordination [16], data dissemination [17], joint sub-carrier and power allocation [18], [19], and for secure communications [20].

UAV communications are susceptible to jamming attacks by terrestrial malicious nodes distributed over a large area on the ground that can exploit the strong LoS channels to launch powerful attacks and interfere with the UAV resulting in communication failure [21]. In addition, smart jammers equipped with cognitive capabilities can pose more security threats. They can sense the radio spectrum and discover the UAV's transmission policy to update their attack strategy and force the UAV to learn wrong behaviours and take misleading actions. Thus, enhancing the physical layer security is of great concern to ensure reliable communications and successfully deploy cognitive UAV Radios. This work focuses on the joint detection and classification of multiple jammers attacking the UAV's control and command link. Jammer detection is the first essential step to determine the radio situation, while jammer classification is an important stage towards an efficient interference management solution [22].

In our previous investigations ([23], [24], [25]), we introduced the concept of Self-Awareness (SA) in CR to empower the radio with a brain for high-level intelligence. The SA module allows the radio to reach the capability of learning a representation of the radio environment encoded in a generative dynamic model to be stored in the radio's brain. The generative model describes in a probabilistic manner how a given signal might have been generated by predicting new data samples and inferring the hidden states that caused the observed signal. This allows the radio to evaluate the situation through different abnormality measurements at multiple hierarchical levels and understand if the situation is normal or abnormal

Ali Krayani is with the Department of Electrical, Electronic, Telecommunications Engineering and Naval Architecture, University of Genoa, 16145 Genoa, Italy, and also with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, U.K. (e-mail: ali.krayani@edu.unige.it, a.krayani@qmul.ac.uk).

Lucio Marcenaro and Carlo Regazzoni are with the Department of Electrical, Electronic, Telecommunications Engineering and Naval Architecture, University of Genoa, 16145 Genoa, Italy. (e-mails: lucio.marcenaro@unige.it, carlo.regazzoni@unige.it).

Atm S. Alam and Arumugam Nallanathan are with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, U.K. (e-mails: a.alam@qmul.ac.uk, a.nallanathan@qmul.ac.uk).

(e.g. detecting normal and jamming signals). If an abnormality is detected, the radio can characterize it to discover the new rules and encode them incrementally in a new dynamic model. However, an important question that needs to be addressed here is *when* the radio must learn a new model based on the current experience? Abnormality detection is not enough to answer this question. In contrast, abnormality classification is an indispensable functionality towards this understanding by comparing multiple abnormality signals generated by multiple models already learned in previous experiences and evaluating how much the current situation differs from them.

In this work, we extend the SA module by adding the Abnormality Classification functionality to jointly detect and classify multiple jammers according to their modulation scheme. Initially, the Cognitive-UAV begins with null memory without any a priori knowledge about the radio environment, supposing that no signals present and observations are due to a stationary noise process, i.e., a process evolving according to static rules. Then the Cognitive-UAV starts to build up knowledge about the environment by exploiting the Generalized Errors (i.e. prediction errors) at the state level to discover the real dynamic rules of how the signals (related to the commands) are behaving inside the radio spectrum. These errors can be clustered in an unsupervised manner to learn the corresponding reference Generalized Dynamic Bayesian Network (GDBN) model under a normal situation (i.e. where the jammer is absent). The cognitive-UAV can use the acquired reference GDBN model in future experiences to predict the commands that it is supposed to receive under normal circumstances by employing a Modified Markov Jump Particle Filter (M-MJPF). Consequently, it can detect any jamming attack using abnormality measurements at hierarchical levels as well as calculating the new Generalized Errors once an abnormality is detected. This computational scheme assumes that the cognitive-UAV generates probabilistic predictions continually on what commands come next based on the rules encoded in the reference model and compares those predictions at different hierarchical levels with the UAV's real communications stimuli that lead to the computation of hierarchical Generalized Error signals. These Generalized Error signals are of great importance to understand why the current dynamic model can not explain the current radio situation and how we can update the model to adapt to the abnormal situation. In addition, those errors can be informative enough to understand the cause behind the abnormality and provide a way to extract the jammer's signal. Exploiting the Generalized Errors allows extracting the jammer's signal and guides the cognitive-UAV to learn a separated GDBN model for each detected jammer. In this way, the cognitive-UAV's brain consists of a reference GDBN model representing the command signals that the UAV is expecting to receive in a typical radio situation and a set of multiple GDBN models representing the jammers' behaviours incrementally learned in previous experiences under different modulation schemes in abnormal radio situations. The link between the reference model and the other ones is described by the Generalized Errors provided by the reference model and used by the set of models as observations. In other words, the UAV uses the reference model to infer the hidden states of the radio environment, detect abnormalities in case of attack and calculate the Generalized Errors. Those errors can be used as observations by the other set of models (representing the jammers) while performing multiple predictions in parallel to evaluate the best GDBN model (inside the set) that better explains the current observation (i.e. Generalized Error provided by the reference model) and recognize the modulation scheme of the jammer consequently. The classification task is formulated in terms of objective function that maximizes the Bayesian model evidence (or marginal likelihood), which is the probability of observing signals conditioned to a model generating those signals or to minimize the surprise (i.e. abnormality). This means that we will test different hypotheses (i.e. models) and weighting them to select the model that has the greatest evidence and minimum surprise (i.e. abnormality).

The main contributions of this work are as follows: *i)* we propose an efficient learning mechanism within the Growing Neural Gas (GNG) to capture the dynamic transitions of the radio signal modulated under certain modulation scheme; *ii)* we formulate the modulation classification problem in terms of an objective function that aims to minimize the surprise (i.e. abnormality) by testing different models learned by the radio and weighting them to select the model that causes the minimum surprise and thus that better explains the modulation scheme of the detected jamming signals; *iii)* extensive simulations verify that the proposed GDBN-based framework for automatic jamming signal classification performs with superiority classification accuracy than LSTM and CNN; *iv)* the GDBN models can achieve higher interpretability than Deep Learning-based models since they can explain the predictions explicitly at hierarchical levels and use the abnormality measurements and Generalized Errors as self-information to keep learning by understanding incrementally.

II. Related Work

Radio Signal Classification is an important task in many communications systems [26]. It is mainly based on Automatic Modulation Classification (AMC) that servers as an intermediate step between signal detection and signal demodulation. AMC is widely used in both civil and military fields and finds applications in Cognitive Radio (CR) for efficient spectrum management, and secure communications [27], [28]. Traditional approaches for modulation classification include Likelihood-Based (LB) approach and Feature-Based (FB) approach [29], [30], [31]. The LB approach is based on comparing the likelihood ratio of the received signal with a threshold. The LB is optimal in the Bayesian sense by minimizing the probability of false classification. However, it is computationally complex and requires an estimation of parameters (e.g. channel parameters) to calculate the likelihood probability, which is not always possible in real radio scenarios as in CR. Also, the performance degrades in the presence of phase and frequency offset. The FB approach does not require an estimation of parameters, and it is based on some features as the variance of the centered normalized signal amplitude, phase and frequency. Thus it is less complicated compared to the LB approach and easy to use. Even though it is suboptimal, however, a proper design allows achieving optimal performance.

Deep learning-based methods for AMC are extensively investigated in the literature. In [28], a Long Term Short Memory (LSTM) is used for this purpose where the data-augmentation methods (i.e. rotation, flip, and Gaussian noise) are studied to cope with small datasets by expanding the data and thus improving the robustness of deep learning models and classification accuracy as well. However, expanding the dataset might lead to several problems as increasing latency which is vital in some applications as vehicular communications. An improved Convolutional Neural Network (CNN)-based automatic modulation classification network (IC-AMCNet) is proposed in [32] where different types of layers as convolutional, dropout and Gaussian noise are applied for regularization and to overcome the overfitting issue. In addition, a small number of filters is used in each layer to reduce the processing time. Authors in [33] proposed a gated recurrent, residual network (GrrNet) for modulation classification consisting of a ResNet extractor module, fusion module and GRU-based classification module. However, both [32], and [33] used supervised training by feeding the networks with the signal features along with the labels that indicate the modulation scheme of the input. This may require a significant effort to label large amounts of training examples that can be expensive and time-consuming. Interesting research has been conducted in [34] to study the visualization methods for deep learning-based radio modulation classifiers (based on CNN and LSTM) and thus to understand the modulation classification mechanism for better interpretability. However, such visualization techniques do not exploit the extracted radio features in an unsupervised way, allowing the radio to encode the dynamic changes between different modulation schemes, which enhance the learning and perception processes of the radio.
In [35], a compressive convolutional neural network (CCNN) is proposed for AMC where different regular constellation images and contrast-enhanced grid constellation images are generated from received signals and used as network inputs. In addition, a compressive loss constraint is proposed to train the CCNN to capture high-dimensional features as well as utilizing the intra-class compactness and inter-class separability to enhance robustness performance for a different order of modulations. Simulation results showed the superior classification compared with RNN, DNN and CNN. Other works also converted the radio signal into images, e.g. Choi-Williams time-frequency distribution (CWD) image [36], Feature Point (FP) images [37], Contour Stellar Image (which gets more color feature compared to the Constellation Diagram) [38], amplitude spectrums of bispectrum (ASB) images [39], cyclic spectrum images [40]. The studies mentioned above have obtained promising results in modulation classification. However, they require high computational processing to convert signals to images that can be unfeasible in the UAV scenario, and they might lose important information and ignore crucial details by passing from time-frequency representation to image representation.

In this article, we propose a novel GDBN-based method for AMC which has a high degree of interpretability that can determine and associate causes and effects at hierarchical levels thanks to the underlying Bayesian learning and reasoning processes. In addition, it achieves a high degree of explainability of its own decisions where hidden variables used in the generalized model make it possible to draw explicit causal dynamic probabilistic relationships among continuous signals and their symbolic higher-level counterparts to study how significance each parameter in contributing to the final decision.

III. System Model

The system model depicted in Fig. 1 consists of a cellular-connected UAV, Base Station (BS), a UAV operator, and multiple terrestrial jammers aiming to attack the Command and Control (C2) link by sending false commands to alter the trajectory and take control of the UAV. The jammers are smart; they can identify and locate the resources allocated to the UAV by the BS inside the radio spectrum and attack consequently using different modulation schemes. The propagation model is shown in Fig. 2. We assume that the ground-to-UAV link is always a Line-Of-Sight (LOS) under an Additive White Gaussian Noise (AWGN) channel condition. The 3GPP path loss model defined in [41] is adopted under the RMa-AV scenario. The BS sends continuously a Radio Frame of 10 ms to the active users already synchronized with BS in the cell and allocates a specific number of sub-carriers to each user for a predetermined slot which are organized in Physical Resource Blocks (PRBs). We assume that the BS follows the third allocation scheme for UAV command & control (C2) data as mentioned in [42], since the 3GPP specifications require a maximum of 100 kb/s for C2 data, latency of 50 ms, and inter-arrival time (also known as Transmission Time Interval TTI)

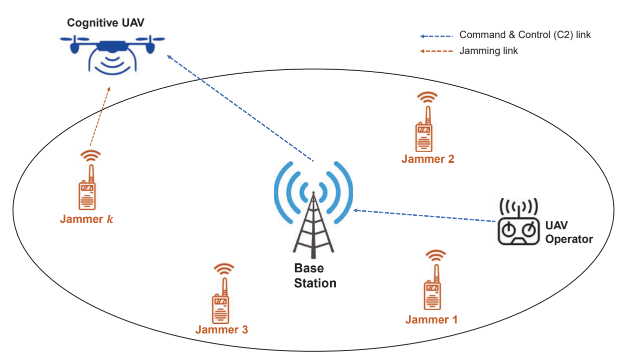


Fig. 1. Illustration of the system model.

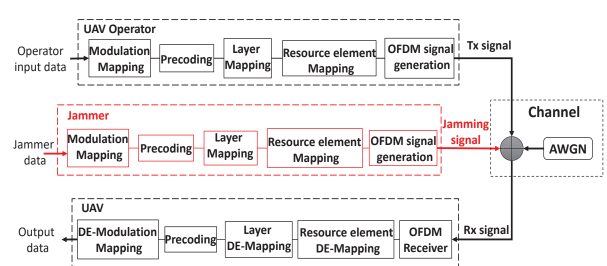


Fig. 2. Illustration of the Propagation Model including the Operator, Jammer and the UAV.

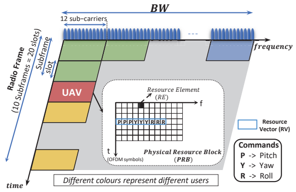


Fig. 3. Physical resource allocation and RF structure.

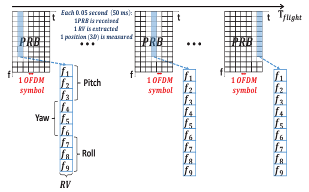


Fig. 4. Timing of the PRBs and RVs received by the UAV.

of 100 ms for efficient management of a UAV [43]. The UAV is equipped with a GPS receiver and RF antenna which are supposed to be synchronized, i.e. the UAV receives one PRB and measures the 3D position (by the GPS receiver) every 50 ms. The Pitch, Yaw, and Roll commands are transmitted over 9 consecutive sub-carriers (frequency domain) within 1 OFDM symbol (time domain) in each PRB received by the UAV, where we call these Resource Elements (REs) as a Resource Vector (RV) as shown in Fig. 3 and Fig. 4. The remaining sub-carriers and OFDM symbols of the PRB are related to other information sent to the UAV. In our study, the data is extracted after the OFDM receiver block. More precisely, after the FFT where at this level the UAV can scan all the Resource Elements (REs) of the time-frequency grid and capture the IQ data without any extra hardware. In our study, we considered the RVs solely because we aim to analyze the command signals only. However, this can be simply extended in future investigations to consider the whole PRB.

IV. Proposed Automatic Jamming Modulation Classification

A. Radio Environment Representation

In our approach, we use a generalized state-space model to represent the radio environment. We assume, that the observed signal $\tilde{Z}_t$ is a linear combination of one latent generalized state $\tilde{X}_t$ that represents the direct cause of the observation and a multivariate generalized Gaussian noise $\tilde{v}_t$ ($\tilde{v}_t \sim \mathcal{N}(0, \Sigma_{\tilde{v}_t})$) and defined as follows:

$$\tilde{Z}_t = H\tilde{X}_t + \tilde{v}_t, \tag{1}$$

where $H \in \mathbb{R}^{d \times d}$ is the matrix that maps hidden states to observations. The generalized observation $\tilde{Z}_t \in \mathbb{R}^d$ comprises the signal's states in terms of I and Q components and the corresponding first-order temporal derivatives ($\dot{I}$, $\dot{Q}$), where d is the space dimensionality and it is equal to the total number of sub-carriers where the commands are transmitted.
The evolution of the hidden generalized states $\tilde{X}_t$ can be approximated as a linear combination of the previous state $\tilde{X}_{t-1}$ which is guided by the deep hidden cause $\tilde{S}_t$ and formulated as follows:

$$\tilde{X}_t = A\tilde{X}_{t-1} + BU_{\tilde{S}_t} + \tilde{w}_t, \tag{2}$$

where $A \in \mathbb{R}^{d \times d}$ and $B \in \mathbb{R}^{d \times d}$ are the dynamic model and control model matrices while $U_{\tilde{S}_t}$ is the control vector and $\tilde{w}_t$ is the generalized process noise such that $\tilde{w}_t \sim \mathcal{N}(0, \Sigma_{\tilde{w}_t})$. The generalized superstates are discrete variables that explain the discrete regions of the signal. The evolution of these variables is expressed in the following form:

$$\tilde{S}_t = f(\tilde{S}_{t-1}) + \tilde{w}_t, \tag{3}$$

where $f(.)$ is a non-linear function that describes the relationship between the previous superstate and the current superstate, realizing the dynamics of how the signal is transiting among the discrete regions and its evolution over time.

B. Learning Stage

We propose to learn a GDBN as a representation of the radio environment. GDBN can model dynamic processes describing the signal's temporal evolution at hierarchical levels. GDBN provides a graphical structure representing hidden and observed states in terms of random state variables encoding the conditional dependencies among them and specifying a compact parameterization of the model. Two sets of parameters can represent it. The first includes the number of nodes in each time slice and the corresponding topology, while the second set consists of the conditional probability distributions (CPDs) described by edges of the network. Learning a GDBN consists of parameter learning and structure learning. The former is the process of learning the distributions of discrete or continuous hidden variables in the GDBN, while the latter uses data to learn the links among random variables in the GDBN. Both parameter and structure learning depends on the generalized state-space model in question. The proposed GDBN consists of three levels. The discrete level stands for the discrete variables describing the discrete regions of the signal. The medium level stands for the continuous states encoded inside each discrete region, and the bottom level stands for the observation.

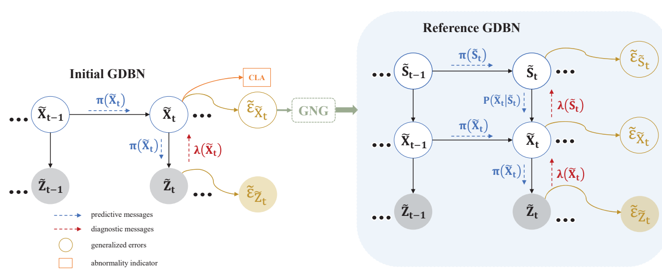


Fig. 5. Schematic that illustrates the process of learning incrementally by exploiting the generalized errors. The cognitive-UAV starts perceiving the environment using an initial GDBN model with a static assumption about the signal's evolution. While the cognitive-UAV is predicting and sensing the environment, it can calculate the generalized errors ($\tilde{\mathcal{E}}_{\tilde{X}_t}$) and stored them to perform the clustering after finishing the current experience.

The cognitive-UAV aims to learn and encode the radio environment representation in a GDBN under a normal radio situation. Initially, it starts with null memory without prior knowledge about the surrounding radio environment assuming that signals are evolving according to static rules. Thus, the cognitive-UAV starts perceiving the radio environment using an initial GDBN (consisting solely of the observation and state levels) on which an Unmotivated Kalman Filter (UKF) is employed (i.e., a null force filter with a static assumption about the environmental states) to predict the continuous signal using the following equation:

$$\tilde{X}_t = A\tilde{X}_{t-1} + \tilde{w}_t, \tag{4}$$

and then interpreting the received generalized observations $\tilde{Z}_t$ that comprises the variable and its generalized coordinates of motion coming from the receiver. In fact, since the signals inside the radio spectrum are following a certain dynamic behavior, the cognitive-UAV will detect abnormalities all the time and calculate the generalized errors ($\tilde{\varepsilon}_{\tilde{Z}_t}$) which are the differences between predictions and observations and it is expressed as:

$$\tilde{\varepsilon}_{\tilde{Z}_t} = \tilde{Z}_t - H\tilde{X}_t. \tag{5}$$

The UKF works by predicting the generalized states ($\tilde{X}_t$), projecting this into the measurement space and taking the difference between the current observed generalized measurement ($\tilde{Z}_t$) and the predicted one. This difference is known as innovation, which is computed in the measurement space. Thus, to project this difference back to the generalized state-space we must use the following formula:

$$\tilde{\varepsilon}_{\tilde{X}_t} = H^{-1}\tilde{\varepsilon}_{\tilde{Z}_t} = H^{-1}(\tilde{Z}_t - H\tilde{X}_t) = H^{-1}\tilde{Z}_t - \tilde{X}_t. \tag{6}$$

The generalized errors ($\tilde{\varepsilon}_{\tilde{X}_t}$) that capture the real dynamics of the signal are used as input to an unsupervised clustering technique, the Growing Neural Gas (GNG) (refer to Fig. 5). GNG encodes the generalized errors into discrete regions described by a set of neurons or superstates $\mathbf{S}$, such that:

$$\mathbf{S} = \{S_1, S_2, \dots, S_M\}, \tag{7}$$

where M is the total number of neurons. After obtaining the neurons, we analyzed how the signal is transiting between them to learn the transition matrix Π by estimating the transition probabilities $\pi_{ij} = P(S_t = i|S_{t-1} = j)$ over a period of time (i.e. the training time), where $i, j \in \mathbf{S}$. The $M \times M$ transition matrix is defined as:

$$\Pi = \begin{bmatrix} \pi_{11} & \dots & \pi_{1M} \\ \vdots & \ddots & \vdots \\ \pi_{M1} & \dots & \pi_{MM} \end{bmatrix}. \tag{8}$$

Thus, the generalized superstates ($\tilde{\mathbf{S}}$) can be expressed in terms of current discrete variable S_t and the corresponding event e_t^{ij} in the following way:

$$\tilde{S}_t = [S_t \ \dot{S}_t] = [S_t \ e_t^{ij}]. \tag{9}$$

An event can be described as a change at the discrete level (i.e., the transition from a certain superstate to a new one), such that:

$$e_t^{ij} = (S_{t-1} = i, S_t = j) \mid i \neq j. \tag{10}$$

The null event can be defined as e_t^0 when $i = j$. Furthermore, since the radio environment is dynamic and varies with time, estimating the temporal (i.e., time-varying) transition matrix Π_τ is of great interest. Π_τ encodes not only the possible transitions (transition probabilities) at the discrete level but also when those transitions or events will occur (i.e., the time required for a particular event to occur) and defined as:

$$\Pi_\tau = \begin{bmatrix} \pi_{11,\tau} & \dots & \pi_{1M,\tau} \\ \vdots & \ddots & \vdots \\ \pi_{M1,\tau} & \dots & \pi_{MM,\tau} \end{bmatrix}, \tag{11}$$

where $\pi_{ij,\tau} = P(S_t = i|S_{t-1} = j, \tau)$ realizing a new condition in transiting to the new superstate $S_t = i$ after being in $S_{t-1} = j$ for a certain time (i.e., τ). It is worth noting that as the dynamics of the signal become faster the time τ become smaller. So, the time-varying transition matrix encodes how transition probabilities vary with time; some probabilities increase and others decrease as time evolves, allowing to keep tracking the dynamic changes in the environment.

Each discrete variable $\tilde{S}_m$ ($\tilde{S}_m \in \mathbf{S}$) is associated with specific statistical properties as covariance matrix $\Sigma_{\tilde{S}_m}$ and *generalized mean value* $\tilde{\mu}_{\tilde{S}_m}$ defined as:

$$\tilde{\mu}_{\tilde{S}_m} = [\mu_{\tilde{S}_m}, \dot{\mu}_{\tilde{S}_m}], \tag{12}$$

that consists of the mean value $\mu_{\tilde{S}_m}$ describing the average of all the data samples encoded in this superstate in terms of I and Q as well as the average of the corresponding derivatives (i.e. $\dot{\mu}_{\tilde{S}_m}$).

C. Testing Stage

GDBN can decompose data with complex and non-linear dynamics into segments that are explainable by simpler dynamical units. The Modified Markov Jump Particle Filter (M-MJPF) (which is an evolved version of the MJPF introduced in [44]) is a specific class of switching dynamic systems employed on the learned GDBN model to discover the dynamical units and explain their switching behaviour and their dependency on both observations and discrete/continuous hidden states during the real-time process. The M-MJPF uses a combination of Particle Filter (PF) to predict the generalized superstates at the discrete level and a bank of Kalman Filters (KFs) at the continuous level to predict the generalized states. The M-MJPF within the Bayesian Filtering framework provides two probabilistic inference modes: predictive or causal inference (top-down) and diagnostic inference (bottom-up). The predictive inference is based on passing predictive messages in a top-down manner, where predictions are performed based on the acquired knowledge in previous experience. The diagnostic inference is based on propagating likelihood messages after receiving the real measurement in a backward manner from bottom to up, where the likelihood messages evaluate how much the observation matches the predictions at hierarchical levels to update the belief in hidden variables accordingly. PF relies on a proposal density encoded in the learned transition matrix to sample a set of particles realizing the predicted superstates at the discrete level. Initially, PF propagates N equally weighted particles (<.>) associated with a specific superstate, such that:

$$< \tilde{S}_t^n, W_t^n > \sim < \pi(\tilde{S}_t), 1/N >, \ n \in N. \tag{13}$$

It is worth noting that in our scenario, there is no need to use a large number of particles since the discrete level consists of a finite number of discrete regions. Thus, it is sufficient to use few particles to represent the posterior accurately (unlike the continuous space which may need a huge number of particles to represent the posterior correctly). After that, a KF is employed for each particle $(.^n)$ to predict $\tilde{X}_t$. The prediction at this level (continuous level) is guided by the prediction performed at the higher level as pointed out in (2) and can be expressed in terms of the conditional probability $P(\tilde{X}_t|\tilde{X}_{t-1}, \tilde{S}_t)$. In (2), the control vector $(U_{\tilde{S}_t})$ which realize the dynamic flow of the signal starting from the previous state is encoded in the generalized mean value defined in (12), hence $U_{\tilde{S}_t} = \dot{\mu}_{\tilde{S}_m}$ which by the way depends on the predicted generalized superstate $(\tilde{S}_t)$ at the discrete level. The posterior probability associated with the predicted generalized state is given by:

$$\pi(\tilde{X}_t) = P(\tilde{X}_t, \tilde{S}_t|\tilde{Z}_{t-1}) = \int P(\tilde{X}_t|\tilde{X}_{t-1}, \tilde{S}_t)\lambda(\tilde{X}_{t-1})d\tilde{X}_{t-1}, \tag{14}$$

where $\lambda(\tilde{X}_{t-1})$=$P(\tilde{Z}_{t-1}|\tilde{X}_{t-1})$. Accordingly, a message backward propagated from the bottom-level to the higher levels once a new evidence $\tilde{Z}_t$ is received can be exploited to adjust the expectations in hidden variables and estimate the posterior probability $P(\tilde{X}_t, \tilde{S}_t|\tilde{Z}_t)$ which is defined as:

$$P(\tilde{X}_t, \tilde{S}_t|\tilde{Z}_t) = \pi(\tilde{X}_t)\lambda(\tilde{X}_t). \tag{15}$$

Consequently, the likelihood message $\lambda(\tilde{S}_t)$ is propagated towards the top-level to update the belief in the hidden discrete variable by updating the weights according to:

$$W_t^n = W_t^n \lambda(\tilde{S}_t), \tag{16}$$

$\lambda(\tilde{S}_t)$ is a discrete probability distribution represented by:

$$\lambda(\tilde{S}_t) = \lambda(\tilde{X}_t)P(\tilde{X}_t|\tilde{S}_t) = P(\tilde{Z}_t|\tilde{X}_t)P(\tilde{X}_t|\tilde{S}_t), \tag{17}$$

where $P(\tilde{X}_t|\tilde{S}_t) \sim \mathcal{N}(\mu_{\tilde{S}_m}, \Sigma_{\tilde{S}_m})$ denotes a Gaussian distribution with mean $\mu_{\tilde{S}_m}$ and covariance $\Sigma_{\tilde{S}_m}$. While, $\lambda(\tilde{X}_t) \sim \mathcal{N}(\mu_{\tilde{Z}_t}, R)$ denotes a Gaussian distribution with mean $\mu_{\tilde{Z}_t}$ and covariance R. The multiplication between $\lambda(\tilde{X}_t)$ and $P(\tilde{X}_t|\tilde{S}_t)$ can be estimated by calculating the Battacharyya distance (D_B) as follows:

$$D_B\big(\lambda(\tilde{X}_t), P(\tilde{X}_t|\tilde{S}_t = \tilde{S}_k)\big) = -ln \int \sqrt{\lambda(\tilde{X}_t)P(\tilde{X}_t|\tilde{S}_t = \tilde{S}_k)}d\tilde{X}_t, \tag{18}$$

where $\tilde{S}_k \in \tilde{\mathbf{S}}$. The vector D_λ containing all the D_B values between $\lambda(\tilde{X}_t)$ and all the superstates in the set $\tilde{\mathbf{S}}$ is here estimated as:

$$D_\lambda = \Big[D_B\big(\lambda(\tilde{X}_t), P(\tilde{X}_t|\tilde{S}_t = \tilde{S}_1)\big), \dots, D_B\big(\lambda(\tilde{X}_t), P(\tilde{X}_t|\tilde{S}_t = \tilde{S}_L)\big)\Big]. \tag{19}$$

Therefore, the vector $\lambda(\tilde{S}_t)$ in terms of probability can be computed as:

$$\lambda(\tilde{S}_t) = \left[\frac{1/D_\lambda(1)}{1/\sum_{l=1}^{L} D_\lambda(l)}, \dots, \frac{1/D_\lambda(L)}{1/\sum_{l=1}^{L} D_\lambda(l)}\right] \tag{20}$$

After updating the weights, particles with very low weights are abandoned while particles with high weights are kept and multiplied so that all particles have equal weight; this process is known as sequential importance resampling (SIR). The logic of the M-MJPF is reported in Algorithm 1.

Algorithm 1: M-MJPF

Input: $\tilde{Z}_t$, N, T, Π, Π_τ, $\tilde{S}_m : (\tilde{\mu}_{\tilde{S}_m}, \Sigma_{\tilde{S}_m}) \ \forall \ \tilde{S}_m \in \mathbf{S}$

for *each* $t \in T$ **do**
  for *each* $n \in N$ **do**
    Prediction at Discrete Level:
    $W_n = \frac{1}{N} \leftarrow$ particle weight
    if $t = 1 \leftarrow$ *initial state* **then**
      Sample: $\tilde{X}_1$ from initial prior density $P(\tilde{X}_1)$
      $\tilde{X}_t = \tilde{X}_1 \leftarrow$ current state
      Estimate: $\tilde{S}_t^n$ from $P(\tilde{X}_t|\tilde{S}_t^n)$
    else $\leftarrow$ *remaining states*
      $\tilde{S}_t^n \sim \pi_\tau(\tilde{S}_{t-1})$
      if $\tilde{S}_t^n = \tilde{S}_{t-1}^n$ **then**
        $\tau^n = \tau^n + 1 \leftarrow$ time elapsed in $\tilde{S}_t^n$
      else
        $\tau^n = 1$
    Prediction at Continuous Level:
    $U_{\tilde{S}_t} = U_{\tilde{S}_t^n} \leftarrow$ Control Vector
    $P_{t-1|t-1} = \Sigma_{\tilde{S}_t^n} \leftarrow$ Covariance matrix
    $\tilde{X}_t = A\tilde{X}_{t-1} + BU_{\tilde{S}_t} \leftarrow$ state
    $P_{t|t-1} = AP_{t-1|t-1}A^\intercal + \Sigma_{\tilde{w}_t} \leftarrow$ covariance
    Calculate: $\pi(\tilde{X}_t)$ using (14)
    Current Observation $\tilde{Z}_t$**:**
    $\lambda(\tilde{X}_t) = P(\tilde{Z}_t|\tilde{X}_t) \leftarrow$ *diagnostic msg1*
    Calculate: $\lambda(\tilde{S}_t)$ using (17) $\leftarrow$ *diagnostic msg2*
    Abnormality measurements:
    Calculate: **KLDA** using (21)
    Calculate: **CLB** using (25)
    Calculate: **CLA** using (27)
    Generalized Errors:
    Calculate: $\tilde{\boldsymbol{\varepsilon}}_{\tilde{Z}_t}^{[1]}$, $\tilde{\boldsymbol{\varepsilon}}_{\tilde{Z}_t}^{[2]}$ using (31), (32)
    Calculate: $\tilde{\boldsymbol{\varepsilon}}_{\tilde{X}_t}^{[1]}$, $\tilde{\boldsymbol{\varepsilon}}_{\tilde{X}_t}^{[2]}$ using (29), (30)
    Calculate: $\tilde{\boldsymbol{\varepsilon}}_{\tilde{S}_t}$ using (24)
    Update Belief in hidden variables:
    $K_t = P_{t|t-1}H^\intercal(HP_{t|t-1}H + \Sigma_{\tilde{Z}_t})^{-1} \leftarrow$ Kalman gain
    $\hat{\tilde{X}}_t = \tilde{X}_t + K_t\tilde{\varepsilon}_{\tilde{X}_t}^{[1]} \leftarrow$ updated state
    $\hat{P}_{t|t} = (1 - K_tHP_{t|t-1}) \leftarrow$ updated covariance
    $W_n = W_n \times \lambda(\tilde{S}_t) \leftarrow$ update particles weight
  SIR resampling

Output: *KLDA*, *CLA*, *CLB*, $\tilde{\varepsilon}_{\tilde{Z}_t}^{[1]}$, $\tilde{\varepsilon}_{\tilde{Z}_t}^{[2]}$, $\tilde{\varepsilon}_{\tilde{X}_t}^{[1]}$, $\tilde{\varepsilon}_{\tilde{X}_t}^{[2]}$, $\tilde{\varepsilon}_{\tilde{S}_t}$

D. Hierarchical Abnormality measurements and Generalized errors

We have seen that predictive and diagnostic reasoning can be used to estimate a joint posterior at different hierarchical levels. An additional process can be done here to evaluate the differences between two messages arriving at a given node and:

- estimate the surprise (i.e. the abnormality) using a proper probabilistic distance (e.g. Bhattacharyya distance, Kullback–Leibler divergence).

- calculate the generalized errors by subtracting the stochastic variables related to predictions and observations.

1) Discrete Level: this level describes the signal's evolution at a high level of abstraction. In order to evaluate to what extent the current signal's evolution matches the predicted one based on the learned and encoded dynamics in the reference GDBN, we used the Symmetric Kullback-Leibler Divergence (D_{KL}) to calculate the similarity between the two messages (that represent discrete probability distributions) entering to node $\tilde{S}_t$, namely, $\pi(\tilde{S}_t)$ and $\lambda(\tilde{S}_t)$ which is formulated as:

$$KLDA = \sum_{i \in \mathcal{S}} Pr(\tilde{S}_t = i) D_{KL}\big(\pi(\tilde{S}_t) || \lambda(\tilde{S}_t)\big) + \sum_{i \in \mathcal{S}} Pr(\tilde{S}_t = i) D_{KL}\big(\lambda(\tilde{S}_t) || \pi(\tilde{S}_t)\big), \tag{21}$$

where $Pr(\tilde{S}_t)$ is the probability of occurrence of each superstate picked from the histogram at time instant t and calculated as follows:

$$Pr(\tilde{S}_t) = \frac{fr(\tilde{S}_t = i)}{N}, \tag{22}$$

where $fr(.)$ is the frequency of occurrence of a specific superstate i and N is the total number of particles propagated by PF and $\mathcal{S}$ is the set consisting of all the winning particles, such that:

$$\mathcal{S} = \big\{ i | Pr(\tilde{S}_t) > 0 \big\}, \; i \in \mathbf{S}. \tag{23}$$

In addition, the generalized errors ($\tilde{\varepsilon}_{\tilde{S}_t}$) associated with the abnormality indicator (21) allows to understand how the jammer affected the discrete level of the reference model. Thus, after detecting the jammer at the discrete level using (21), it is possible to explain why we noticed a high abnormality by calculating the difference between the diagnostic message $\lambda(\tilde{S}_t)$ and the predictive message $\pi(\tilde{S}_t)$, such that:

$$\tilde{\varepsilon}_{\tilde{S}_t} = \lambda(\tilde{S}_t) - \pi(\tilde{S}_t), \tag{24}$$

2) Continuous Level: this level describes the continuous evolution of the signal guided by the evolution at the discrete level. Measuring the distance between the predictive message $\pi(\tilde{X}_t)$ and $P(\tilde{X}_t|\tilde{S}_t)$ using D_B defined as:

$$CLB = -\ln\Big(\mathcal{BC}\big(\pi(\tilde{X}_t), P(\tilde{X}_t|\tilde{S}_t)\big)\Big), \tag{25}$$

where

$$\mathcal{BC} = \int \sqrt{\pi(\tilde{X}_t) P(\tilde{X}_t|\tilde{S}_t)} d\tilde{X}_t, \tag{26}$$

is the Bhattacharyya Coefficient. CLB allows evaluating if the predictions at the continuous level match the predictions at the discrete level and thus explains if the signal's dynamics at both the discrete and continuous level evolve according to the rules learned before in a way that it can explain the received signal. Moreover, it is possible to understand how much the observation supports the predictions using the second abnormality detector at this level defined as:

$$CLA = -\ln\Big(\mathcal{BC}\big(\pi(\tilde{X}_t), \lambda(\tilde{X}_t)\big)\Big), \tag{27}$$

where

$$\mathcal{BC} = \int \sqrt{\pi(\tilde{X}_t)\lambda(\tilde{X}_t)} d\tilde{X}_t. \tag{28}$$

The abnormality indicators mentioned above can be used to evaluate the radio situation and discover if something wrong occurred in the radio environment that violates the dynamic rules learned in previous experience. However, computing the generalized errors at the continuous level allows discovering the new force (related to the detected jammer) present in the surrounding environment and understanding how much it changed the evolution at the continuous level. The generalized errors at this level are based on the difference between the lateral predictive message $\pi(\tilde{X}_t)$ and the hierarchical messages coming from the bottom level that are projected on the discrete space and on the continuous space. As mentioned before (in Section IV-B), the generalized error ($\tilde{\varepsilon}^{[1]}_{\tilde{X}_t}$) projected on the continuous space and associated with (27) is defined as follows:

$$\tilde{\varepsilon}^{[1]}_{\tilde{X}_t} = H^{-1} \tilde{\varepsilon}^{[1]}_{\tilde{Z}_t}. \tag{29}$$

On the other hand, it would be possible to calculate the Generalized Errors $\tilde{\varepsilon}^{[2]}_{\tilde{X}_t}$ (associated with (25)) between the continuous and the observation level by subtracting the posterior from the real measurement that is projected on the discrete level and formulated in the following way:

$$\tilde{\varepsilon}^{[2]}_{\tilde{X}_t} = \begin{cases} \tilde{\mu}\big(\underset{\tilde{S}_t \in \mathbf{S}}{\operatorname{argmax}}\, \lambda(\tilde{S}_t)\big) - \tilde{X}_t & if \;\; \tilde{S}^{\pi}_t = \tilde{S}^{\lambda}_t \\ \tilde{\mu}\big(\underset{\tilde{S}_t \in \mathbf{S}}{\operatorname{argmax}}\, \lambda(\tilde{S}_t)\big) - \tilde{\mu}\big(\underset{\tilde{S}_t \in \mathbf{S}}{\operatorname{argmax}}\, \pi(\tilde{S}_t)\big) & if \;\; \tilde{S}^{\pi}_t \neq \tilde{S}^{\lambda}_t \end{cases} \tag{30}$$

where $\tilde{S}^{\pi}_t = \underset{\tilde{S}_t \in \mathbf{S}}{\operatorname{argmax}}\, \pi(\tilde{S}_t)$ is the expected superstate and $\tilde{S}^{\lambda}_t = \underset{\tilde{S}_t \in \mathbf{S}}{\operatorname{argmax}}\, \lambda(\tilde{S}_t)$ is the observed superstate. The distinction between these errors at the continuous level is that the first ($\tilde{\varepsilon}^{[1]}_{\tilde{X}_t}$) is used by KF to correct the predictions and adapt to the new situation during the testing phase, while the second ($\tilde{\varepsilon}^{[2]}_{\tilde{X}_t}$) is used off-line after finishing the experience to discover the dynamic behaviour of the detected jammer that can be encoded in a new dynamic model.

3) Observation Level: at this level we can calculate two generalized errors as well. The first one is related to the difference between actual measurement and prediction projected on the measurement space (as mentioned before) and defined as:

$$\tilde{\varepsilon}^{[1]}_{\tilde{Z}_t} = \tilde{Z}_t - H\tilde{X}_t. \tag{31}$$

On the other hand, since we know which superstates of the model are affected by the jammer (from the discrete level), calculating the distance from the superstates' centroid allows to extract the source of the cause (jammer) that affected the shift noticed at higher levels. So, $\tilde{\varepsilon}^{[2]}_{\tilde{Z}}$ can be calculated in the following way:

$$\tilde{\varepsilon}^{[2]}_{\tilde{Z}_t} = \tilde{Z}_t - H\tilde{\mu}(\underset{\tilde{S}_t \in \mathbf{S}}{\operatorname{argmax}}\, \lambda(\tilde{S}_t)), \tag{32}$$

which represent the jammer's signal. This can be explained by the fact that the received signal $\tilde{Z}_t$ in an abnormal situation consists of both the normal signal that the UAV is supposed to

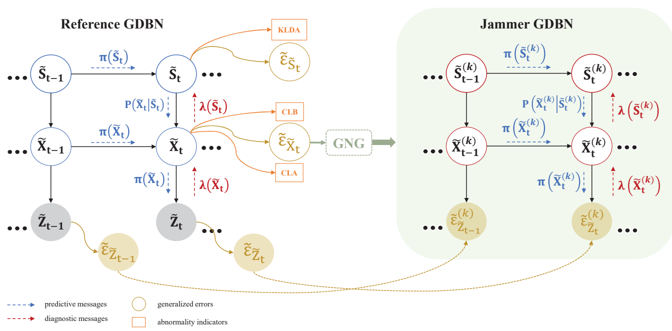


Fig. 6. Schematic that illustrate the process of learning a separated model of the detected jammer under the k-th modulation mode by exploiting the Generalized Errors. Also, it illustrates the relationship between the reference GDBN and the jammer's GDBN under the k-th modulation scheme. The reference GDBN model acts as the generative model that infers the direct cause and deep cause of its own observations and as the process of generating observations for the jammer's model.

receive and the jamming signal. So, subtracting the received jammed signal from its estimated superstate (at the top level) gives the new force signal (i.e. jammer). It is important to recall that estimating the new emergent force is possible since we represented the random hidden variables in generalized coordinates of motion (including the state per se and the corresponding temporal derivative).

E. Extract jammer and learn the corresponding dynamic model

The generalized errors at the continuous level and the observation level can be used to extract the jammer's dynamic rules as well as the jammer's signal, which can be used to learn the corresponding dynamic model by clustering those errors following the same approach seen before (to learn the reference GDBN model) for each jamming signal under the k-th modulation scheme (see Fig. 6). The generalized errors representing the jamming signal under the k-th modulation scheme are clustered using GNG, which provides a set $\mathbf{S}^{(\mathbf{k})}$ of discrete regions as mentioned before. Following the same mechanism we used to learn the reference model, i.e., estimating the transition matrix $\Pi^{(k)}$, time-varying transition matrix $\Pi_{\tau}^{(k)}$ and the statistical properties of each super-state in $\mathbf{S}^{(\mathbf{k})}$, we obtain a set $\mathcal{S}_{\mathcal{M}}$ of jamming dynamic models describing the jammers dynamic behaviours under different modulations, such that:

$$\mathcal{S}_{\mathcal{M}} = \{\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_K\}. \tag{33}$$

However, here we propose to learn additional statistical properties for each $\tilde{S}_m^{(k)} \in \mathcal{M}_k$ (where $\mathcal{M}_k \subset \mathcal{S}_{\mathcal{M}}$), namely, a set $\tilde{\boldsymbol{\mu}}_{\tilde{\mathbf{S}}_{\boldsymbol{m}}^{(\mathbf{k})}}$ of *conditional generalized mean values* defined as:

$$\tilde{\boldsymbol{\mu}}_{\tilde{\mathbf{S}}_{\boldsymbol{m}}^{(\mathbf{k})}} = \left[\tilde{\mu}_{\tilde{S}_m^{(k)}|\tilde{S}_1^{(k)}}, \tilde{\mu}_{\tilde{S}_m^{(k)}|\tilde{S}_2^{(k)}}, \dots, \tilde{\mu}_{\tilde{S}_m^{(k)}|\tilde{S}_M^{(k)}}\right], \tag{34}$$

where the *conditional control vectors* ($\boldsymbol{U}_{\tilde{\mathbf{S}}_{\boldsymbol{m}}^{(\mathbf{k})}}$) are encoded such that:

$$\boldsymbol{U}_{\tilde{\mathbf{S}}_{\boldsymbol{m}}^{(\mathbf{k})}} = \left[U_{\tilde{S}_m^{(k)}|\tilde{S}_1^{(k)}}, U_{\tilde{S}_m^{(k)}|\tilde{S}_2^{(k)}}, \dots, U_{\tilde{S}_m^{(k)}|\tilde{S}_M^{(k)}}\right], \tag{35}$$

and a set $\boldsymbol{\Sigma}_{\tilde{\mathbf{S}}_{\boldsymbol{m}}^{(\mathbf{k})}}$ of *conditional covariance matrices* defined as:

$$\boldsymbol{\Sigma}_{\tilde{\mathbf{S}}_{\boldsymbol{m}}^{(\mathbf{k})}} = \left[\Sigma_{\tilde{S}_m^{(k)}|\tilde{S}_1^{(k)}}, \Sigma_{\tilde{S}_m^{(k)}|\tilde{S}_2^{(k)}}, \dots, \Sigma_{\tilde{S}_m^{(k)}|\tilde{S}_M^{(k)}}\right]. \tag{36}$$

This additional information allows understanding not only the dynamic random changes at the discrete level (through the transition probabilities encoded in the transition matrix) but also to discover and represent the force that generated those changes and the rules by which the signal is shifting among them. This realizes the key to predict the dynamic changes of different modulation modes efficiently.

Algorithm 2: AJC

Input: $\tilde{Z}_t$, N T, Π, Π_τ, $\tilde{S}_m : (\tilde{\mu}_{\tilde{S}_m}, \Sigma_{\tilde{S}_m})\ \forall\ \tilde{S}_m \in \mathbf{S} \leftarrow$ Reference GDBN Model
$\mathcal{N}$, $\Pi^{(k)}$, $\Pi_t^{(k)}$, $\tilde{S}_m^{(k)} : (\tilde{\mu}_{\tilde{S}_m^{(k)}}, \Sigma_{\tilde{S}_m^{(k)}})\ \forall\ \tilde{S}_m^{(k)} \in \mathcal{M}_k$ (where $\mathcal{M}_k \subset \mathcal{S}_{\mathcal{M}}$) $\leftarrow$ Jammers GDBN Models

for *each* $t \in T$ **do**
  Predict normal commands and detect jammer using M-MJPF (Algorithm 1):
  $[KLDA, CLA, CLB, \tilde{\varepsilon}_{\tilde{Z}_t}^{[1]}, \tilde{\varepsilon}_{\tilde{Z}_t}^{[2]}, \tilde{\varepsilon}_{\tilde{X}_t}^{[1]}, \tilde{\varepsilon}_{\tilde{X}_t}^{[2]}, \tilde{\varepsilon}_{\tilde{S}_t}\ ]$ = M-MJPF($\tilde{Z}_t$, N, T, Π, $\tilde{S}_m$, $\tilde{\mu}_{\tilde{S}_m}$, $\Sigma_{\tilde{S}_m}$)
  if *KLDA* $> \psi$ || *CLA* $> \psi \leftarrow$ *Jammer is present* **then**
    Predict jammer dynamics:
    for *each* $k \in \mathcal{S}_{\mathcal{M}} \leftarrow$ *available jammers' models* **do**
      for *each* $n \in \mathcal{N}$ **do**
        Prediction at Discrete Level:
        $W_n = \frac{1}{N} \leftarrow$ particle weight
        if $t = 1 \leftarrow$ *initial state* **then**
          Sample: $\tilde{X}_1^{(k)}$ from initial prior density $P(\tilde{X}_1^{(k)})$
          $\tilde{X}_t^{(k)} = \tilde{X}_1^{(k)} \leftarrow$ current state
          Estimate: $\tilde{S}_{n,t}^{(k)}$ from $P(\tilde{X}_t^{(k)}|\tilde{S}_{n,t}^{(k)})$
        else $\leftarrow$ *remaining states*
          $\tilde{S}_{n,t}^{(k)} \sim \pi_\tau(\tilde{S}_{n,t-1}^{(k)})$
          if $\tilde{S}_{n,t}^{(k)} = \tilde{S}_{n,t-1}^{(k)}$ **then**
            $\tau^{n,k} = \tau^{n,k} + 1 \leftarrow$ time elapsed in $\tilde{S}_{n,t}^{(k)}$
          else
            $\tau^{n,k} = 1$
        Prediction at Continuous Level:
        $U_{\tilde{S}_t^{(k)}} = U_{\tilde{S}_{n,t}^{(k)}|\tilde{S}_{n,t-1}^{(k)}} \leftarrow$ *Conditional Control Vector* using (35)
        $P_{t-1|t-1} = \Sigma_{\tilde{S}_{n,t}^{(k)}|\tilde{S}_{n,t-1}^{(k)}} \leftarrow$ *Conditional Covariance matrix* using (36)
        $\tilde{X}_t^{(k)} = A\tilde{X}_{t-1}^{(k)} + BU_{\tilde{S}_{n,t}^{(k)}} \leftarrow$ state
        $P_{t|t-1} = AP_{t-1|t-1}A^\intercal + \Sigma_{\tilde{w}_t} \leftarrow$ covariance
        Calculate: $\pi(\tilde{X}_t^{(k)})$ using (14)
        Estimate the current observation $\tilde{\varepsilon}_{\tilde{Z}_t}^{[2]}$ **using** (32)**:**
        $\lambda(\tilde{X}_t^{(k)}) = P(\tilde{\varepsilon}_{\tilde{Z}_t}^{[2]}|\tilde{X}_t^{(k)}) \leftarrow$ *diagnostic msg1*
        Calculate: $\lambda(\tilde{S}_t^{(k)})$ using (17) $\leftarrow$ *diagnostic msg2*
        Abnormality measurements:
        Calculate: $\boldsymbol{Abn_k}$ using (37)
        Update Belief in hidden variables:
        $\hat{\tilde{\varepsilon}}_{\tilde{Z}_t}^{[2]} = (\tilde{\varepsilon}_{\tilde{Z}_t}^{[2]} - H\tilde{X}_t^{(k)}) \leftarrow$ Kalman innovation
        $K_t = P_{t|t-1}H^\intercal(HP_{t|t-1}H + \Sigma_{\tilde{\varepsilon}_{\tilde{Z}_t}^{[2]}})^{-1} \leftarrow$ Kalman gain
        $\hat{\tilde{X}}_t = \tilde{X}_t + K_t\hat{\tilde{\varepsilon}}_{\tilde{Z}_t}^{[2]} \leftarrow$ updated state
        $\hat{P}_{t|t} = (1 - K_tHP_{t|t-1}) \leftarrow$ updated covariance
        $W_n = W_n \times \lambda(\tilde{S}_t^{(k)}) \leftarrow$ update particles weight

Output: $\mathcal{S}_{Abn}$

F. On-line Automatic Jamming modulation Classification (AJC)

In order to recognize the correct modulation scheme of the detected jammer (i.e. current observation), the UAV will perform multiple predictions in parallel using the learned and stored models in $\mathcal{S}_{\mathcal{M}}$ during the training process and the corresponding statistical properties (defined in (34), (35) and

(36)). Thus, at each time instant t, we have multiple predictions related to multiple GDBN models, where each model $\mathcal{M}_k$ explains the dynamics of the jammer modulated under the *k-th* modulation scheme (refer to Fig. 7). The UAV can evaluate which of these predictions explain the current radio situation by using the abnormality measurement defined in (27) applied to the jammer model and defined as:

$$Abn_k = -\ln\left(\mathcal{BC}\left(\pi(\tilde{X}_t^{(k)}), \lambda(\tilde{X}_t^{(k)})\right)\right), \tag{37}$$

where

$$\mathcal{BC} = \int \sqrt{\pi(\tilde{X}_t^{(k)})\lambda(\tilde{X}_t^{(k)})} d\tilde{X}_t^{(k)}. \tag{38}$$

A set of abnormalities $\mathcal{S}_{Abn}$ is available at each time instant t, such that:

$$\mathcal{S}_{Abn}(t) = \{Abn_1, Abn_2, \ldots, Abn_K\}. \tag{39}$$

The classifier at the UAV is supposed to recognize correctly the modulation scheme of the received signal from a set ($\mathcal{S}_{mod}$) of candidate modulations denoted by integer values, such that:

$$\mathcal{S}_{mod} = \{1, \ldots, K\}. \tag{40}$$

Then, the modulation classification can be made by comparing

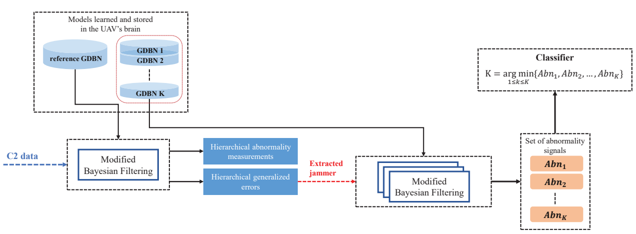


Fig. 7. GDBN-based Jamming Modulation Classification Framework.

between all the abnormality values and selecting the index of the minimum abnormality in the set $\mathcal{S}_{Abn}(t)$ to recognize the modulation scheme, which is given by:

$$\hat{k}(t) = \underset{1 \le k \le K}{\text{argmin}} \{\mathcal{S}_{Abn}\}, \text{where } \hat{k}(t) \in \mathcal{S}_{mod}. \tag{41}$$

The probability of correct classification P_{cc} can be used as a performance metric to evaluate the classification task, and it is expressed as follows:

$$P_{cc} = \frac{1}{T}\sum_{t=1}^{T} P(\hat{k}(t) = k(t)|k(t)), \tag{42}$$

where T is the total testing time and $P(\hat{k}(t) = k(t)|k(t))$ is the probability that the modulation scheme is correctly predicted as $k(t)$ at time (t). The AJC method is summarized in Algorithm 2.

V. Simulation Results and Discussion

A. Simulation setup

The proposed framework for joint detection and classification of multiple jammers is evaluated using simulated data. The UAV trajectory is simulated based on [45]. We study the relationship between the commands and the velocities of the UAV to generate the appropriate bits and consequently generate the LTE signal according to the 3GPP specifications [46] and the important parameters defined in Table I. Similarly, the altered trajectory is extracted from the jammed LTE signal.

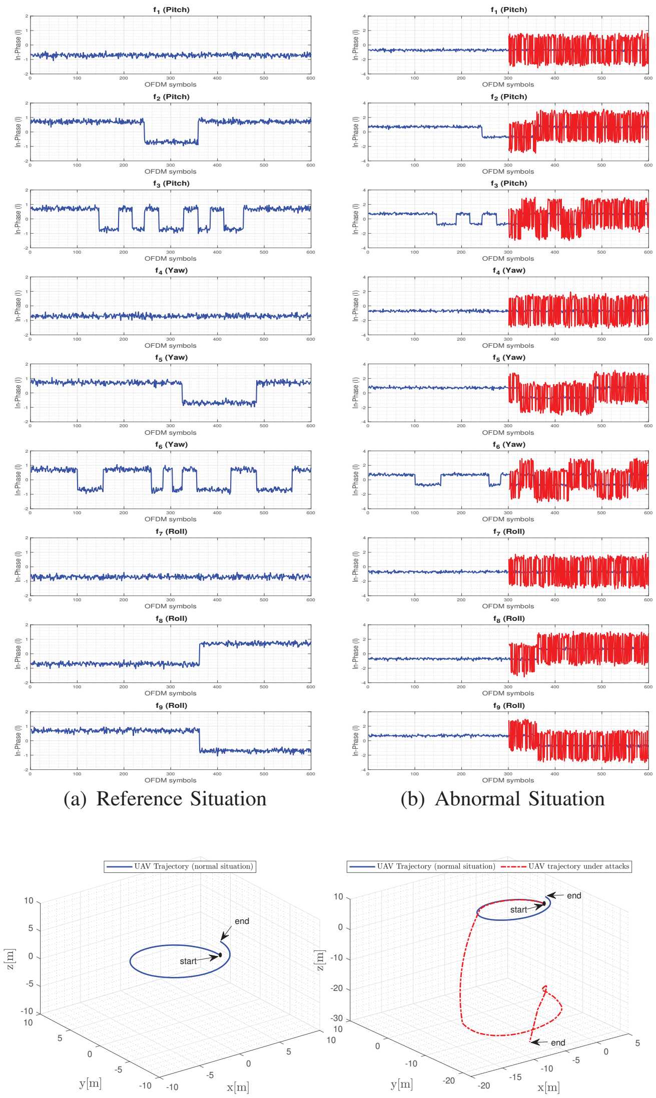
TABLE I
Simulation Parameters

Parameter	Value
BW	1.4 MHz
Duplex mode	FDD
Δf	15 kHz
Number of PRBs per BW	6
Sampling frequency	1.92 MHz
N_{FFT}	128
OFDM symbols per slot	7
CP length	normal
SNR	[-20 dB, ..., +20 dB]
C2 Modulation	QPSK
Jammer Modulation	$\mathcal{S}_{mod}$ = {BPSK, QPSK, 8-PSK, 16-QAM, 32-PSK, 64-QAM, 256-QAM}
Jamming to Signal Power Ratio (JSR)	6 dB
Channel	AWGN
Total Radio Frames	600

The UAV flight time is T_{flight}=*30s* consisting of 600 samples (aka, 600 sets of commands corresponding to 600 OFDM symbols in time domain (Fig. 8-a)). In addition, the UAV extracts the RV from the received PRB every 50 ms, where the RV contains a set of commands transmitted over 9 consecutive sub-carriers in 1 OFDM symbol. Each set of commands will indicate the movement of the UAV in the 3D space.
The output of the digital modulators for both the normal signal and the jammers is normalized based on the average power. The considered situations are:
***(i)* Reference Situation**: representing the normal behaviour (without attacks) of the signal related to the original commands sent by the operator (see Fig. 8-a) which is used to learn the reference GDBN model. The UAV trajectory during this situation is depicted in Fig. 8-c.
***(ii)* Abnormal Situation**: during this situation the jammer uses 2 configurations. The first one (used in Section V-B) is related to the jammer who is attacking continuously all the sub-carriers starting from time (in terms of OFDM symbols) $t = 300$ till $t = 600$ in different radio experiences adopting one modulation scheme from $\mathcal{S}_{mod}$ in each experience. While the second (used in Section V-C) is related to the jammer who is attacking from $t = 1$ till $t = 300$ to evaluate the classification performance after learning the jamming models.

B. Learning Reference Model and jamming Models

Initially, the UAV starts perceiving the radio environment and predicting the environmental states using an initial GDBN model, supposing that the signals' dynamics are static. Such an assumption leads to high abnormalities all the time since the UAV fails to predict the actual states of the signals. Exploiting the Generalized Errors calculated during the abnormal situation (using (6)) allows the UAV to discover the real dynamics by clustering those errors in an unsupervised manner and store them in the reference GDBN model. After that, the UAV equipped with the reference GDBN can accurately predict the future states of the commands at multiple sub-carriers



(a) Reference Situation (b) Abnormal Situation

(c) Reference Situation (d) Abnormal Situation

Fig. 8. The received commands and the corresponding trajectories during different situations. (a) Received commands at multiple sub-carriers during normal situations. (b) An example of the received commands at multiple sub-carriers under jamming attacks (BPSK jammer SNR=14dB). (c) UAV trajectory in a normal situation. (d) UAV trajectory under jamming attacks. Blue and red colors represent the trajectory without and with jammer attacks, respectively.

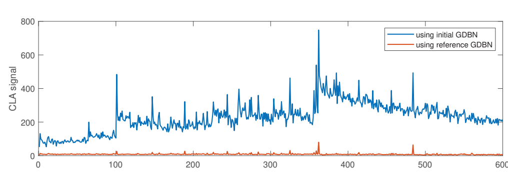
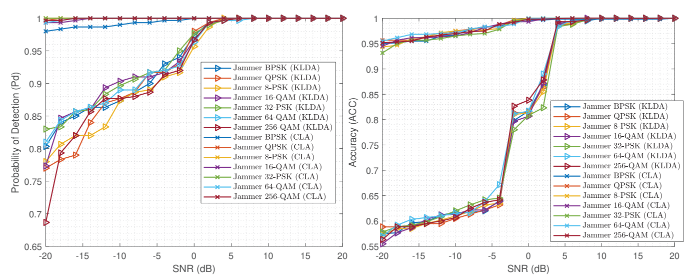


Fig. 9. Abnormality indicator at the continuous levels (defined in (27)) using the initial GDBN model and the reference GDBN model.

without being surprised anymore by the observations. Fig. 9 verifies this where we can observe a high abnormality signal all the time by using the initial GDBN (due to the lack of knowledge about the environmental dynamics) and a quasi-zero abnormality signal by using the reference GDBN that encodes the dynamic rules of the signals allowing by that the UAV to perform correct predictions and so avoid surprising states.

After learning the reference GDBN model when the jammers are absent and by facing a new radio experience, the cognitive-UAV can predict the future commands that it is expecting to receive at multiple sub-carriers and consequently detect any jamming attacks at different hierarchical levels using the abnormality measurements (KLDA and CLA) defined in (21) and (27). We evaluate the detection performance of the proposed approach for multiple jammers with different modulation schemes in different radio conditions by varying the SNR from -20 dB to +20dB as shown in Fig. 10. It can be observed that the cognitive-UAV is capable of detecting the jammer efficiently at the continuous level (through the CLA) with high probability and high accuracy even at very low SNR values regardless of the modulation scheme adopted by the jammer. From the figure, we can also observe that the performance of detecting the jammer at the discrete level (through KLDA) degrades as the SNR decrease, this is due to the fact that the signal dynamics at low SNR become faster and thus the transitions among the discrete variables are speedy which make it difficult to capture the dynamic transitional rules efficiently. However, the advantage of detecting the abnormality at multiple hierarchical levels is that when the performance degrades at the discrete level, we can rely more on the continuous level for better performance.



(a) (b)

Fig. 10. Performance of detecting multiple jammers modulated under different modulation scheme as varying the SNR. (a) Probability of Detection at hierarchical levels. (b) Detection Accuracy at hierarchical levels

We showed that it is possible to extract and estimate the jamming signal after detecting its malicious activities on the ongoing communication by exploiting the generalized errors defined in (32). Fig. 11 shows some examples of the I/Q time domain plot of the extracted jammers under different modulation schemes at sub-carrier f_1 and 10dB SNR. Fig. 12 shows the scatter plots of the extracted jammer and the corresponding ground truth.

The estimated jamming signals in different radio experiences are used to learn separated GDBN models encoding the jamming behaviours under different modulation schemes. After employing the unsupervised method (GNG) to cluster the extracted jammers, we obtain a set of GDBN models forming the set $\mathcal{S}_{\mathcal{M}}$ as defined in (33). In this way, the UAV's brain consists of the reference model that describes what commands the UAV is expecting to receive under normal circumstances and another set of models ($\mathcal{S}_{\mathcal{M}}$) representing the dynamic behaviour of multiple jammers using different modulation

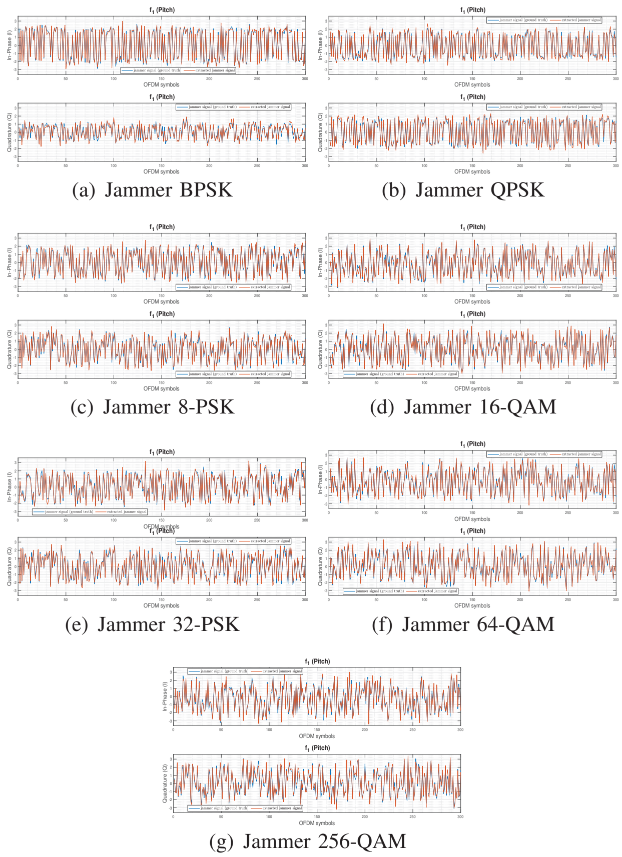


Fig. 11. I/Q time domain plot of the extracted jamming signals (based on $\tilde{\varepsilon}^{[2]}_{\tilde{Z}_t}$ defined in (32)) under different modulation schemes and SNR=10 dB at sub-carrier f_1 and of the ground truth jamming signals.

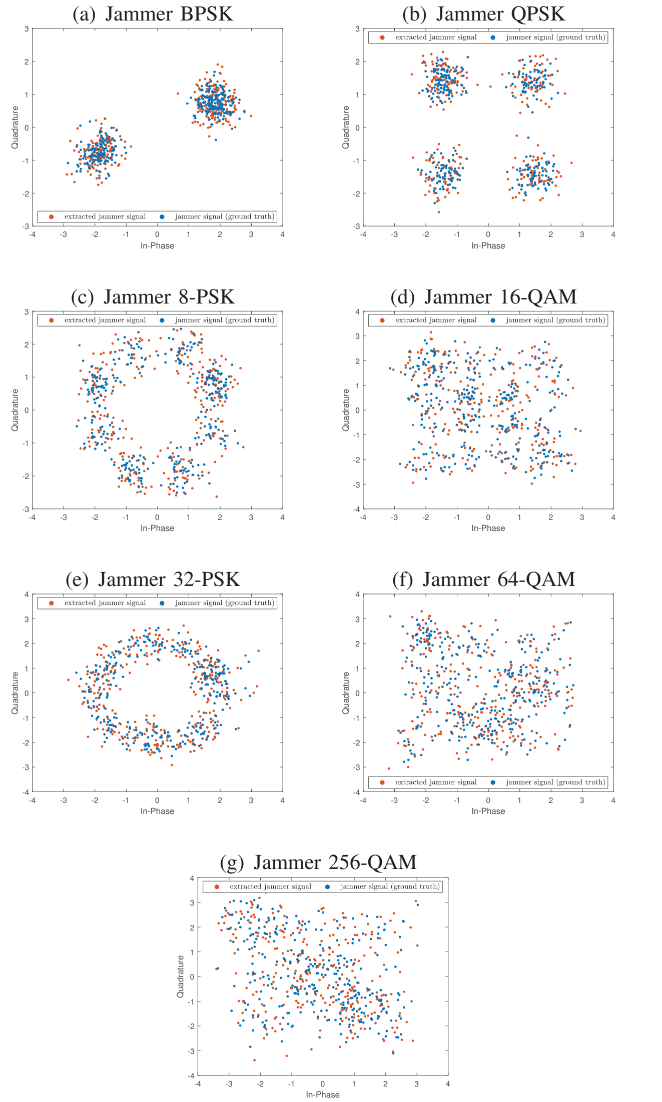


Fig. 12. Scatter plot of the extracted jamming signals (based on $\tilde{\varepsilon}^{[2]}_{\tilde{Z}_t}$ defined in (32)) and the corresponding ground truth at sub-carrier f_1 and 10 dB SNR.

schemes. In this way, the UAV predicts the future commands using the reference model, calculates the abnormality measurements and the generalized errors. If an abnormality is occured, the UAV will perform multiple predictions in parallel and calculates the corresponding abnormality measurements. The UAV compares among the multiple abnormality measurements and pick the index of the minimum one which is associated with the index of the jamming models in the $\mathcal{S}_{\mathcal{M}}$ to recognize the modulation scheme of the detected jammer.

C. Online Classification Process

In Fig. 13, we showed the classification accuracy of the proposed GDBN for each modulation scheme in the candidate set ($\mathcal{S}_{mod}$). We can observe that GDBN achieves high classification accuracy for most of the modulation schemes, especially at $SNR > 5dB$. The low accuracy at low SNRs ($SNR < 0dB$) for the majority of the modulation schemes in $\mathcal{S}_{\mathcal{M}}$ can be explained by the fact that at low SNR the data samples of each modulation are concentrated around the origin (in the complex IQ plane), and the dynamics at low SNR become very fast which makes it difficult to discover and capture these dynamic rules that are encoded in the GDBN model in an efficient way. Some examples of the resultant confusion matrices at various SNR ratios are exhibited in Fig. 14.

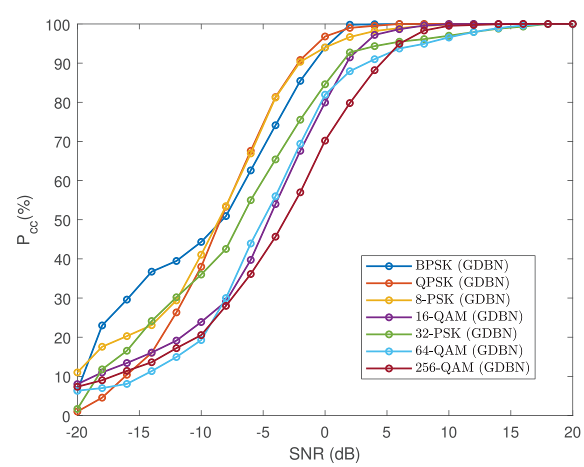


Fig. 13. Performance evaluation of the proposed GDBN-based framework: Probability of correct classification for each modulation scheme versus SNR.

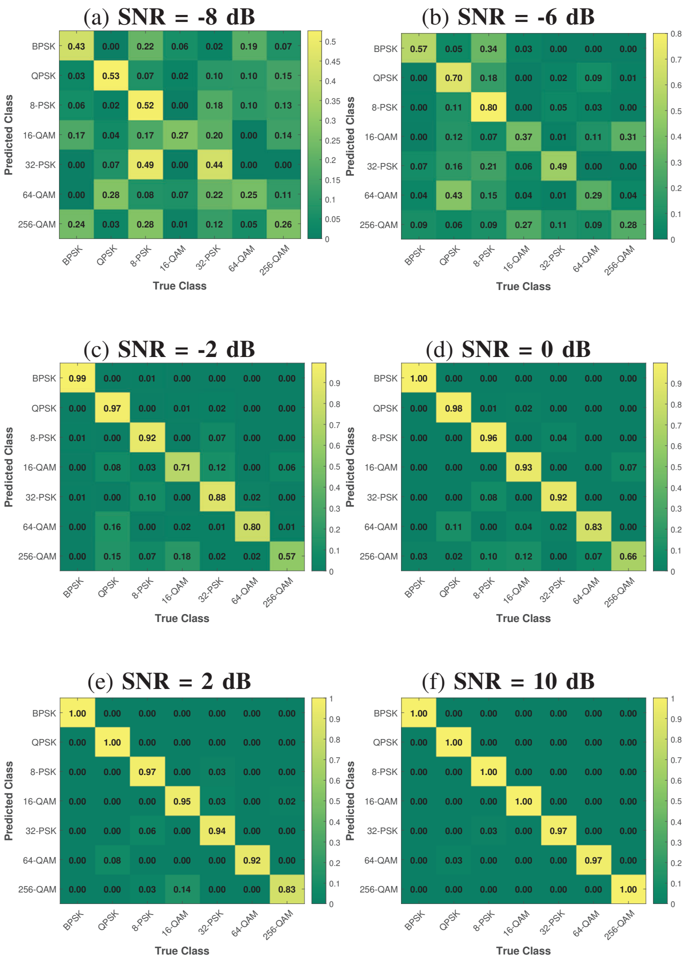


Fig. 14. Confusion matrices for the proposed GDBN method at various SNR values.

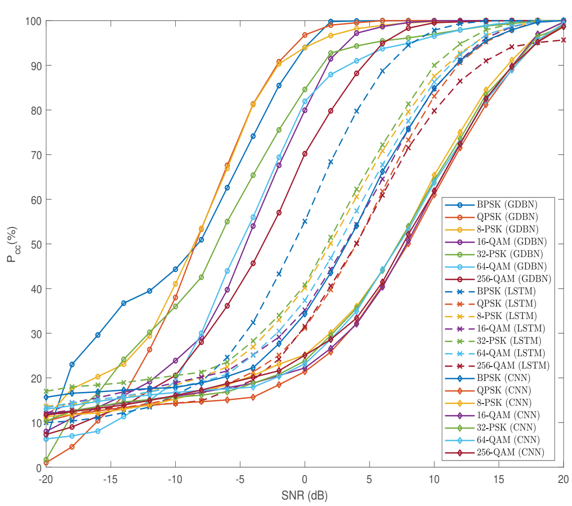


Fig. 15. Performance comparison between the GDBN, LSTM and CNN: Probability of correct classification for each modulation scheme versus SNR.

In addition, we compare the performance of the proposed GDBN with the Convolutional Neural Network (CNN) and the Long Short Term Memory (LSTM). We followed the same approach used to learn the GDBN (thus using the same state vector used as input to the GNG to learn the GDBN)

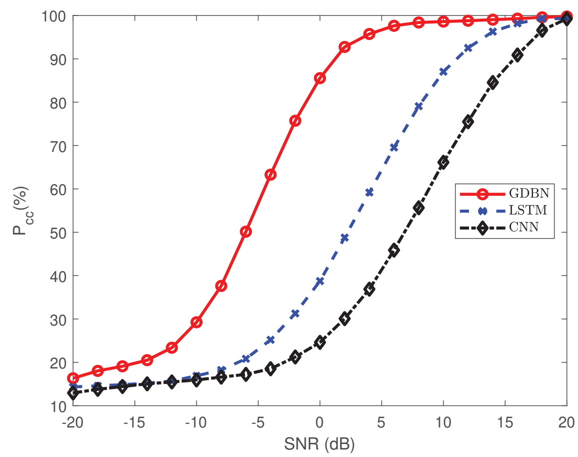


Fig. 16. Performance comparison between the GDBN, LSTM and CNN: Overall probability of correct classification versus SNR.

for both CNN and LSTM for a fair comparison. For CNN, we used the same configuration (i.e. same number of layers) employed in [47], but with different input, here we used a state vector consisting of IQ components and the corresponding derivatives. While the LSTM used here has 3 layers, one LSTM layer, one fully connected layer, and finally, a dense softmax layer that maps the classified features to one of the available modulations schemes in $\mathcal{S}_{mod}$. Fig. 15, shows the performance comparison between the proposed GDBN, LSTM and CNN. It can be seen that the GDBN outperforms the other techniques in the majority of the available modulation schemes. This can be understood better by plotting the overall comparison performance, i.e., the average probability of correct classifications among all the Pcc related to each modulation. The overall comparison is depicted in Fig. 16, and it shows that the proposed GDBN beats LSTM and CNN, especially at $SNR > 5dB$. This means that the proposed approach succeeded to learn the dynamic proprieties (at hierarchical levels) of the signal under a certain modulation scheme, which allows predicting the future behaviour of the signal based on the rules encoded in that model. In addition, LSTM and CNN perform the supervised learning by using the input vector along with the labels of each modulation scheme during the learning process, while in the case of GDBN, we followed an unsupervised approach to learn the model. Also, we have seen that GDBN allows to learn the relationships among the random variables (at hidden layers) in the network explicitly and evaluate the situation using abnormality measurements which can be used as self-information by the radio itself to extract new features and learn emergent rules representing new radio situations incrementally. This is difficult in LSTM and CNN, where the dependencies between the hidden variables at multiple layers are viewed as a black box, so results can not be explained. This limitation impacts the capability of learning by understanding which is crucial in CR to learn continually while observing the environment.

Furthermore, we analyzed the performance of the proposed framework to automatically classify the detected jammers by changing the number of neurons (i.e. the number of superstates

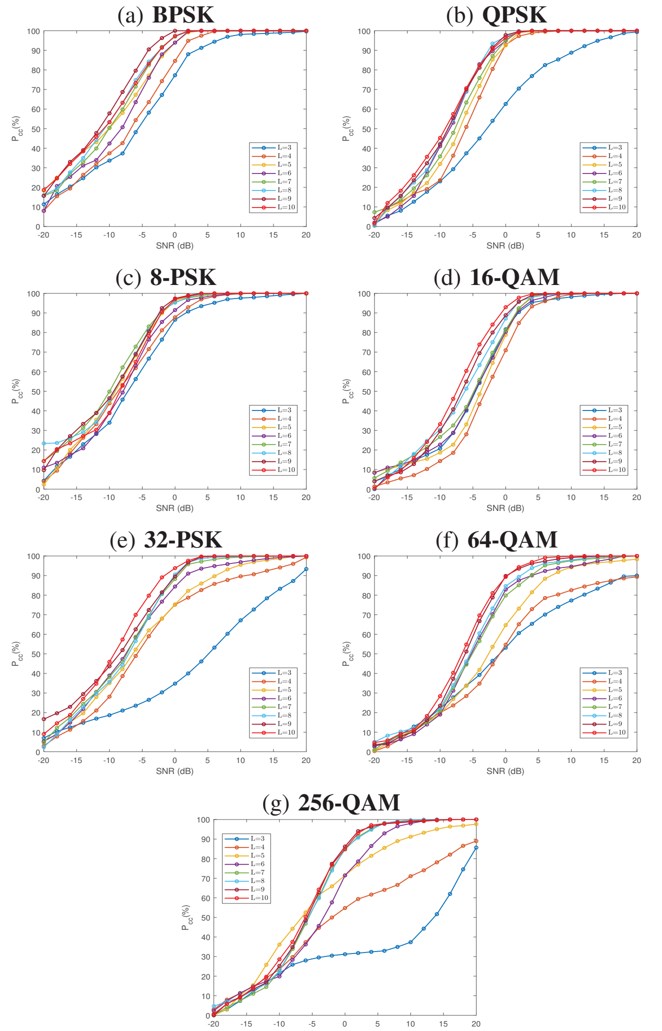


Fig. 17. Performance evaluation in terms of classification accuracy of the proposed GDBN method using different number of neurons (L) to learn the jamming models.

representing the discrete level of the model) used to learn the jamming models. It is to note that in the previous results, we used a fixed number of neurons ($L = 4$) also when we compare with other methods. Considering the influence of the number of neurons on the classification process in addition to the influence of the SNR ratio is of great importance. We applied Bayesian optimization to improve the classification performance by using different L values (related to L models) and finding the model that returns the best classification accuracy (P_{cc}). The performance comparison of the jammer's GDBN models with a different number of neurons is shown in Fig. 17. It is clear that increasing the number of neurons (L) improves the classification accuracy, especially for high order modulations (i.e., 32-PSK, 64-QAM and 256-QAM). This can be explained by the fact that having a high number of constellations can not be represented efficiently by few neurons since it deteriorates the capture of the dynamic transitions of the data samples under the high order modulations. At low SNR ratios, the confusion between different schemes is high due to the high interference caused by the channel, leading to low classification accuracy. Fig. 18 shows the overall probability of correct classification and gives a clear idea of how the performance changes as changing the L parameter.

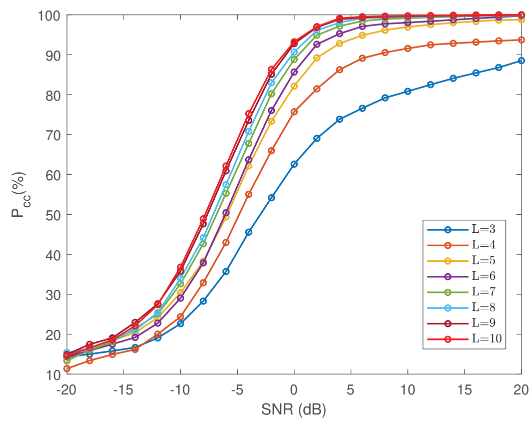


Fig. 18. Performance evaluation in terms of overall classification accuracy of the proposed GDBN method using a different number of neurons (L) to learn the jamming models.

VI. Conclusion

We proposed a novel method for joint detection and classification of jamming attacks in a Cognitive-UAV-based radio application. The method is based on learning a dynamic model representing the radio environment under normal circumstances encoded in a GDBN model. The acquired knowledge encoded in the dynamic model can be used during the online phase to predict what the cognitive-UAV is supposed to receive, evaluate hierarchical abnormality measurements and generalised errors to explain the current situation by differentiating between normal and abnormal situations (i.e., jammer detection) and extract the jamming signal by exploiting the errors to encode it incrementally in a new GDBN model. Thus, in future experiences, if the cognitive-UAV detects a jamming attack, it can perform multiple predictions in parallel using the M-MJPF and selects the best model that explains the current situation to recognize the modulation scheme of the detected jammer. Simulation results showed that the proposed method achieves high probabilities in detecting jammer and high accuracy in classifying them and outperforms LSTM and CNN in classifying multiple jamming signals. In addition, the proposed approach provides interpretable results where multiple abnormality measurements and generalised errors can be used as self-information to keep learning incrementally. Our future objectives include optimizing the proposed approach to achieve high classification accuracy at low SNR and studying the interaction between the cognitive-UAV and multiple jammers to design an optimal resource allocation strategy for anti-jamming.

References

[1] L. Zhang, H. Zhao, S. Hou, Z. Zhao, H. Xu, X. Wu, Q. Wu, and R. Zhang. A Survey on 5G Millimeter Wave Communications for UAV-Assisted Wireless Networks. *IEEE Access*, 7:117460–117504, 2019.

[2] B. Li, Z. Fei, and Y. Zhang. UAV Communications for 5G and Beyond: Recent Advances and Future Trends. *IEEE Internet of Things Journal*, 6(2):2241–2263, 2019.

[3] Zheng Hui Ernest Tan, A.S. Madhukumar, Rajendra Prasad Sirigina, and Anoop Kumar Krishna. Addressing spectrum efficiency through hybrid-duplex UAV communications: Challenges and opportunities. *Vehicular Communications*, 24:100235, 2020.

[4] H. V. Abeywickrama, B. A. Jayawickrama, Y. He, and E. Dutkiewicz. Empirical Power Consumption Model for UAVs. In *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pages 1–5, 2018.

[5] H. V. Abeywickrama, B. A. Jayawickrama, Y. He, and E. Dutkiewicz. Comprehensive energy consumption model for unmanned aerial vehicles, based on empirical studies of battery performance. *IEEE Access*, 6:58383–58394, 2018.

[6] X. Sun, D. W. K. Ng, Z. Ding, Y. Xu, and Z. Zhong. Physical layer security in uav systems: Challenges and opportunities. *IEEE Wireless Communications*, 26(5):40–47, 2019.

[7] N. A. Khalek and W. Hamouda. From Cognitive to Intelligent Secondary Cooperative Networks for the Future Internet: Design, Advances, and Challenges. *IEEE Network*, pages 1–8, 2020.

[8] Z. Yuheng, W. Yan, and X. Yanyong. Optimal Sensing Slots Determination for UAV under Correlated Log-normal Shadowing. In *2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN)*, pages 318–321, 2019.

[9] Z. Xu, P. Ivan, T. Li, and A. Tsourdos. Greedy Based Proactive Spectrum Handoff Scheme for Cognitive Radio Systems. In *2019 IEEE Cognitive Communications for Aerospace Applications Workshop (CCAAW)*, pages 1–5, 2019.

[10] R. Wei, Z. Xu, Q. Zhang, K. Zhou, and T. Ni. Analysis and Application to collision avoidance stability of cognitive UAV. In *2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC)*, pages 1–5, 2018.

[11] N. Tang, H. Tang, B. Li, and X. Yuan. Cognitive NOMA for UAV-Enabled Secure Communications: Joint 3D Trajectory Design and Power Allocation. *IEEE Access*, 8:159965–159978, 2020.

[12] A. Bhowmick, S. D. Roy, and S. Kundu. Throughput of an Energy-Harvesting UAV Assisted Cognitive Radio Network. In *2020 National Conference on Communications (NCC)*, pages 1–6, 2020.

[13] A. Chriki, H. Touati, H. Snoussi, and F. Kamoun. Centralized Cognitive Radio Based Frequency Allocation for UAVs Communication. In *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*, pages 1674–1679, 2019.

[14] Y. Pan, X. Da, H. Hu, Z. Zhu, R. Xu, and L. Ni. Energy-Efficiency Optimization of UAV-Based Cognitive Radio System. *IEEE Access*, 7:155381–155391, 2019.

[15] H. Hu, X. Da, Y. Huang, H. Zhang, L. Ni, and Y. Pan. SE and EE Optimization for Cognitive UAV Network Based on Location Information. *IEEE Access*, 7:162115–162126, 2019.

[16] W. Mei and R. Zhang. UAV-Sensing-Assisted Cellular Interference Coordination: A Cognitive Radio Approach. *IEEE Wireless Communications Letters*, 9(6):799–803, 2020.

[17] A. M. Almasoud and A. E. Kamal. Data Dissemination in IoT Using a Cognitive UAV. *IEEE Transactions on Cognitive Communications and Networking*, 5(4):849–862, 2019.

[18] Y. Wang, L. Chen, Y. Zhou, X. Liu, F. Zhou, and N. Al-Dhahir. Resource Allocation and Trajectory Design in UAV-Assisted Jamming Wideband Cognitive Radio Networks. *IEEE Transactions on Cognitive Communications and Networking*, pages 1–1, 2020.

[19] H. T. Nguyen, H. D. Tuan, T. Q. Duong, H. V. Poor, and W. J. Hwang. Joint D2D Assignment, Bandwidth and Power Allocation in Cognitive UAV-Enabled Networks. *IEEE Transactions on Cognitive Communications and Networking*, 6(3):1084–1095, 2020.

[20] W. Khalid and H. Yu. Residual Energy Analysis with Physical-Layer Security for Energy-Constrained UAV Cognitive Radio Systems. In *2020 International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–3, 2020.

[21] Q. Wu, W. Mei, and R. Zhang. Safeguarding Wireless Network with UAVs: A Physical Layer Security Perspective. *IEEE Wireless Communications*, 26(5):12–18, 2019.

[22] Ruben Ferre, Alberto Fuente, and Elena Simona Lohan. Jammer Classification in GNSS Bands Via Machine Learning Algorithms. *Sensors*, 19:4841, 11 2019.

[23] A. Toma, A. Krayani, M. Farrukh, H. Qi, L. Marcenaro, Y. Gao, and C. S. Regazzoni. AI-Based Abnormality Detection at the PHY-Layer of Cognitive Radio by Learning Generative Models. *IEEE Transactions on Cognitive Communications and Networking*, 6(1):21–34, 2020.

[24] A. Krayani, M. Baydoun, L. Marcenaro, Y. Gao, and C. S. Regazzoni. Smart Jammer Detection for Self-Aware Cognitive UAV Radios. In *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–7, 2020.

[25] A. Krayani, M. Baydoun, L. Marcenaro, A. S. Alam, and C. Regazzoni. Self-Learning Bayesian Generative Models for Jammer Detection in Cognitive-UAV-Radios. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pages 1–7, 2020.

[26] S. Peng, H. Jiang, H. Wang, H. Alwageed, Y. Zhou, M. M. Sebdani, and Y. Yao. Modulation Classification Based on Signal Constellation Diagrams and Deep Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(3):718–727, 2019.

[27] Y. Liu, O. Simeone, A. M. Haimovich, and W. Su. Modulation Classification for MIMO-OFDM Signals via Approximate Bayesian Inference. *IEEE Transactions on Vehicular Technology*, 66(1):268–281, 2017.

[28] L. Huang, W. Pan, Y. Zhang, L. Qian, N. Gao, and Y. Wu. Data Augmentation for Deep Learning-Based Radio Modulation Classification. *IEEE Access*, 8:1498–1506, 2020.

[29] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su. Survey of automatic modulation classification techniques: classical approaches and new trends. *IET Communications*, 1(2):137–156, 2007.

[30] J. Zheng and Y. Lv. Likelihood-Based Automatic Modulation Classification in OFDM With Index Modulation. *IEEE Transactions on Vehicular Technology*, 67(9):8192–8204, 2018.

[31] F. Meng, P. Chen, L. Wu, and X. Wang. Automatic Modulation Classification: A Deep Learning Enabled Approach. *IEEE Transactions on Vehicular Technology*, 67(11):10760–10772, 2018.

[32] A. P. Hermawan, R. R. Ginanjar, D. Kim, and J. Lee. CNN-Based Automatic Modulation Classification for Beyond 5G Communications. *IEEE Communications Letters*, 24(5):1038–1041, 2020.

[33] S. Huang, R. Dai, J. Huang, Y. Yao, Y. Gao, F. Ning, and Z. Feng. Automatic Modulation Classification Using Gated Recurrent Residual Network. *IEEE Internet of Things Journal*, 7(8):7795–7807, 2020.

[34] L. Huang, Y. Zhang, W. Pan, J. Chen, L. P. Qian, and Y. Wu. Visualizing Deep Learning-based Radio Modulation Classifier. *IEEE Transactions on Cognitive Communications and Networking*, pages 1–1, 2020.

[35] S. Huang, L. Chai, Z. Li, D. Zhang, Y. Yao, Y. Zhang, and Z. Feng. Automatic Modulation Classification Using Compressive Convolutional Neural Network. *IEEE Access*, 7:79636–79643, 2019.

[36] M. Zhang, M. Diao, and L. Guo. Convolutional Neural Networks for Automatic Cognitive Radio Waveform Recognition. *IEEE Access*, 5:11074–11082, 2017.

[37] J. H. Lee, K. Kim, and Y. Shin. Feature Image-Based Automatic Modulation Classification Method Using CNN Algorithm. In *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 1–4, 2019.

[38] B. Tang, Y. Tu, Z. Zhang, and Y. Lin. Digital Signal Modulation Classification With Data Augmentation Using Generative Adversarial Nets in Cognitive Radio Networks. *IEEE Access*, 6:15713–15722, 2018.

[39] Y. Li, G. Shao, and B. Wang. Automatic Modulation Classification Based on Bispectrum and CNN. In *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, pages 311–316, 2019.

[40] R. Li, L. Li, S. Yang, and S. Li. Robust Automated VHF Modulation Recognition Based on Deep Convolutional Neural Networks. *IEEE Communications Letters*, 22(5):946–949, 2018.

[41] 3GPP TR 36.777. Technical specification group radio access network: study on enhanced LTE support for aerial vehicles. V15.0.0, Dec., 2017.

[42] H. C. Nguyen, R. Amorim, J. Wigard, I. Z. KováCs, T. B. Sørensen, and P. E. Mogensen. How to Ensure Reliable Connectivity for Aerial Vehicles Over Cellular Networks. *IEEE Access*, 6:12304–12317, 2018.

[43] J. Stanczak, D. Kozioł, I. Z. Kovács, J. Wigard, M. Wimmer, and R. Amorim. Enhanced Unmanned Aerial Vehicle Communication Support in LTE-Advanced. In *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, pages 1–6, Oct 2018.

[44] M. Baydoun, D. Campo, V. Sanguineti, L. Marcenaro, A. Cavallaro, and C. Regazzoni. Learning Switching Models for Abnormality Detection for Autonomous Driving. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 2606–2613, 2018.

[45] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar. The GRASP Multiple Micro-UAV Testbed. *IEEE Robotics Automation Magazine*, 17(3):56–65, Sep. 2010.

[46] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); LTE physical layer; General description (Release 14). Technical Specification (TS) 36.201, 3rd Generation Partnership Project (3GPP), 2017.

[47] T. J. O'Shea, T. Roy, and T. C. Clancy. Over-the-Air Deep Learning Based Radio Signal Classification. *IEEE Journal of Selected Topics in Signal Processing*, 12(1):168–179, 2018.