Chapter 5

Signal Identification and Automatic Modulation Classification

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 interference management. This chapter extends the SA module presented in Chapter 4 by proposing 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 Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN) and Stacked Autoencoder (SAE) 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. Furthermore, this chapter introduces for the first time the automatic modulation conversion which is a prospective candidate technology in the future wireless communications. The proposed framework is based data-driven approach following the inherent intelligent capability of generalized filtering integrated with transport planning.

5.1 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 [32]. 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 [111]. However, UAV-based communications will face several problems such as spectrum scarcity due to the explosively increasing number of connected UAVs [259], energy-efficiency due to the on-board limited battery lifetime [260, 261] 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 [262]. 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 [263]. 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 [264, 265], collision avoidance stability [266], trajectory optimization [267], energy harvesting [268], spectrum scarcity [269], energy efficiency optimization [270, 271], interference coordination [272], data dissemination [273], joint sub-carrier and power allocation [274, 275], and for secure communications [276]. Moreover, studies on UAV-aided networks supported by machine learning has been investigated in [277, 278].

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 [279]. In addition, smart jammers equipped with cognitive capabilities can pose more security threats [185]. 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 chapter 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 [280].

In the previous chapter, 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. Studies from neuroscience have shown that the brain can efficiently use sensory information to resolve uncertainty about its computations and the surrounding world by representing sensory signals *probabilistically* in the form of probability distributions [281]. Inspired by such a brain functionality (known as the 'Bayesian Brain Hypothesis'), we propose to equip CR with a *probabilistic* Generative dynamic model, such as Dynamic Bayesian Networks (DBNs) using Generalized variables, i.e., Generalized DBN (GDBN), encoding knowledge about the radio itself and the structural regularities from its external milieu variations via sensory signals. GDBNs describe 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. Since CR operates in stochastic wireless environments under uncertainty, using a powerful statistical tool as Bayesian filtering is fundamental to dealing with uncertainty and performing inference and estimation of environmental states efficiently. The motivation of studying Bayesian filtering for environmental state estimation is that it is optimal in a conceptual sense [282]. Bayesian Filtering employed on GDBNs allows the radio to evaluate the situation through different abnormality measurements at multiple hierarchical levels and understand if the situation is normal or abnormal (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 chapter, 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 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).

Besides, 6G is expected to have a high-level hierarchy involving low bit-rate IoT devices to Gigabit rate connectivity (e.g., backhaul links). In such a cooperative communication scenario, intermediate or relay nodes are deployed to assist the source and the intended destination (i.e., a Macro Base Station MBS). In this case, a Self-Aware relay node (SAN) receiving signals with low-order modulation from IoT sensors can convert the modulation format of received signals into a higher-order format. The SAN then redirects the transported signals (with higher order modulation by performing modulation conversion) over

the relay backhaul link between SAN and MBS, thus improving transmission capacity and spectral efficiency. Therefore, modulation conversions, which find its origin in optical communications [283], can be a prospective candidate technology in future generation wireless communications. In addition, automatic modulation classification is a possible application where modulation format conversion can play a significant role, providing an explainable approach to identifying the received signal and the type of information it carries. Wherefore, this chapter introduces for the first time the automatic modulation conversion in wireless communications which allows an AI-enabled node to predict signals' dynamics of different modulation schemes and explain how it can be transported (converted) with the minimal effort and forwarded with higher spectral efficiency. To achieve this goal, we propose to integrate the Generalized Filtering framework by Transport Planning to learn the way of converting low-order modulations to high-order modulations, which has also been validated by performing the automatic modulation classification. Simulation results demonstrate the effective performance of our novel framework on converting and classifying multiple modulation formats.

The main contributions of this chapter 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, CNN and SAE; *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; ; *v)* we show how the radio is capable of predicting the dynamics of different modulation formats and of explaining how the dynamic rules evolve by transporting low-order into high-order modulations using the acquired transport maps.

5.2 Related Work

Radio Signal Classification is an important task in many communications systems [284]. 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 [285, 286]. Traditional approaches for modulation classification include Likelihood-Based (LB) approach and Feature-Based (FB) approach [287–289]. 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 sub-optimal, however, a proper design allows achieving optimal performance.

Deep learning-based methods for AMC are extensively investigated in the literature. In [286], 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 [290] 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 [291] 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 [290], and [291] 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 [292] 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 [293], 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 [294], Feature Point (FP) images [295], Contour Stellar Image (which gets more color feature compared to the Constellation Diagram) [296], amplitude spectrums of bispectrum (ASB) images [297], cyclic spectrum images [298]. 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 chapter, we propose a novel GDBN-based method for AMC that can overcome the drawbacks of other existing methods in the literature discussed previously since *i)* it does not require involving data augmentation (as in [286]) to create bigger datasets that might increase latency. *ii)* It follows a data-driven unsupervised approach by allowing the radio to build up knowledge about the radio spectrum from null memory (without using explicit labels of the input signals as done in [290, 291]). Hence, radio's autobiographical memories are grown up incrementally by observing real-time data and learning autonomously from the extracted generalized errors. *iii)* Deep learning approaches ([286, 290–292]) where the hidden variables are at a sub-symbolic level considered as a black box cannot provide a high level of explainability of their decisions, creating results that are hard to understand. The proposed approach has a higher 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 higher 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. *iv)* It relies on raw In-Phase (I) and Quadrature (Q) components of radio signals that are easy to extract (unlike [293–297] that convert radio signals into images) and offers flexibility in implementing the proposed framework in different systems and environments.

5.3 System Model

The system model depicted in Fig. 5.1 extends that illustrated in Fig. 4.3 and 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. Signals of all jammers and the UAV operator are generated according to the propagation model that is shown in Fig. 4.4. The positions of BS, UAV and jammers are denoted by $q^g = [x^g, y^g, z^g]$, $q^{j_k} = [x^{j_k}, y^{j_k}, z^{j_k}]$ and $q^u_t = [x^u_t, y^u_t, z^u_t]$, respectively. The UAV's position q^u_t varies with time while q^g and q^{j_k} are fixed positions where $j_k \in K$ represents the k-th jammer adopting the k-th modulation scheme and K denotes the number of candidate modulation schemes. The line-of-sight (LOS) channels of the UAV communication links become more dominant than other channel impairments, such as small scale fading and shadowing, if the altitude of the UAV is much higher than that of the terrestrial users or BS [299]. We adopt the 3GPP Rural-macro with aerial vehicles (RMa-AV) scenario under LOS conditions [254]. In addition, the doppler frequency shift caused by the UAV mobility is assumed to be compensated at the receiver as in [299]. Thus, the path loss model ($PL^{bu}_{t|dB}$) for the LOS RMa-AV from the BS to the UAV at a given time instant is described as:

$$PL^{bu}_{t|dB} = 20\log_{10}\left(\frac{40\pi d_{bu,t} f_c}{3}\right) + \min(0.03\kappa, 10)\log_{10}(d_{bu,t}) + \min(0.044\kappa, 14.77) + 0.002\log_{10}(\kappa) d_{bu,t}, \quad (5.1)$$

where κ represents the average building height gain and $d_{bu,t}$ is the 3D between BS and the UAV at time slot t defined as:

$$d_{bu,t} = \sqrt{(x^g - x^u_t)^2 + (y^g - y^u_t)^2 + (z^g - z^u_t)^2}. \quad (5.2)$$

The path loss model defined in (5.1) can be transformed into the linear domain according to:

$$PL^{bu}_t = 10^{\frac{PL^{bu}_{t|dB}}{10}}. \quad (5.3)$$

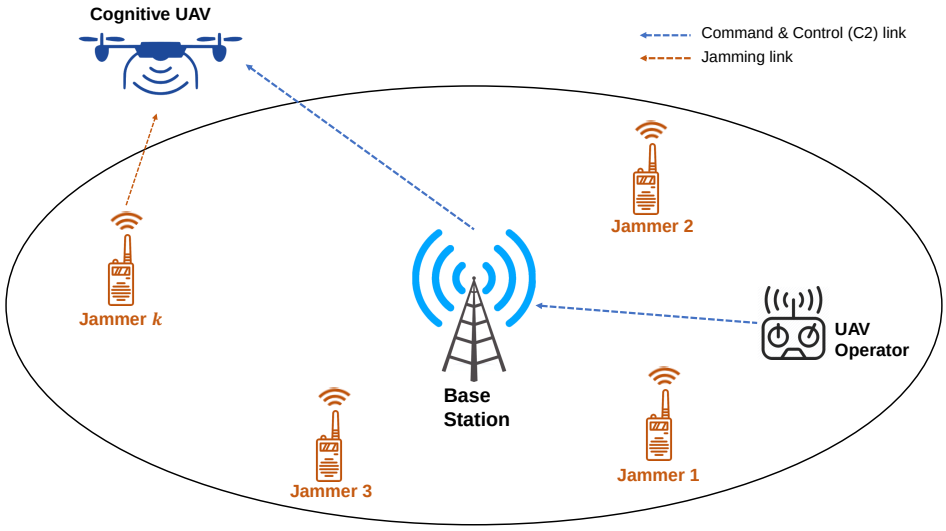


Fig. 5.1 Illustration of the system model.

Likewise, the path loss model ($PL_{t|dB}^{j_k u}$) from the k-th jammer and the UAV is expressed as:

$$PL_{t|dB}^{j_k u} = 20\log_{10}\left(\frac{40\pi d_{j_k u,t} f_c}{3}\right) + \min(0.03\kappa, 10)\log_{10}(d_{j_k u,t}) + \min(0.044\kappa, 14.77) + 0.002\log_{10}(\kappa) d_{j_k u,t}, \quad (5.4)$$

where $d_{j_k u,t}$ is the 3D between the k-th jammer and the UAV at time slot t defined as:

$$d_{j_k u,t} = \sqrt{(x^{j_k} - x_t^u)^2 + (y^{j_k} - y_t^u)^2 + (z^{j_k} - z_t^u)^2}. \quad (5.5)$$

The path loss model defined in (5.4) can be transformed into the linear domain according to:

$$PL_t^{j_k u} = 10^{\frac{PL_{t|dB}^{j_k u}}{10}}. \quad (5.6)$$

Accordingly, the channel gain (h_t^{bu}) from the BS to the UAV at a given time instant is described as:

$$h_t^{bu} = \frac{1}{PL_t^{bu}}, \quad (5.7)$$

and the channel gain ($h_t^{j_k u}$) from the k-th jammer to the UAV is:

$$h_t^{j_k u} = \frac{1}{PL_t^{j_k u}}. \quad (5.8)$$

The UAV receives a PRB each 50ms where the Pitch, Yaw, and Roll commands are trans-

mitted over 9 consecutive sub-carriers (frequency domain) within 1 OFDM symbol (time domain) and it 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 as shown in Fig. 4.5 and Fig. 4.6. The UAV extracts the commands' features (i.e., IQ components) after the OFDM receiver block by exploiting the FFT. At this level (i.e. the output of FFT), all the Resource Elements (REs) in the time-frequency grid can be scanned without any extra hardware or computation by reusing the hardware of FFT cores [300]. In our study, we considered the REs carrying commands (i.e., RV in Fig. 4.6) solely because we aim to analyze the command signals only. However, this can be simply extended in future investigations to consider the whole PRB.

5.4 Problem Formulation

At each instant t, the cognitive-UAV will receive a set of commands (i.e., C2 data) and move accordingly. Our objective is to give the cognitive-UAV the capability to self-evaluate the received commands by identifying if the commands are corrupted by the jammer (i.e., jammer detection) and, consequently, learn jammer dynamics to predict its behaviour in the future and recognize the modulation scheme of the detected jammer (i.e., jammer classification).

The jammer detection task can be formulated in terms of a binary hypothesis testing problem given as:

$$\begin{cases} \mathcal{H}_0 : \mathrm{z}_t = h_t^{bu}\mathrm{x}_t + \mathrm{n}_t, \\ \mathcal{H}_1 : \mathrm{z}_t = h_t^{bu}\mathrm{x}_t + h_t^{j_k u}\mathrm{x}_t^j + \mathrm{n}_t, \end{cases} \tag{5.9}$$

where hypothesis $\mathcal{H}_0$ and $\mathcal{H}_1$ indicate the absence and presence of the jammer, respectively. z_t is the received C2 signal, x_t is the desired signal, x_t^j is the jamming signal and n_t is an additive white Gaussian noise. The proposed approach is based on learning a reference dynamic model ($\mathcal{M}_0$) explaining the normal situation (without jamming attacks) under hypothesis $\mathcal{H}_0$. During testing, cognitive-UAV performs predictions x_t^* conditioned on the learned model $\mathcal{M}_0$ and characterized by the posterior $\mathrm{P}(\mathrm{x}_t^*, \mathcal{M}_0|\mathrm{z}_t)$. Jammer can be detected by comparing the similarity between predictions and likelihood ($\mathrm{P}(\mathrm{z}_t|\mathrm{x}_t^*)$) using a probabilistic distance $\mathcal{D}$ (i.e., abnormality) between $\mathrm{P}(\mathrm{x}_t^*, \mathcal{M}_0|\mathrm{z}_t)$ and $\mathrm{P}(\mathrm{z}_t|\mathrm{x}_t^*)$.

On the other hand, the jamming modulation classification task can be formulated as a classification problem with K modulation schemes. The aim of the classifier is to identify x_t^j of the received signal z_t from a learned set $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_K\}$ of K dynamic models representing K possible modulation schemes and give out $\mathrm{P}(\mathrm{x}_t^j \in \mathcal{M}_k|\mathrm{z}_t)$ where $\mathcal{M}_k \in \mathcal{M}$. Since, all modulation schemes are equally likely, then the optimal classifier is the maximum-

log-likelihood classifier finding the maximum among K conditional probabilities $\mathrm{P}(\mathrm{z}_t|\mathrm{x}_t,\mathcal{M}_k)$, according to:

$$\hat{k} = \underset{\mathcal{M}_k}{\operatorname{argmax}} \log \mathrm{P}(\mathrm{z}_t|\mathrm{x}_t,\mathcal{M}_k), \tag{5.10}$$

where $k = 1,2,\ldots,K$. It is to note that (5.10) is equivalent to finding the minimum among K probabilistic distances $\mathcal{D}\left[\mathrm{P}(\mathrm{z}_t|\mathrm{x}_t^{j*}),\mathrm{P}(\mathrm{x}_t^{j*},\mathcal{M}_k|\mathrm{z}_t)\right]$, according to:

$$\hat{k} = \underset{\mathcal{M}_k}{\operatorname{argmin}} \mathcal{D}\left[\mathrm{P}(\mathrm{z}_t|\mathrm{x}_t^{j*}),\mathrm{P}(\mathrm{x}_t^{j*},\mathcal{M}_k|\mathrm{z}_t)\right]. \tag{5.11}$$

where x_t^{j*} is the predicted jammer signal based on model $\mathcal{M}_k$. Thus, by using a similar approach described by Friston for Bayesian filtering in [301], it can be shown that each likelihood using a different model can be approximated by searching the minimum free energy configuration as an upper-bound estimate. Finding the maximum upper bound is equivalent to obtaining the highest value of (5.10). The free energy can be expressed as the negative of the distance in (5.11). So, obtaining the model that minimizes (5.11) is an approximation to (5.10).

5.5 Proposed Automatic Jamming Modulation Classification

5.5.1 Radio Environment Representation

In our approach, we use a generalized state-space model [1] 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:

$$\tilde{Z}_t = H\tilde{X}_t + \tilde{v}_t, \tag{5.12}$$

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

[1] Random variables involved in the Generalized state-space model are represented in terms of Generalized coordinates of motion. The latter consists of the variable per se and its n-th order temporal derivatives as proposed by Karl Friston [301]. In this work, we consider only generalized motion up to order 1 (i.e., the 1^{st}-order derivatives), and we refer to random variables (discrete and continuous) in generalized coordinates as generalized superstates (discrete) and generalized states (continuous).

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 $\tilde{X}_{t-1}$ by the control vector (or force) $U_{\tilde{S}_t}$ and formulated as follows:

$$\tilde{X}_t = A\tilde{X}_{t-1} + BU_{\tilde{S}_t} + \tilde{w}_t, \tag{5.13}$$

where $A \in \mathbb{R}^{d\times d}$ and $B \in \mathbb{R}^{d\times d}$ are the dynamic model and control model matrices, respectively. $U_{\tilde{S}_t}$ is associated with the generalized superstate $\tilde{S}_t$ where $\tilde{X}_t$ is expected to be found 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 ($\tilde{S}_t$) consisting of the superstate ($\tilde{S}_t$) and the corresponding event are the deep hidden causes generating observations ($\tilde{Z}_t$) and direct hidden causes affecting generalized states ($\tilde{X}_t$). Superstates are the neurons (or clusters) obtained after employing an unsupervised clustering method on the input signals (discussed in the following section). Those neurons represent discrete regions of the physical signal where each neuron contains a set of homogeneous IQ data samples (i.e., samples with very close characteristics). So, the radio environment evolves probabilistically, occupying a finite set of discrete states. Each generalized superstate ($\tilde{S}_t$) is assumed to follow a Gaussian distribution, and so it can be represented by its sufficient statistics, namely, generalized mean (i.e., mean on the IQ samples and mean on the IQ derivatives) and covariance. We know that continuous data samples (i.e., generalized states) inside each superstate evolve according to the same dynamic rule, i.e., the control vector $U_{\tilde{S}_t}$, that guides the evolution of generalized states at the lower level. This motivates the choice of using a linear approximation in (5.13) to model the dynamic evolution of the generalized states (describing the physical signals' dynamics).
At a high abstraction level (discrete level), the evolution of generalized superstates can be expressed in the following form:

$$\tilde{S}_t = f(\tilde{S}_{t-1}) + \tilde{w}_t, \tag{5.14}$$

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.

5.5.2 Learning Stage

GDBN encodes conditional dependencies explicitly among random variables at multiple levels, allowing the radio to understand the cause-effect relationships, which endows the radio to explain predictions, extract errors, and adapt to dynamic environmental changes. In

addition, GDBN provides a probabilistic inference that is computationally efficient (i.e., they can specify dependencies only when necessary, leading to a significant reduction in the cost of inference). Moreover, due to its capability to provide discriminative property to assess received signal characteristics, GDBN can be used as a joint generative and discriminative model. Motivated by the above discussion, 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.

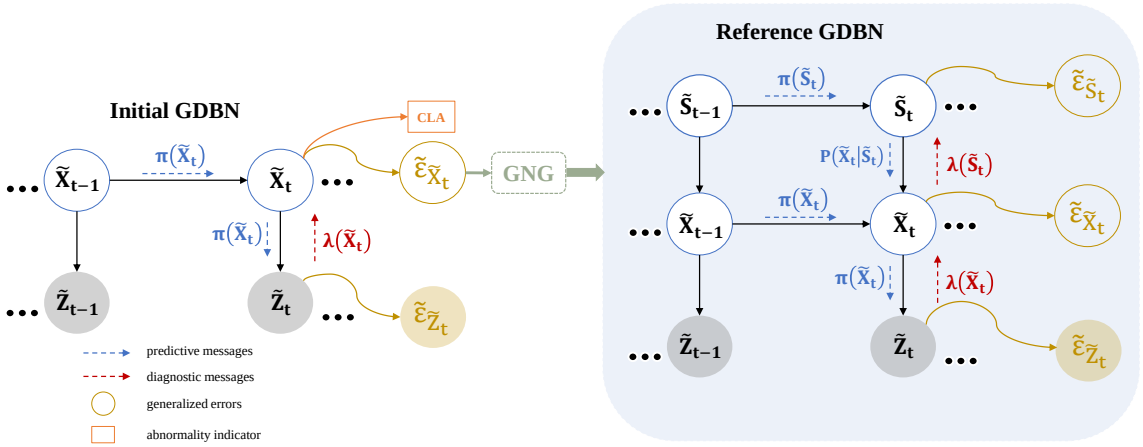


Fig. 5.2 Schematic that illustrates the process of learning incrementally by exploiting the generalized errors. The cognitive-UAV starts perceiving the environment using an initial GDBN model with a static assumption about the signal's evolution. While the cognitive-UAV is predicting and sensing the environment, it can calculate the generalized errors ($\tilde{\varepsilon}_{\tilde{X}_t}$) 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:

$$\tilde{X}_t = A\tilde{X}_{t-1} + \tilde{w}_t, \tag{5.15}$$

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 ($\tilde{\varepsilon}^{[1]}_{\tilde{Z}_t}$) which are the differences between predictions and observations and it is expressed as:

$$\tilde{\varepsilon}^{[1]}_{\tilde{Z}_t} = \tilde{Z}_t - H\tilde{X}_t. \tag{5.16}$$

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:

$$\tilde{\varepsilon}^{[1]}_{\tilde{X}_t} = H^{-1}\tilde{\varepsilon}^{[1]}_{\tilde{Z}_t} = H^{-1}(\tilde{Z}_t - H\tilde{X}_t) = H^{-1}\tilde{Z}_t - \tilde{X}_t. \tag{5.17}$$

The generalized errors ($\tilde{\varepsilon}^{[1]}_{\tilde{X}_t}$) 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. 5.2). GNG encodes the generalized errors into discrete regions described by a set of neurons or superstates $\mathbf{S}$, such that:

$$\mathbf{S} = \{S_1, S_2, \ldots, S_M\}, \tag{5.18}$$

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:

$$\Pi = \begin{bmatrix} \pi_{11} & \pi_{12} & \dots & \pi_{1M} \\ \pi_{21} & \pi_{22} & \dots & \pi_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{M1} & \pi_{M2} & \dots & \pi_{MM} \end{bmatrix}. \tag{5.19}$$

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:

$$\tilde{S}_t = [S_t \ \dot{S}_t] = [S_t \ e_t^{ij}]. \tag{5.20}$$

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:

$$e_t^{ij} = (S_{t-1} = i, S_t = j) \mid i \neq j. \tag{5.21}$$

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:

$$\Pi_\tau = \begin{bmatrix} \pi_{11,\tau} & \pi_{12,\tau} & \dots & \pi_{1M,\tau} \\ \pi_{21,\tau} & \pi_{22,\tau} & \dots & \pi_{2M,\tau} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{M1,\tau} & \pi_{M2,\tau} & \dots & \pi_{MM,\tau} \end{bmatrix}, \tag{5.22}$$

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 generalized superstate $\tilde{S}_m$ ($\tilde{S}_m \in \mathbf{S}$) is assumed to follow a multivariate Gaussian distribution with dimensionality $4d$. Thus, it can be represented by its sufficient statistics as generalized mean ($\tilde{\mu}_{\tilde{S}_m}$) and covariance ($\Sigma_{\tilde{S}_m}$). $\tilde{\mu}_{\tilde{S}_m}$ consists of the mean value $\mu_{\tilde{S}_m}$ describing the average of all the data samples encoded in this superstate $\tilde{S}_m$ in terms of I and Q as well as

the mean $\dot{\mu}_{\tilde{S}_m}$ describing the average of the corresponding derivatives ($\dot{I}$, $\dot{Q}$) and it is defined as:

$$\tilde{\mu}_{\tilde{S}_m} = [\mu_{\tilde{S}_m}, \dot{\mu}_{\tilde{S}_m}], \tag{5.23}$$

such that,

$$\mu_{\tilde{S}_m} = \frac{1}{Y}\sum_{y=1}^{Y} s_y, \quad \dot{\mu}_{\tilde{S}_m} = \frac{1}{Y}\sum_{y=1}^{Y} \dot{s}_y, \tag{5.24}$$

where Y is the total number of samples in $\tilde{S}_m$, $\mu_{\tilde{S}_m} \in \mathbb{R}^{2d}$, $\dot{\mu}_{\tilde{S}_m} \in \mathbb{R}^{2d}$, $s_y \in \tilde{S}_m$ representing the I, Q samples of the distribution and $\dot{s}_y \in \tilde{S}_m$ representing the $\dot{I}$, $\dot{Q}$ samples. $\Sigma_{\tilde{S}_m}$ is a $4d \times 4d$ matrix defined as:

$$\Sigma_{\tilde{S}_m} = \begin{bmatrix} \sigma_1\sigma_1 & \dots & \sigma_1\sigma_{4d} \\ \vdots & \ddots & \vdots \\ \sigma_{4d}\sigma_1 & \dots & \sigma_{4d}\sigma_{4d} \end{bmatrix}, \tag{5.25}$$

where σ_l is the variance of each component $l \in \{1, 2, \dots, 4d\}$ calculated as follows:

$$\sigma_l = \begin{cases} \mathbb{E}\left[(s^l - \mu_{\tilde{S}_m})(s^l - \mu_{\tilde{S}_m})^\intercal\right] & \text{if } 1 \le l \le d, \\ \mathbb{E}\left[(\dot{s}^l - \dot{\mu}_{\tilde{S}_m})(\dot{s}^l - \dot{\mu}_{\tilde{S}_m})^\intercal\right] & \text{if } d < l \le 2d. \end{cases} \tag{5.26}$$

5.5.3 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 [249]) 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. As mentioned previously, M-MJPF like MJPF uses a combination of Particle Filter (PF) and a bank of Kalman Filters (KFs) to predict the generalized superstates (at discrete level) and generalized states (at the continuous level), respectively. MJPF has been modified (i.e., M-MJPF) to involve multiple generalized errors in the filtering process in order to explain the detected abnormalities at multiple levels (both continuous and discrete), characterize the cause of such abnormalities (i.e., the jammer) and consequently learn the new emergent rules occurred in the environment (to be discussed in Sec.5.5.4 and Sec.5.5.5). Also, the M-MJPF has been adapted to capture discriminatory features, allowing the use of multiple abnormalities for jammer classification (to be discussed in Sec.5.5.6).

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:

$$< \tilde{S}_t^n, W_t^n > \sim < \pi(\tilde{S}_t), 1/N >,\ n \in N. \tag{5.27}$$

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 (5.13) and can be expressed in terms of the conditional probability $P(\tilde{X}_t|\tilde{X}_{t-1},\tilde{S}_t)$. In (5.13), 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 (5.23), 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:

$$\pi(\tilde{X}_t) = P(\tilde{X}_t, \tilde{S}_t|\tilde{Z}_{t-1}) = \int P(\tilde{X}_t|\tilde{X}_{t-1},\tilde{S}_t)\lambda(\tilde{X}_{t-1})d\tilde{X}_{t-1}, \tag{5.28}$$

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:

$$P(\tilde{X}_t,\tilde{S}_t|\tilde{Z}_t) = \pi(\tilde{X}_t)\lambda(\tilde{X}_t). \tag{5.29}$$

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:

$$W_t^n = W_t^n \lambda(\tilde{S}_t), \tag{5.30}$$

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

$$\lambda(\tilde{S}_t) = \lambda(\tilde{X}_t) P(\tilde{X}_t|\tilde{S}_t) = P(\tilde{Z}_t|\tilde{X}_t) P(\tilde{X}_t|\tilde{S}_t), \tag{5.31}$$

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:

$$D_B\big(\lambda(\tilde{X}_t), P(\tilde{X}_t|\tilde{S}_t = \tilde{S}_k)\big) = -ln \int \sqrt{\lambda(\tilde{X}_t) P(\tilde{X}_t|\tilde{S}_t = \tilde{S}_k)} d\tilde{X}_t, \tag{5.32}$$

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

$$D_\lambda = \Big[D_B\big(\lambda(\tilde{X}_t), P(\tilde{X}_t|\tilde{S}_t = \tilde{S}_1)\big), \ldots, D_B\big(\lambda(\tilde{X}_t), P(\tilde{X}_t|\tilde{S}_t = \tilde{S}_L)\big) \Big]. \tag{5.33}$$

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

$$\lambda(\tilde{S}_t) = \left[\frac{1/D_\lambda(1)}{1/\sum_{l=1}^{L} D_\lambda(l)}, \ldots, \frac{1/D_\lambda(L)}{1/\sum_{l=1}^{L} D_\lambda(l)} \right] \tag{5.34}$$

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** (see Appendix A.1).

5.5.4 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.

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:

$$KLDA = \sum_{i \in \mathcal{S}} Pr(\tilde{S}_t = i) D_{KL}\big(\pi(\tilde{S}_t) || \lambda(\tilde{S}_t)\big) + \sum_{i \in \mathcal{S}} Pr(\tilde{S}_t = i) D_{KL}\big(\lambda(\tilde{S}_t) || \pi(\tilde{S}_t)\big), \tag{5.35}$$

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:

$$Pr(\tilde{S}_t) = \frac{fr(\tilde{S}_t = i)}{N}, \tag{5.36}$$

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:

$$\mathcal{S} = \big\{ i | Pr(\tilde{S}_t) > 0 \big\},\ i \in \mathbf{S}. \tag{5.37}$$

The jammer detection decision is made by comparing $KLDA$ to a threshold (ψ). Thus, the hypothesis testing problem defined in (5.9) can be rewritten as:

$$\mathcal{H}_0 : KLDA < \psi, \quad \mathcal{H}_1 : KLDA > \psi. \tag{5.38}$$

In addition, the generalized errors ($\tilde{\varepsilon}_{\tilde{S}_t}$) associated with the abnormality indicator (5.35) allows to understand how the jammer affected the discrete level of the reference model. Thus, after detecting the jammer at the discrete level using (5.35), 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:

$$\tilde{\varepsilon}_{\tilde{S}_t} = \lambda(\tilde{S}_t) - \pi(\tilde{S}_t), \tag{5.39}$$

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:

$$CLB = -\ln\left(\mathcal{BC}\big(\pi(\tilde{X}_t), P(\tilde{X}_t|\tilde{S}_t)\big)\right), \tag{5.40}$$

where

$$\mathcal{BC} = \int \sqrt{\pi(\tilde{X}_t)P(\tilde{X}_t|\tilde{S}_t)}d\tilde{X}_t, \tag{5.41}$$

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:

$$CLA = -\ln\left(\mathcal{BC}\big(\pi(\tilde{X}_t), \lambda(\tilde{X}_t)\big)\right), \tag{5.42}$$

where

$$\mathcal{BC} = \int \sqrt{\pi(\tilde{X}_t)\lambda(\tilde{X}_t)}d\tilde{X}_t. \tag{5.43}$$

Thus, jammer detection at the continuous level is made by comparing *CLA* to a threshold (η) which is stated as:

$$\mathcal{H}_0 : CLA < \eta, \quad \mathcal{H}_1 : CLA > \eta. \tag{5.44}$$

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 5.5.2), the generalized error ($\tilde{\varepsilon}^{[1]}_{\tilde{X}_t}$) projected on the continuous space and associated with (5.42) is defined in (5.17). On the other hand, it would be possible to calculate the Generalized Errors $\tilde{\varepsilon}^{[2]}_{\tilde{X}_t}$ (associated with (5.40)) 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:

$$\tilde{\varepsilon}^{[2]}_{\tilde{X}_t} = \begin{cases} \tilde{\mu}\left(\underset{\tilde{S}_t \in S}{\operatorname{argmax}}\, \lambda(\tilde{S}_t)\right) - \tilde{X}_t \quad if \;\; \tilde{S}^{\pi}_t = \tilde{S}^{\lambda}_t, \\ \tilde{\mu}\left(\underset{\tilde{S}_t \in S}{\operatorname{argmax}}\, \lambda(\tilde{S}_t)\right) - \tilde{\mu}\left(\underset{\tilde{S}_t \in S}{\operatorname{argmax}}\, \pi(\tilde{S}_t)\right) \quad if \;\; \tilde{S}^{\pi}_t \neq \tilde{S}^{\lambda}_t, \end{cases} \tag{5.45}$$

where $\tilde{S}^{\pi}_t = \underset{\tilde{S}_t \in S}{\operatorname{argmax}}\, \pi(\tilde{S}_t)$ is the expected superstate and $\tilde{S}^{\lambda}_t = \underset{\tilde{S}_t \in S}{\operatorname{argmax}}\, \lambda(\tilde{S}_t)$ 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.

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 defined in (5.16).
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:

$$\tilde{\varepsilon}^{[2]}_{\tilde{Z}_t} = \tilde{Z}_t - H\tilde{\mu}(\underset{\tilde{S}_t \in S}{\operatorname{argmax}}\, \lambda(\tilde{S}_t)), \tag{5.46}$$

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 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).

5.5.5 Jammer extraction and learning dynamic models

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

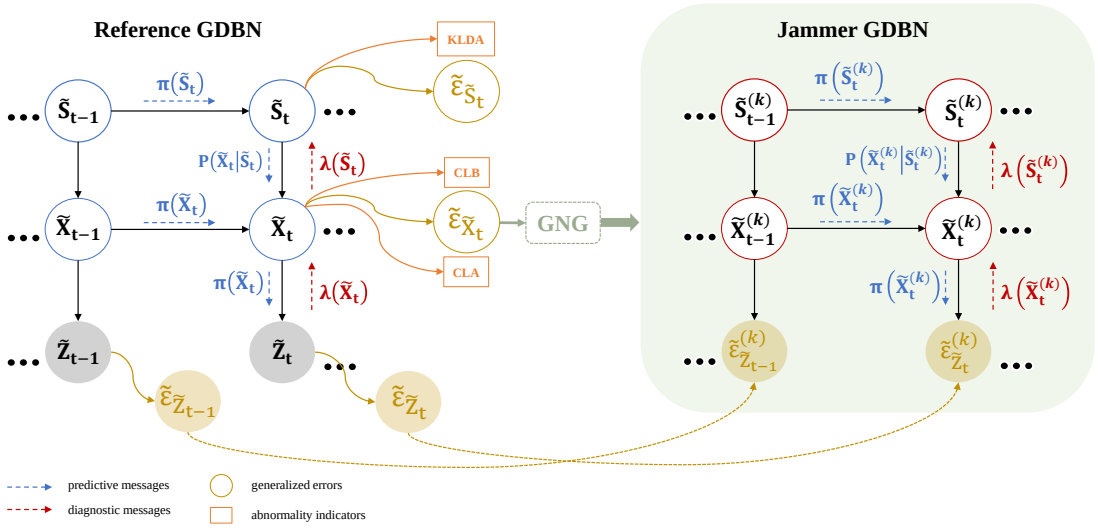


Fig. 5.3 Schematic that illustrate the process of learning a separated model of the detected jammer under the k-th modulation mode by exploiting the Generalized Errors. Also, it illustrates the relationship between the reference GDBN and the jammer's GDBN under the k-th modulation scheme. The reference 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.

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. 5.3). The generalized errors representing the jamming signal under the k-th modulation scheme are clustered using GNG, which provides a set $\mathbf{S^{(k)}}$ 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 $\mathbf{S^{(k)}}$, we obtain a set $\mathcal{S}_{\mathcal{M}}$ of jamming dynamic models describing the jammers dynamic behaviours under different modulations, such that:

$$\mathcal{S}_{\mathcal{M}} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_K\}. \tag{5.47}$$

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 $\tilde{\mu}_{\tilde{S}_m^{(k)}}$ of *conditional generalized mean values* defined as:

$$\tilde{\mu}_{\tilde{S}_m^{(k)}} = \left[\tilde{\mu}_{\tilde{S}_m^{(k)}|\tilde{S}_1^{(k)}}, \tilde{\mu}_{\tilde{S}_m^{(k)}|\tilde{S}_2^{(k)}}, \ldots, \tilde{\mu}_{\tilde{S}_m^{(k)}|\tilde{S}_M^{(k)}}\right], \tag{5.48}$$

where the *conditional control vectors* ($U_{\tilde{S}_m^{(k)}}$) are encoded such that:

$$U_{\tilde{S}_m^{(k)}} = \left[U_{\tilde{S}_m^{(k)}|\tilde{S}_1^{(k)}}, U_{\tilde{S}_m^{(k)}|\tilde{S}_2^{(k)}}, \ldots, U_{\tilde{S}_m^{(k)}|\tilde{S}_M^{(k)}} \right], \tag{5.49}$$

and a set $\Sigma_{\tilde{S}_m^{(k)}}$ of *conditional covariance matrices* defined as:

$$\Sigma_{\tilde{S}_m^{(k)}} = \left[\Sigma_{\tilde{S}_m^{(k)}|\tilde{S}_1^{(k)}}, \Sigma_{\tilde{S}_m^{(k)}|\tilde{S}_2^{(k)}}, \ldots, \Sigma_{\tilde{S}_m^{(k)}|\tilde{S}_M^{(k)}} \right]. \tag{5.50}$$

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.

5.5.6 Online 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 (5.48), (5.49) and (5.50)). 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. 5.4). The UAV can evaluate which of these predictions explain the current radio situation by using the abnormality measurement defined in (5.42) applied to the jammer model and defined as:

$$Abn_k = -\ln\left(\mathcal{BC}\big(\pi(\tilde{X}_t^{(k)}), \lambda(\tilde{X}_t^{(k)})\big) \right), \tag{5.51}$$

where

$$\mathcal{BC} = \int \sqrt{\pi(\tilde{X}_t^{(k)})\lambda(\tilde{X}_t^{(k)})}\, d\tilde{X}_t^{(k)}. \tag{5.52}$$

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

$$\mathcal{S}_{Abn}(t) = \{Abn_1, Abn_2, \ldots, Abn_K\}. \tag{5.53}$$

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

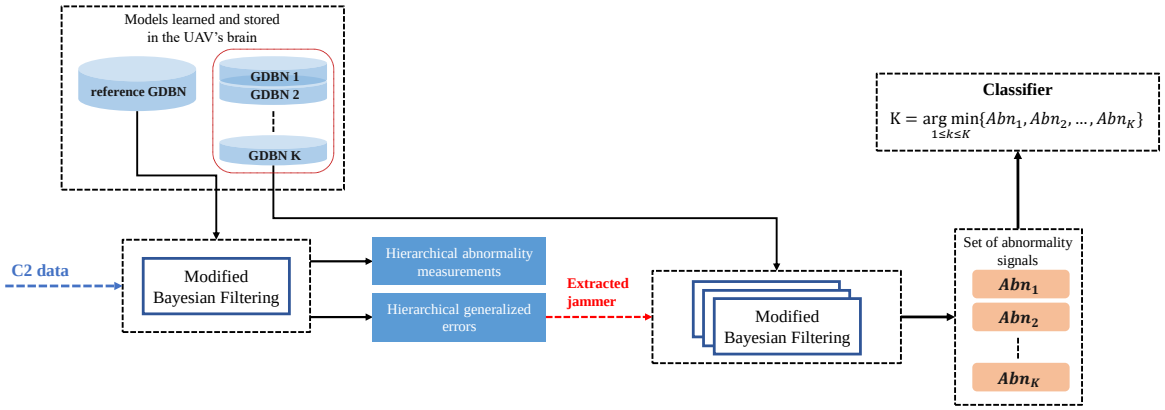


Fig. 5.4 GDBN-based Jamming Modulation Classification Framework.

that:

$$\mathcal{S}_{mod} = \{1, \ldots, K\}. \tag{5.54}$$

Then, the modulation classification can be made by comparing 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:

$$\hat{k}(t) = \underset{1 \leq k \leq K}{\operatorname{argmin}} \{\mathcal{S}_{Abn}\}, \text{where } \hat{k}(t) \in \mathcal{S}_{mod}, \tag{5.55}$$

where $\hat{k}(t)$ is the index of the jamming model $\mathcal{M}_{\hat{k}(t)} \in \mathcal{S}_{\mathcal{M}}$ providing the minimum abnormality (i.e., $Abn_{\hat{k}(t)}$). 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:

$$P_{cc} = \frac{1}{T} \sum_{t=1}^{T} P(\hat{k}(t) = k(t) | k(t)), \tag{5.56}$$

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** (see Appendix A.2).

5.5.7 Complexity Analysis

In this section, we analyse the complexity of Algorithm 1 and Algorithm 2. The main operations in Algorithm 1 are the joint prediction at multiple levels and the computation of abnormality measurements. At each time instant t, the cognitive-UAV performs N predictions at the discrete level and N predictions at the continuous level (i.e., predictions of Generalized

state vectors of $4 \times d$ dimensions where d is the number of the sub-carriers sensed by the UAV). Therefore, considering the whole time T (i.e., the total number of OFDM symbols received by the UAV during the total flight time), the time computational complexity is given by $\mathcal{O}(N \times T) + \mathcal{O}(N \times T)$. The complexity of detecting the jammer at two hierarchical levels concerns calculating the Kullback-Leibler-Divergence (*KLDA*) at the discrete level and the Bhattacharyya distance (*CLA* and *CLB*) at the continuous level. The complexity of *KLDA* is $O(I \times M \times T)$ where I is the total number of winning superstates in $\mathcal{S}$ (defined in (5.37)) and M is the total number of superstates in $\mathbf{S}$ (defined in (5.18)). The complexity of *CLA* is $O(2d \times T)$ where d is the number of the sensed sub-carriers. It is worth noting that the predictions and generation of abnormalities must be made before the time of arrival, i.e. if the UAV receives a set of commands each $50ms$ it must perform the predictions and determine whether or not a jammer is attacking within this time. So, the execution of this process and how fast it is, depends on the processors onboard the UAV. The complexity of the main operations involved in Algorithm 1 are summarized in Table 5.1.

Table 5.1 Complexity Analysis of **Algorithm 1**

Procedure	**Complexity Order**
Prediction at Discrete Level	$\mathcal{O}(N \times T)$
Prediction at Continuous Level	$\mathcal{O}(N \times T)$
Abnormality Measurement (KLDA)	$\mathcal{O}(I \times M \times T)$
Abnormality Measurement (CLA)	$\mathcal{O}(2d \times T)$
Abnormality Measurement (CLB)	$\mathcal{O}(2d \times T)$

In Algorithm 2, at each time instant t, predictions and jammer detection (through abnormality measurements) are performed by recalling Algorithm 1 (i.e., M-MJPF). Once an abnormality is detected (i.e., jammer is present), the algorithm uses all the available jamming models to provide multiple predictions at multiple levels. The complexity of the prediction operation at discrete level is given by $\mathcal{O}(K \times N \times T)$ where K is the total number of jamming models learned so far, such that $K \in \mathcal{S}_{\mathcal{M}}$ and N is the total number of particles propagated by PF. Likewise, the complexity of the prediction operation at the continuous level is $\mathcal{O}(K \times N \times T)$. The computational complexity of the abnormality measurement (*Abn*), which is based on the Bhattacharyya distance, is the same as for CLA and CLB and given by $\mathcal{O}(2d \times T)$ (refer to Table 5.2).

Moreover, in the proposed framework, the cognitive-UAV is not transmitting or exchanging any signal with other entities in the network, which usually has a significant impact on the computational complexity and can impose an additional burden on the UAV.

Table 5.2 Complexity Analysis of **Algorithm 2**

Procedure	Complexity Order
Prediction at Discrete Level	$\mathcal{O}(K \times N \times T)$
Prediction at Continuous Level	$\mathcal{O}(K \times N \times T)$
Abnormality Measurement (*Abn*)	$\mathcal{O}(2d \times T)$

5.6 Simulation Results and Discussion

5.6.1 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 [257]. 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 [258] and the important parameters defined in Table 5.3. Similarly, the altered trajectory is extracted from the jammed LTE signal.

Table 5.3 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. 5.5-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

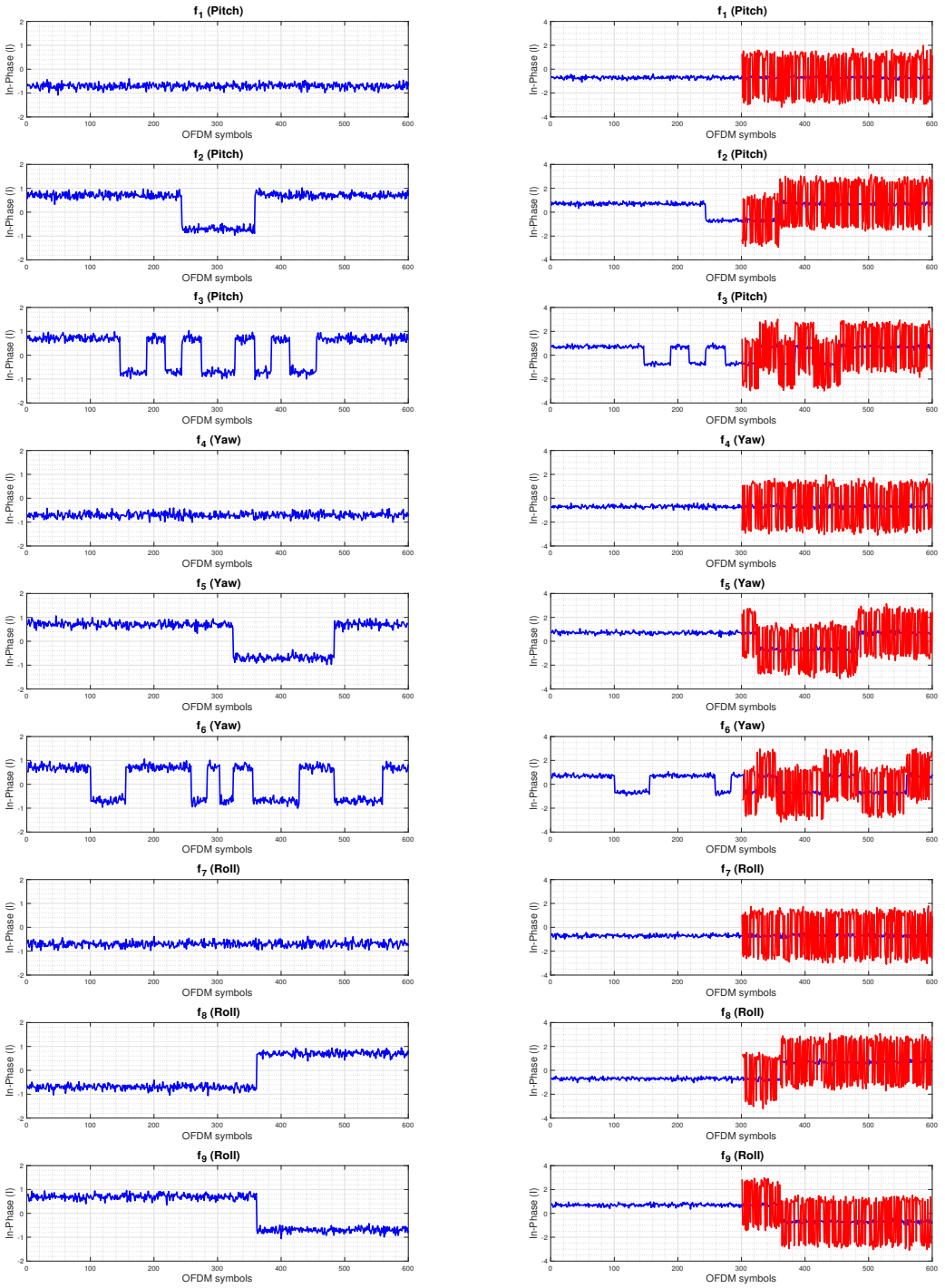
related to the original commands sent by the operator (see Fig. 5.5-a) which is used to learn the reference GDBN model. The UAV trajectory during this situation is depicted in Fig. 5.6-a. ***(ii)*** **Abnormal Situation**: during this situation the jammer uses 2 configurations. The first one (used in Section 5.6.2) 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 5.6.3) is related to the jammer who is attacking from $t = 1$ till $t = 300$ to evaluate the classification performance after learning the jamming models.

5.6.2 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 (5.17)) 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 without being surprised anymore by the observations.

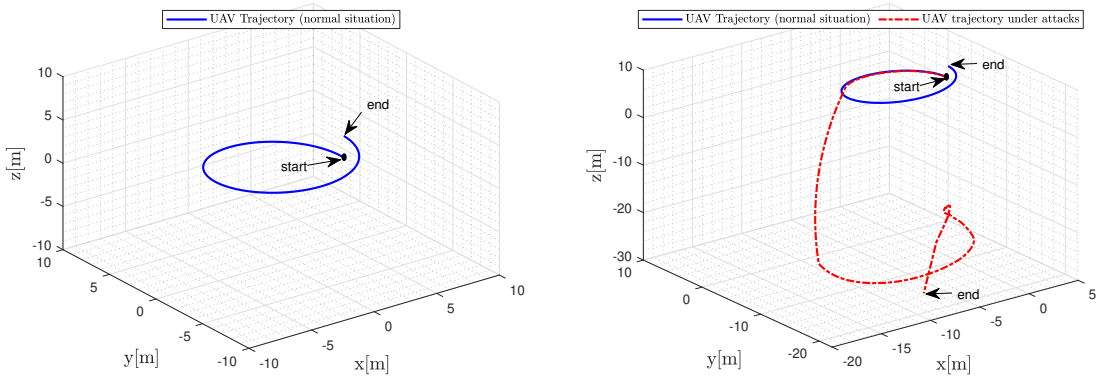
Fig. 5.7 verifies this where we can observe a high abnormality signal all the time by using the initial GDBN (due to the lack of knowledge about the environmental dynamics) and a quasi-zero abnormality signal by using the reference GDBN that encodes the dynamic rules of the signals allowing by that the 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 (5.35) and (6.24). 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 $+20$dB as shown in Fig. 5.8. 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



(a) Reference Situation (b) Abnormal Situation

Fig. 5.5 The received commands 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).



(a) Reference Situation (b) Abnormal Situation

Fig. 5.6 UAV trajectories during different situations. (a) UAV trajectory in a normal situation. (b) An example of UAV trajectory under jamming attacks. Blue and red colors represent the trajectory without and with jammer attacks, respectively.

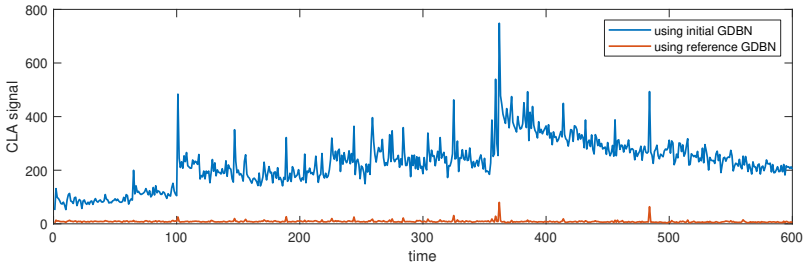
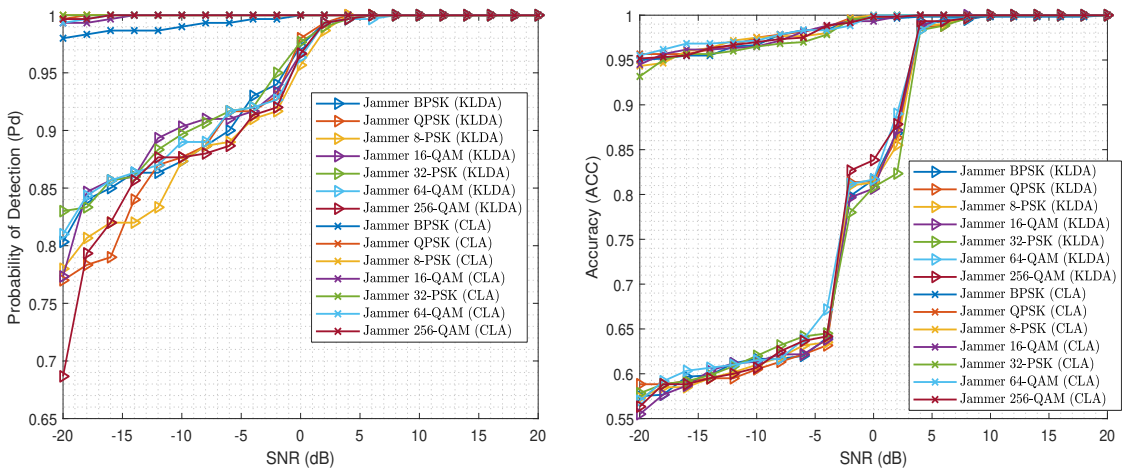


Fig. 5.7 Abnormality indicator at the continuous levels (defined in (6.24)) using the initial GDBN model and the reference GDBN model.

the dynamic transitional rules efficiently. Moreover, rapid changes in the accuracy of KLDA in Fig. 5.8(b) from $-4\text{dB} \leq \text{SNR} \leq +4\text{dB}$ is due to the increased probability of false alarm, which is affected by the rapid transitions of the received signal among the discrete regions. 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.

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 (5.46). Fig. 5.9 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. 5.10 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



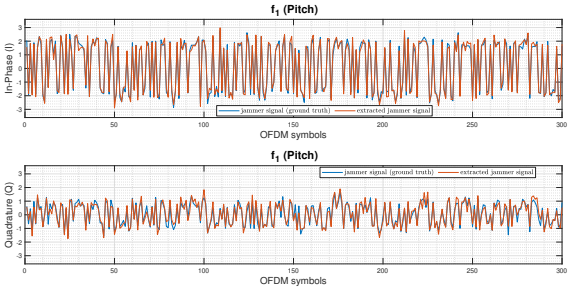
(a) (b)

Fig. 5.8 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.

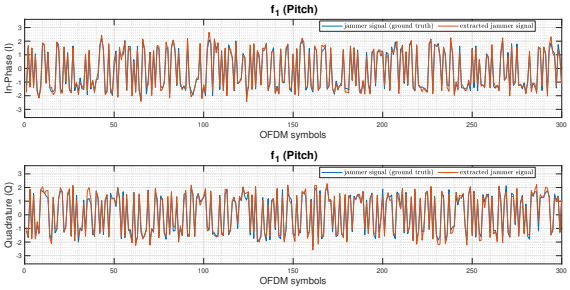
a set of GDBN models forming the set $\mathcal{S}_{\mathcal{M}}$ as defined in (5.47). In this way, the UAV's brain consists of the reference model that describes what commands the UAV is expecting to receive under normal circumstances and another set of models ($\mathcal{S}_{\mathcal{M}}$) representing the dynamic behaviour of multiple jammers using different modulation 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.

5.6.3 Online Classification Process

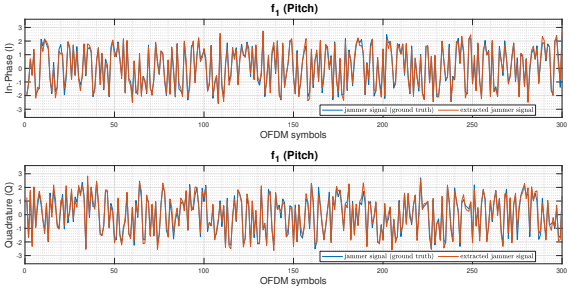
In Fig. 5.11, we showed the classification accuracy of the proposed GDBN for each modulation scheme in the candidate set ($\mathcal{S}_{mod}$). We can observe that GDBN achieves high classification accuracy for most of the modulation schemes, especially at $SNR > 5dB$. The low accuracy at low SNRs ($SNR < 0dB$) 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



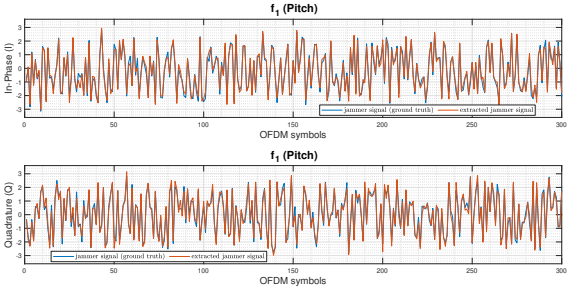
(a) Jammer BPSK



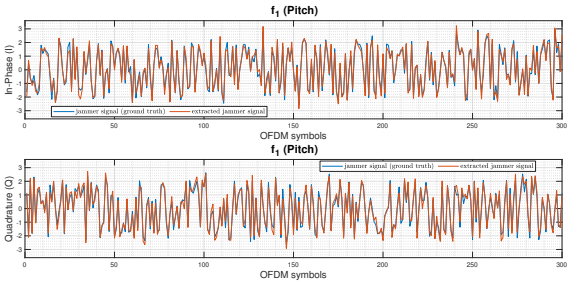
(b) Jammer QPSK



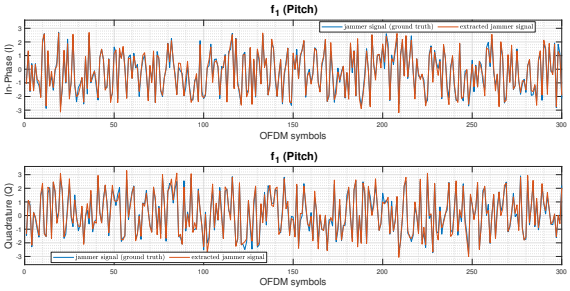
(c) Jammer 8-PSK



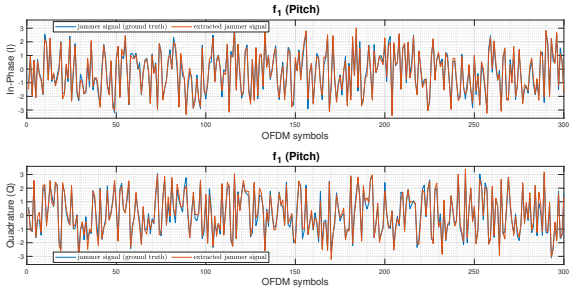
(d) Jammer 16-QAM



(e) Jammer 32-PSK



(f) Jammer 64-QAM



(g) Jammer 256-QAM

Fig. 5.9 I/Q time domain plot of the extracted jamming signals (based on $\tilde{\varepsilon}_{\tilde{Z}_t}^{[2]}$ defined in (5.46)) under different modulation schemes and SNR=10 dB at sub-carrier f_1 and of the ground truth jamming signals.

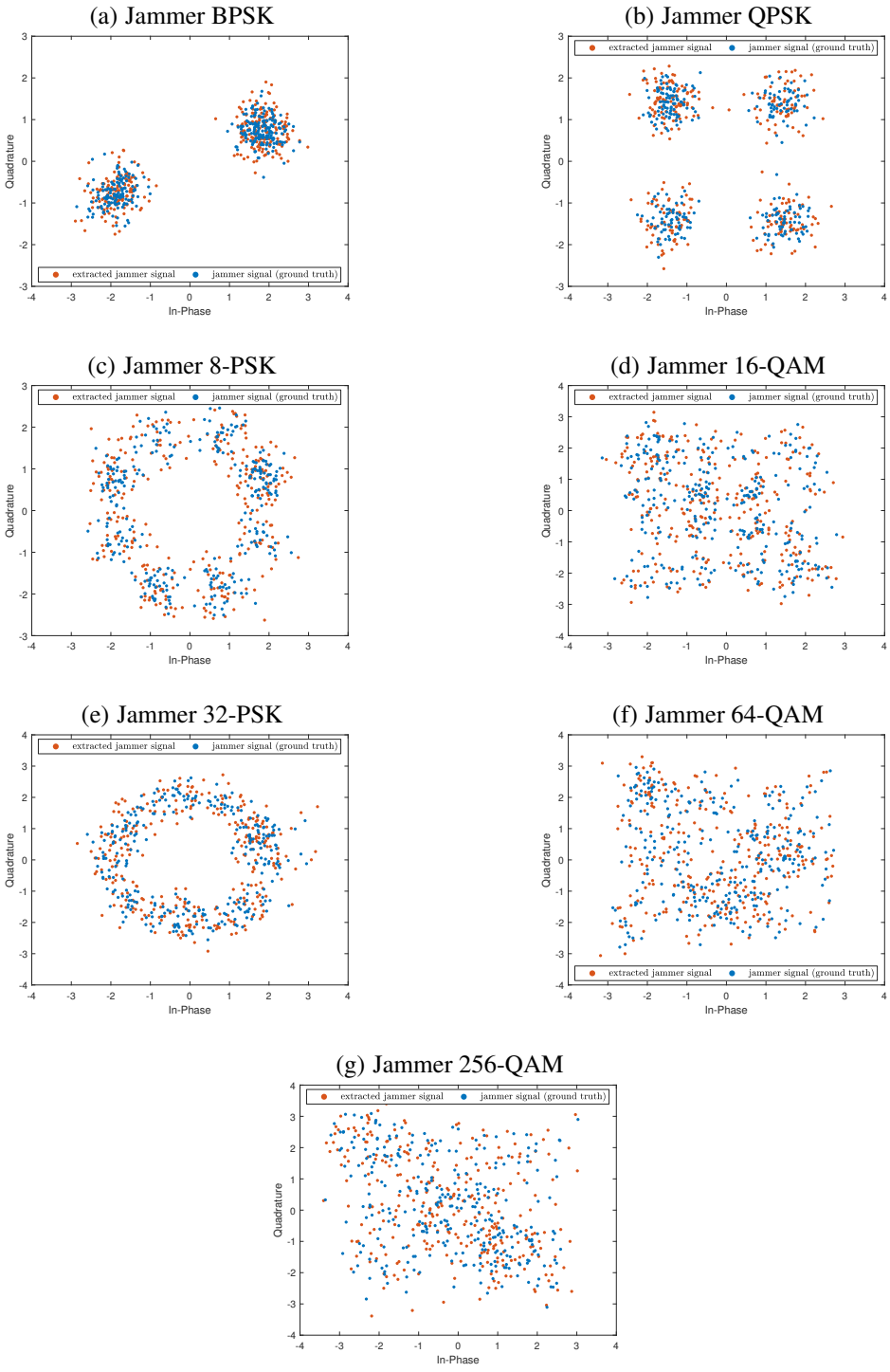


Fig. 5.10 Scatter plot of the extracted jamming signals (based on $\tilde{\varepsilon}^{[2]}_{\tilde{Z}_t}$ defined in (5.46)) and the corresponding ground truth at sub-carrier f_1 and 10 dB SNR.

encoded in the GDBN model in an efficient way. Some examples of the resultant confusion matrices at various SNR ratios are exhibited in Fig. 5.12.

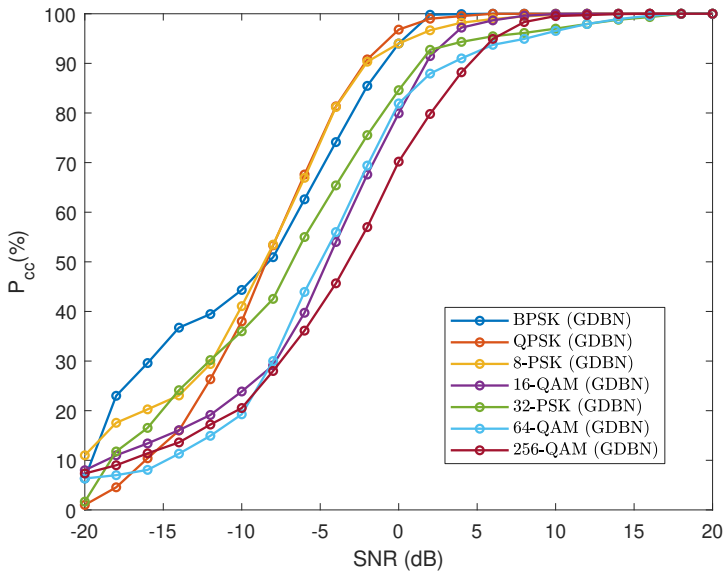


Fig. 5.11 Performance evaluation of the proposed GDBN-based framework: Probability of correct classification for each modulation scheme versus SNR.

In addition, we compare the performance of the proposed GDBN with the Convolutional Neural Network (CNN), the Long Short Term Memory (LSTM) and the Stacked AutoEncoder (SAE). 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) for all three CNN, LSTM and SAE for a fair comparison. For CNN, we used the same configuration (i.e. same number of layers) employed in [302], 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 $\mathcal{S}_{mod}$. The SAE consists of two autoencoders stacked on top of one another (trained in an unsupervised manner) and a softmax layer for classification (trained in a supervised fashion using labels for the training signals). Fig. 5.13, shows the performance comparison between the proposed GDBN, LSTM, CNN and SAE. 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. 5.14, and it shows that the proposed GDBN beats LSTM, CNN and SAE especially at $SNR > 5dB$. This means that the proposed approach succeeded to learn the dynamic proprieties (at hierarchical levels)

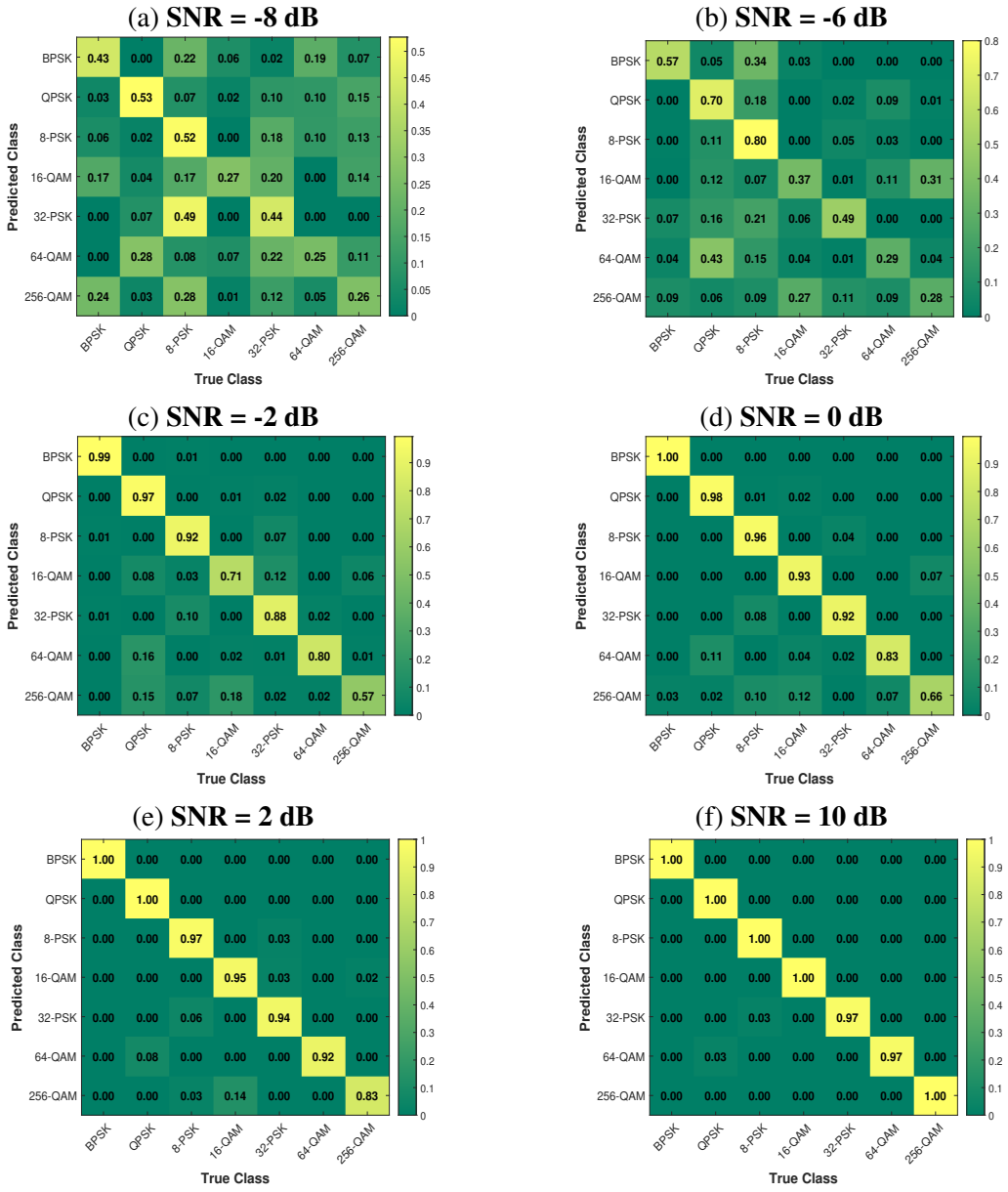


Fig. 5.12 Confusion matrices for the proposed GDBN method at various SNR values.

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, CNN and SAE 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

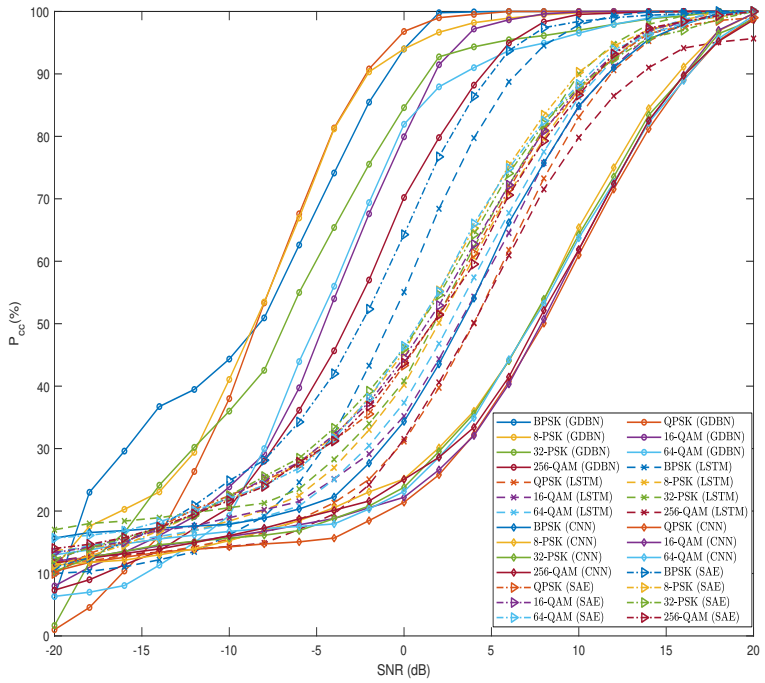


Fig. 5.13 Performance comparison between the GDBN, LSTM, CNN and SAE: Probability of correct classification for each modulation scheme versus SNR.

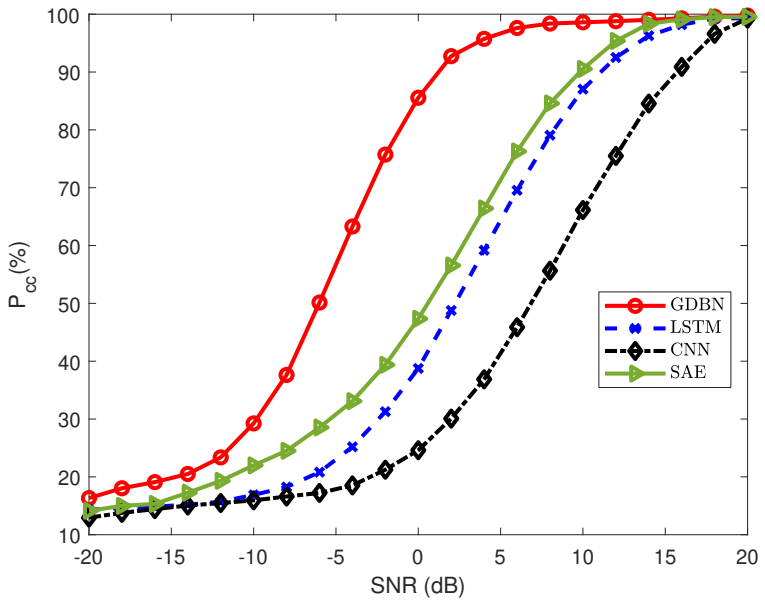


Fig. 5.14 Performance comparison between the GDBN, LSTM, CNN and SAE: Overall probability of correct classification versus SNR.

situations incrementally. This is difficult in LSTM, CNN and SAE 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 super-states 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. 5.15. We can observe from Fig. 5.15, that increasing the number of neurons (L) improves the classification accuracy 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. The impact of the number of neurons on the classification accuracy can be better understood by evaluating the overall performance of the proposed approach in classifying various modulation schemes. Fig. 5.16 shows the overall probability of correct classification (over all modulation schemes) and gives a clear idea of how the performance changes as changing the L parameter. It is clear that increasing the number of neurons (L) improves the overall classification accuracy.

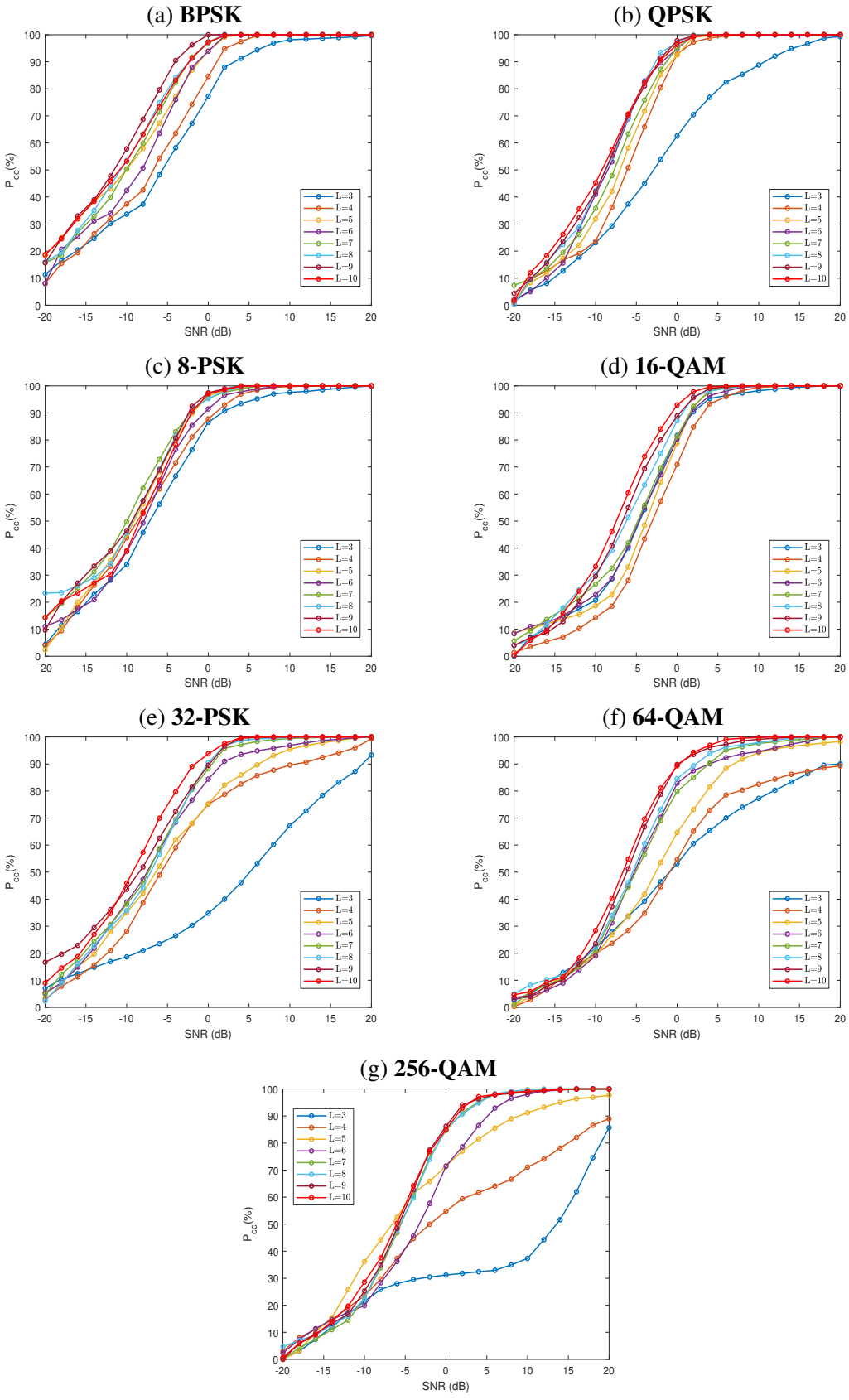


Fig. 5.15 Performance evaluation in terms of classification accuracy of the proposed GDBN method using different number of neurons (L) to learn the jamming models.

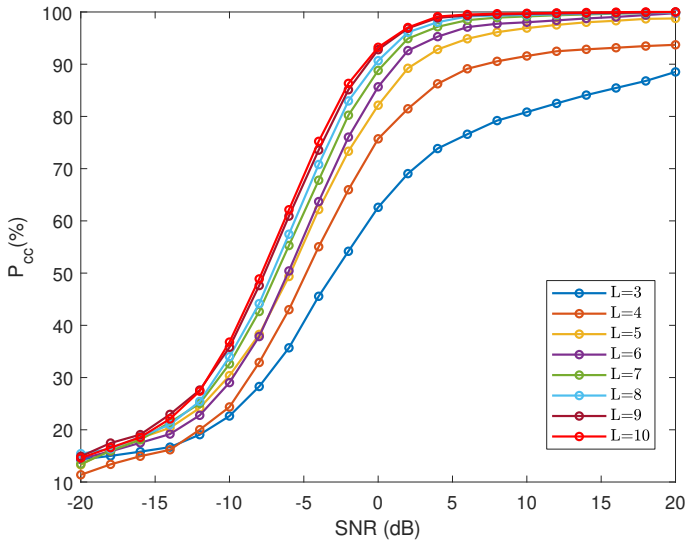


Fig. 5.16 Performance evaluation in terms of overall classification accuracy of the proposed GDBN method using a different number of neurons (L) to learn the jamming models.

5.7 RadioML Dataset

In this section, we validated the proposed approach for automatic modulation classification using a real dataset.

5.7.1 Real Dataset

We employed a real dataset named RadioML (version 2018[2]) [302] to assess the performance of the proposed GDBN-based AMC after running extensive simulations. The candidate set of the modulation schemes picked from the dataset is $\mathcal{S}_{mod}$={OOK, QPSK, 32-PSK, 16-QAM, 32-QAM, 64-QAM, 256-QAM}. The dataset was built using GNU radio block that includes different effects as center frequency offset, sample rate offset, selective fading and AWGN to simulate real-world radio conditions. The dataset consists of about 2 million examples (which we call events) under different SNR values. The SNR ranges from -20dB to +30dB with a step size of 2dB. In our study, each event is divided into two subsets 50% for training and 50% for testing, and the classification task is performed at each event to classify between single complex symbols. The challenge of this approach is the ability to perform accurate classification without requiring many symbols, which improve the latency and make it possible to recognize the modulation scheme in a real-time manner just by processing one symbol, which is crucial in the IoT networks.

[2]Dataset available on https://www.deepsig.ai/datasets

5.7.2 Results and discussion

During the training process, the radio learns a GDBN model for each modulation scheme. After this process, the radio possesses K GDBN models stored in its brain, where each model encodes the dynamic behaviour of the corresponding modulation. During the testing process, the radio performs multiple predictions in parallel and calculate the abnormality indicator as defined in (5.51) where the classifier pick the minimum abnormality signal (among the K abnormality signals) as defined in (5.55) to recognize the modulation scheme.

In Fig. 5.17, we showed the classification accuracy of the proposed GDBN for each modulation scheme in $\mathcal{S}_{mod}$. 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 (<0dB) for the majority of the modulation schemes in $\mathcal{S}_{mod}$ 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 thus the dynamics become very fast which make it difficult to discover and capture these dynamic rules that are encoded in the GDBN model in an efficient way.

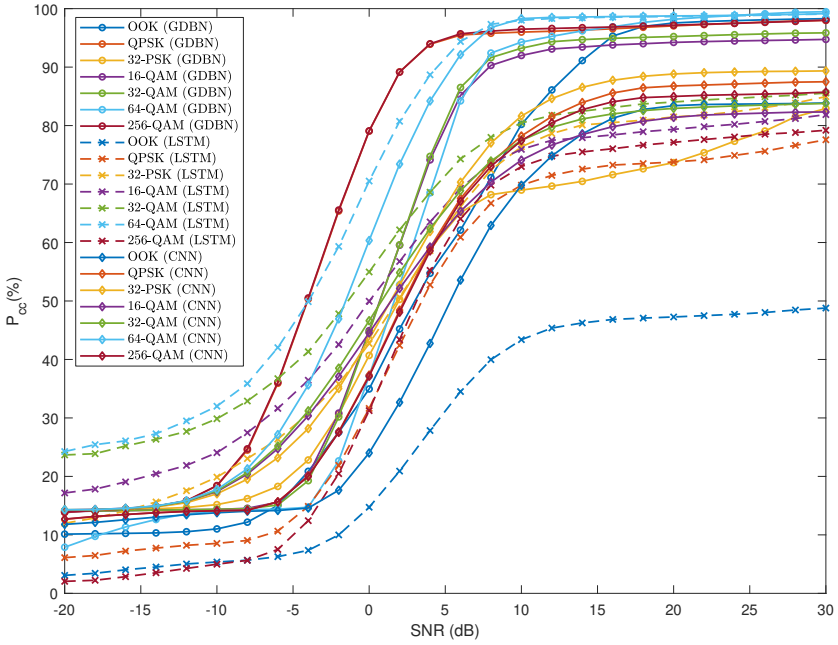


Fig. 5.17 The performance comparison between GDBN, LSTM and CNN.

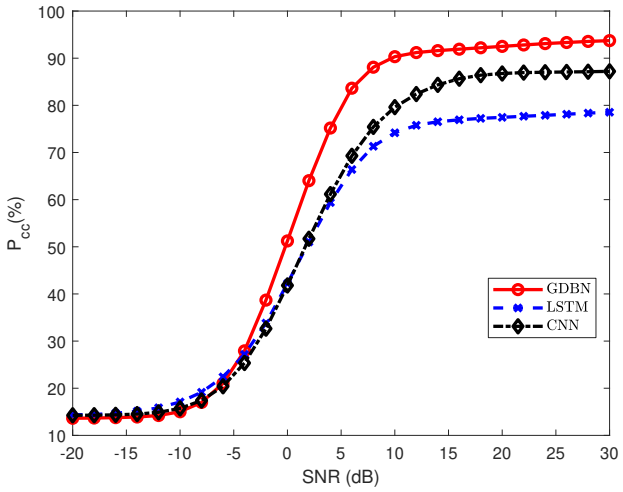


Fig. 5.18 The overall performance comparison among GDBN, LSTM and CNN.

In addition, we compared the performance of the proposed GDBN with the CNN and the 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) for both CNN and LSTM for a fair comparison. Moreover, for CNN, we used the same configuration (i.e. same number of layers) employed in [302], but with different input, here we used a state vector consisting of IQ components and the corresponding derivatives. While the LSTM used here has three layers, one LSTM layer, one fully connected layer, and finally a dense softmax layer that maps the classified features to one of the available modulation schemes in $\mathcal{S}_{mod}$. Again, Fig. 5.17 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 P_{cc} related to each modulation. The overall comparison is depicted in Fig. 5.18, and it shows that the proposed GDBN beats both LSTM and CNN in the considered scenario, especially at SNR>5dB. This means that the proposed approach succeeded in learning 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 the case of LSTM and CNN, where the dependencies between the hidden variables at multiple layers are viewed as a black-box, and thus 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.

5.8 Proposed Framework for Joint Automatic Modulation Format Conversion and Classification (AMCC)

This section proposes a joint automatic modulation conversion and classification (AMCC) framework, which allows an AI-enabled wireless node to predict signals' dynamics of different modulation schemes and explain how it can be transported (converted) with minimal effort and forwarded with higher spectral efficiency. To achieve this goal, we propose a Generalized Filtering framework integrated by Transport Planning to learn the way of converting low-order modulations to high-order modulations, which has also been validated by performing the automatic modulation classification. Simulation results demonstrate the effective performance of our novel framework on converting and classifying multiple modulation formats. This section describes: 1) the representation of the wireless environment; 2) the process of learning the dynamic model where the top-level of the hierarchy can be represented in graphs from which the transport plan can be learned; 3) how the Double Generalized Dynamic Bayesian Network (DGDBN) integrated with the M-MJPF allows to convert the actual modulation format and recognize the modulation scheme of the sensory signals.

5.8.1 Signal Model and Problem Formulation

The modulation conversion task can be expressed as finding an optimal transport plan to transfer signals from source distributions to target ones. In this work we focus on converting low-order to high-order modulation formats. Lets take as an example the case of converting a source format (QPSK) to target format (16QAM), first it is important to find the similarities by mapping the $M_1 = 4$ constellations of QPSK to the $M_2 = 16$ constellations of 16QAM. Mapping is based on measuring the corresponding distances associated with the minimum force needed to move data samples from the source constellation to proper target ones. Second, it is important to synchronize the conversion process by performing re-timing or applying temporal delay since converting 1 symbol of QPSK needs 2 symbols of 16QAM as QPSK encodes 2 bits per symbol and 16QAM encodes 4 bits per symbol so to recover the 4

bits correctly in 16QAM we need to have 4 bits transmitted over 2 symbols in QPSK. The main idea is to convert the signal from QPSK to 16QAM and ensure the correct recovery of the original data sequence (bits) which can be validated through the calculation of the Bit-error-rate (BER).

The modulation classification task can be expressed as a classification problem with K modulation formats. The received/sensed signal by the AI-enabled radio can be stated as:

$$r_t = he^{j(2\pi ft+\theta)}\mathrm{s}_t^k + v_t, \tag{5.57}$$

where h is the channel coefficient, f is the frequency offset and θ is the phase offset and v_t is the Additive Gaussian Noise (AWGN) which is drawn from a zero-mean normal distribution with variance (σ_v^2). $\mathrm{s}_t^{(k)}$ is the complex symbol under the k-th modulation format which can be represented as:

$$\mathrm{s}_t^k = \big[A_m \sum_n a_n g(t-nT_s)\big] cos\big(2\pi(f_c+f_m)t+\phi_0+\phi_m\big), \tag{5.58}$$

if transmitted signal is M-PSK. In (5.58), A_m, a_n, T_s, f_c, f_m, ϕ_0 and ϕ_m are the modulation amplitude, symbol sequence, symbol period, carrier frequency, modulation frequency, initial phase, and modulation phase, respectively. The function $g(t)$ equals to 1 if $1 \leq t \leq T_s$ and to 0 otherwise. In case of M-QAM s_t^k can be represented as:

$$\mathrm{s}_t^k = \big[A_m \sum_n a_n g(t-nT_s)\big] cos(2\pi f_c t+\phi_0) + \big[A_m \sum_n b_n g(t-nT_s)\big] sin(2\pi f_c t+\phi_0), \tag{5.59}$$

where a_n, $b_n \in [(2m-1-\sqrt{M})], m = \{1,2,\ldots,\sqrt{M}\}$, and the two carriers are modulated by a_n and b_n. The aim of the classifier is to identify s_t from the observation r_t and give out $\mathrm{P}(\mathrm{s}_t \in k|r_t)$ where $k \in K$.

The following section explains in detail the proposed framework for the joint modulation conversion and classification following a data-driven approach which can be applicable in a pure software manner using e.g., software defined radios.

5.8.2 RF representation

We assume that the hidden dynamics of the physical signals present in the radio environment and generated by various entity nodes (e.g., IoT sensors) can be cast at hierarchical levels and treated as conditionally dependent variables. We represented those variables in generalized coordinates of motion and employed Generalized Filtering [303], i.e., Bayesian filtering in generalized coordinates. The hierarchical causal models representing the signals' dynamics

can be structured in a Generalized Dynamic Bayesian Network (GDBN) and formulated in terms of stochastic processes defined as:

$$\tilde{\mathrm{S}}_t^{(e)} = \mathrm{f}(\tilde{\mathrm{S}}_{t-1}^{(e)}) + \tilde{w}_t, \tag{5.60}$$

$$\tilde{\mathrm{X}}_t^{(e)} = \mathrm{g}(\tilde{\mathrm{X}}_{t-1}^{(e)}, \tilde{\mathrm{S}}_t^{(e)}) + \tilde{w}_t = \mathrm{A}\tilde{\mathrm{X}}_{t-1}^{(e)} + \mathrm{B}\mathrm{U}_{\tilde{\mathrm{S}}_t^{(e)}} + \tilde{w}_t, \tag{5.61}$$

$$\tilde{\mathrm{Z}}_t^{(e)} = \mathrm{h}(\tilde{\mathrm{X}}_t^{(e)}) + \tilde{v}_t = \mathrm{H}\tilde{\mathrm{X}}_t^{(e)} + \tilde{v}_t. \tag{5.62}$$

The dynamics at the top level of hierarchy evolve according to (5.60) where $\mathrm{f}(.)$ is a non-linear function determining the causal transitions at that level and depends on time-varying transition probabilities which is subject to random noise $\tilde{w}_t$ that is assumed to be drawn from a zero multivariate normal distribution with covariance $\Sigma_{\tilde{w}_t}$ such that $\tilde{w}_t \sim \mathcal{N}(0, \Sigma_{\tilde{w}_t})$. $(.)^{(e)}$ symbolises an entity node adopting a particular digital modulation format. The top level holds a belief about the level below and guides the prediction at that level after indicating the control vector ($\mathrm{U}_{\tilde{\mathrm{S}}_t^{(e)}}$) to be used as pointed out in (5.61). Thus, the Generalized States' (GS) evolution depends on the previous GS ($\tilde{\mathrm{X}}_{t-1}^{(e)}$) and the force enocoded in $\mathrm{U}_{\tilde{\mathrm{S}}_t^{(e)}}$. In (5.61), $\mathrm{A} \in \mathbb{R}^d$ and $\mathrm{B} \in \mathbb{R}^d$ are the dynamic and control model matrices that parametrize the linear dynamics at the medium level of hierarchy where d stands for the signal's features (I,Q components) dimensionality. Hence, (5.60) and (5.61) allow making inferences about hidden states causing sensory signals that can be interpreted by prior beliefs about the underlying model generating those signals. The bottom level of the hierarchy stands for the observed sensory signals modeled in (5.62) where H is the observation matrix that parametrize the observation model and maps hidden GSs ($\tilde{\mathrm{X}}_{t-1}^{(e)}$) to Generalized Observations ($\tilde{\mathrm{Z}}_{t-1}^{(e)}$); $\tilde{v}_t \sim \mathcal{N}(0, \Sigma_{\tilde{v}_t})$ is the measurement noise.

5.8.3 Learning GDBN

We assume that the radio starts perceiving the surrounding environment supposing that no signals are present in the spectrum and observations are only subject to random noise. At this stage (1^{st} iteration), the radio adopts an initial GDBN model consisting of two levels, the GS level and observation level and employs an Unmotivated Kalman Filter (UKF) under this static assumption by using the following simplified dynamic model:

$$\tilde{\mathrm{X}}_t^{(e)} = \mathrm{A}\tilde{\mathrm{X}}_{t-1}^{(e)} + \tilde{w}_t. \tag{5.63}$$

Predicting future states according to (5.63) leads to notice deviations between what the radio is expecting and what it is actually measuring all the time and allows to calculate innovations at the observation level following:

$$\tilde{y}_t^{(e)} = \tilde{Z}_t^{(e)} - H\tilde{X}_t^{(e)}. \tag{5.64}$$

Then, the innovations projected on the GS level can define the Generalized Errors (GEs) in the following form:

$$\tilde{\mathcal{E}}_{\tilde{X}_t^{(e)}} = [\tilde{X}_{t-1}^{(e)}, P(\dot{\mathcal{E}}_{\tilde{X}_t^{(e)}})] = [\tilde{X}_t^{(e)}, H^{-1}\tilde{y}_t^{(e)}]. \tag{5.65}$$

The GEs are treated by the radio as self-information to discover the emergent dynamic rules present in the environment (acquire knowledge about the surroundings) and build up its own long term memory. In order to learn the GDBN model, the radio clusters in an unsupervised manner the GEs ($\tilde{\mathcal{E}}_{\tilde{X}_t^{(e)}}$) calculated during its first operation in the field as mentioned previously and used the GNG for that purpose. GNG outputs a set of Generalized Superstates (GSS) $\mathcal{V}_e = \{\tilde{S}_1^{(e)}, \tilde{S}_2^{(e)}, \ldots, \tilde{S}_N^{(e)}\}$ Since each GSS in $\mathcal{V}_e$ is assumed to follow a multivariate Gaussian distribution, it can be represented by its sufficient statistics, i.e., the generalized mean $\tilde{\mu}_{\tilde{S}_n^{(e)}} = [\mu_{\tilde{S}_n^{(e)}}, \dot{\mu}_{\tilde{S}_n^{(e)}}]$ and covariance matrix $\Sigma_{\tilde{S}_n^{(e)}}$ where $n \in \{1, 2, \ldots, N_e\}$. Analysing the signal's dynamic transitions among the learned clusters allows estimating the transition probabilities $\pi_{ij} = P(\tilde{S}_t^{(e)} = i | \tilde{S}_{t-1}^{(e)} = j)$ to learn the transition matrix $\Pi \in \mathbb{R}^{N_e \times N_e}$

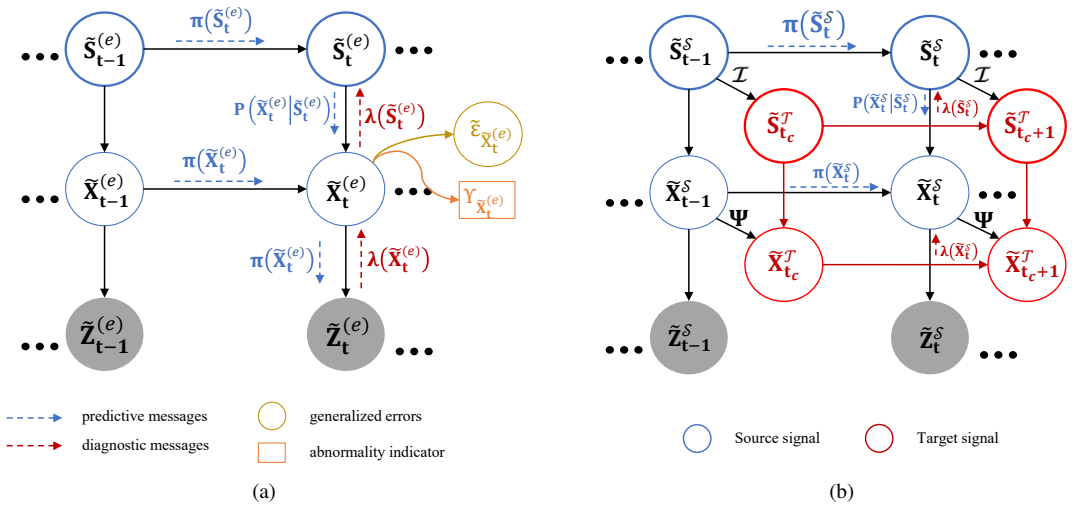


Fig. 5.19 The proposed graphical representations for joint prediction and conversion. (a) GDBN, (b) DGDBN.

defined as:

$$\Pi = \begin{bmatrix} \pi_{11} & \pi_{12} & \dots & \pi_{1N} \\ \pi_{21} & \pi_{22} & \dots & \pi_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{N1} & \pi_{N2} & \dots & \pi_{NN} \end{bmatrix}, \tag{5.66}$$

where $\sum_{j}^{\mathrm{N}_e} \pi_{ij} = 1$, such that, $i, j \in \mathrm{N}_e$. Furthermore, estimating the time-varying transition matrix denoted by Π_τ is of great interest due to the dynamic nature of the radio environment that varies with time, such that:

$$\Pi_\tau = \begin{bmatrix} \pi_{11,\tau} & \pi_{12,\tau} & \dots & \pi_{1N,\tau} \\ \pi_{21,\tau} & \pi_{22,\tau} & \dots & \pi_{2N,\tau} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{N1,\tau} & \pi_{N2,\tau} & \dots & \pi_{NN,\tau} \end{bmatrix}, \tag{5.67}$$

where $\pi_{ij,\tau} = \mathrm{P}(\tilde{S}_t^{(e)} = i | \tilde{S}_{t-1}^{(e)} = j, \tau)$ characterize a new condition on the transition from i to j which depends on the time elapsed (τ) in state i. The learned neurons (or GSS), along with their sufficient statistics and transition matrices (forming the so-called Vocabulary) can be represented in a Graph where vertices express the GSSs and edges express the transitions among vertices, as we will discuss in the following section.

The learning procedure shown in this section is repeated during different experiences involving multiple entities (e) transmitting the same information but adopting different modulation formats to learn their correspondent vocabularies represented as different graphs. Since the transmitted information is the same, then it can become helpful to provide the AI-enabled radio with the capability of translating the languages of the transmitted signals, e.g., translating a QPSK signal carrying an image into a 16QAM signal carrying the same image. In this sense, the information carried by the two signals is the same but represented in different languages (modulation schemes).

5.8.4 Graph Matching and Transport Plan Learning

Assume that we have two entities denoted as source ($\mathcal{S}$) and target ($\mathcal{T}$). These entities can be represented as two graphs, denoted as $\mathcal{G}_\mathcal{S} = (\mathcal{V}_\mathcal{S}, \mathrm{E}_\mathcal{S})$ and $\mathcal{G}_\mathcal{T} = (\mathcal{V}_\mathcal{T}, \mathrm{E}_\mathcal{T})$. $\mathcal{V}_\mathcal{S}$ and $\mathrm{V}_\mathcal{T}$ are a set of $\mathrm{N}_\mathcal{S}$ and $\mathrm{N}_\mathcal{T}$ vertices, each with a set $\mathrm{E}_\mathcal{S}$ and $\mathrm{E}_\mathcal{T}$ of edges. We assume that each graph is connected, directed and edge weighted. The graphs are obtained from different sources (signals with different modulation formats) and have different structures. Thus, we need to define a notion of similarity (distance) to compare them. The aim of this comparison is

to form a matching map (one-to-one or one-to-many) allowing to capture all the possible correspondences among the two graphs (i.e., among $\mathcal{V}_\mathcal{T}$ and $\mathcal{V}_\mathcal{S}$) and consequently to compute an optimal transport plan allowing to convert $\mathcal{G}_\mathcal{S}$ into $\mathcal{G}_\mathcal{T}$ (i.e., to transfer signals from $\mathcal{G}_\mathcal{S}$ to $\mathcal{G}_\mathcal{T}$). Following are the essential steps for learning the transport plan (refer to Fig. 5.20).

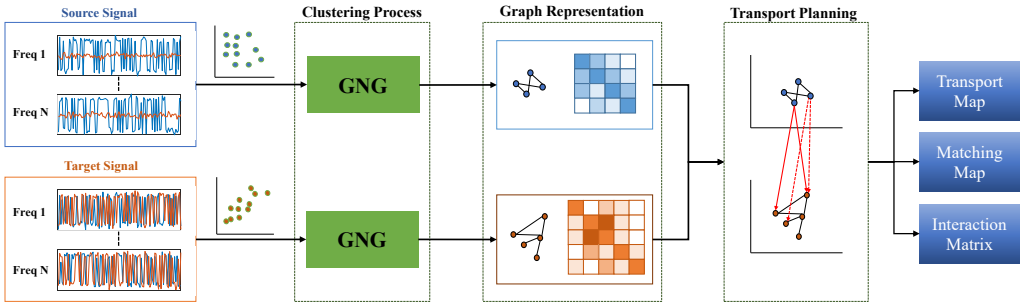


Fig. 5.20 Schematic illustrating the necessary steps to learn the transport plan.

Graph Matching. A possible approach to compare $\mathcal{G}_\mathcal{S}$ and $\mathcal{G}_\mathcal{T}$ is by calculating the divergence between the corresponding vertices of the two graphs encoded in sets $\mathcal{V}_\mathcal{S}$ and $\mathcal{V}_\mathcal{T}$. To this purpose we use the Kullback-Leibler divergence ($\mathcal{D}_{\mathcal{KL}}$) to measure how distant two distributions $\tilde{\mathrm{S}}_i^\mathcal{T} \in \mathcal{V}_\mathcal{S}$ and $\tilde{\mathrm{S}}_j^\mathcal{T} \in \mathcal{V}_\mathcal{T}$ are from each other according to:

$$\mathcal{D}_{\mathcal{KL}}(\tilde{\mathrm{S}}_i^\mathcal{S} || \tilde{\mathrm{S}}_j^\mathcal{T}) = \int \tilde{\mathrm{S}}_i^\mathcal{S} \log\left(\frac{\tilde{\mathrm{S}}_i^\mathcal{S}}{\tilde{\mathrm{S}}_j^\mathcal{T}}\right) dS. \tag{5.68}$$

However, $\mathcal{D}_{\mathcal{KL}}$ is not a real distance metric since it is not symmetric and does not satisfy the triangle inequality. Thus, we used a symmetric version of the $\mathcal{D}_{\mathcal{KL}}$ defined as follows:

$$\mathcal{D}_{\mathcal{SKL}}(\tilde{\mathrm{S}}_\mathrm{i}^\mathcal{S}, \tilde{\mathrm{S}}_\mathrm{j}^\mathcal{T}) = \mathcal{D}_{\mathcal{KL}}(\tilde{\mathrm{S}}_i^\mathcal{S} || \tilde{\mathrm{S}}_j^\mathcal{T}) + \mathcal{D}_{\mathcal{KL}}(\tilde{\mathrm{S}}_j^\mathcal{T} || \tilde{\mathrm{S}}_i^\mathcal{S}). \tag{5.69}$$

Since vertices of the two graphs represent a multivariate Gaussian distributions, we use the approximated $\mathcal{D}_{\mathcal{KL}}$ [252] defined as:

$$\mathcal{D}_{\mathcal{KL}}(\tilde{\mathrm{S}}_i^\mathcal{S} || \tilde{\mathrm{S}}_j^\mathcal{T}) = \frac{1}{2}\left(\ln\left(\frac{|\Sigma_{\tilde{\mathrm{S}}_j^\mathcal{T}}|}{|\Sigma_{\tilde{\mathrm{S}}_i^\mathcal{S}}|}\right) - d + \mathrm{tr}(\Sigma_{\tilde{\mathrm{S}}_j^\mathcal{T}}^{-1}\Sigma_{\tilde{\mathrm{S}}_i^\mathcal{S}}) + (\mu_{\tilde{\mathrm{S}}_i^\mathcal{S}} - \mu_{\tilde{\mathrm{S}}_j^\mathcal{T}})^\intercal \Sigma_{\tilde{\mathrm{S}}_j^\mathcal{T}}^{-1}(\mu_{\tilde{\mathrm{S}}_i^\mathcal{S}} - \mu_{\tilde{\mathrm{S}}_j^\mathcal{T}})\right). \tag{5.70}$$

Accordingly, we build a matching matrix $\mathfrak{M} \in \mathbb{R}^{\mathrm{N_S} \times \mathrm{N_T}}$ encoding all the $\mathrm{N_S} \times \mathrm{N_T}$ normalized distances $\bar{\mathcal{D}}(\tilde{\mathrm{S}}_{\mathrm{i}}^{\mathcal{S}}, \tilde{\mathrm{S}}_{\mathrm{j}}^{\mathcal{T}})$ and expressed as:

$$\mathfrak{M} = \begin{bmatrix} \bar{\mathcal{D}}(\tilde{\mathrm{S}}_1^{\mathcal{S}}, \tilde{\mathrm{S}}_1^{\mathcal{T}}) & \bar{\mathcal{D}}(\tilde{\mathrm{S}}_1^{\mathcal{S}}, \tilde{\mathrm{S}}_2^{\mathcal{T}}) & \dots & \bar{\mathcal{D}}(\tilde{\mathrm{S}}_1^{\mathcal{S}}, \tilde{\mathrm{S}}_{\mathrm{N_T}}^{\mathcal{T}}) \\ \bar{\mathcal{D}}(\tilde{\mathrm{S}}_2^{\mathcal{S}}, \tilde{\mathrm{S}}_1^{\mathcal{T}}) & \bar{\mathcal{D}}(\tilde{\mathrm{S}}_2^{\mathcal{S}}, \tilde{\mathrm{S}}_2^{\mathcal{T}}) & \dots & \bar{\mathcal{D}}(\tilde{\mathrm{S}}_2^{\mathcal{S}}, \tilde{\mathrm{S}}_{\mathrm{N_T}}^{\mathcal{T}}) \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\mathcal{D}}(\tilde{\mathrm{S}}_{\mathrm{N_S}}^{\mathcal{S}}, \tilde{\mathrm{S}}_1^{\mathcal{T}}) & \bar{\mathcal{D}}(\tilde{\mathrm{S}}_{\mathrm{N_S}}^{\mathcal{S}}, \tilde{\mathrm{S}}_2^{\mathcal{T}}) & \dots & \bar{\mathcal{D}}(\tilde{\mathrm{S}}_{\mathrm{N_S}}^{\mathcal{S}}, \tilde{\mathrm{S}}_{\mathrm{N_T}}^{\mathcal{T}}) \end{bmatrix}, \tag{5.71}$$

where

$$\bar{\mathcal{D}}(\tilde{\mathrm{S}}_{\mathrm{i}}^{\mathcal{S}}, \tilde{\mathrm{S}}_{\mathrm{j}}^{\mathcal{T}}) = \frac{\mathcal{D}_{\mathcal{SKL}}(\tilde{\mathrm{S}}_{\mathrm{i}}^{\mathcal{S}}, \tilde{\mathrm{S}}_{\mathrm{j}}^{\mathcal{T}})}{\sum_j^{\mathrm{N_T}} \mathcal{D}_{\mathcal{SKL}}(\tilde{\mathrm{S}}_{\mathrm{i}}^{\mathcal{S}}, \tilde{\mathrm{S}}_{\mathrm{j}}^{\mathcal{T}})}, \tag{5.72}$$

such that, $i = \{1, 2, \dots, \mathrm{N_S}\}$ and $j = \{1, 2, \dots, \mathrm{N_T}\}$.

Graph Interaction. While mapping a certain vertex in $\mathcal{G}_{\mathcal{S}}$ to the set of vertices in $\mathcal{G}_{\mathcal{T}}$ it might appear a 1-to-many mapping based on the similarity distance calculated in (5.70), i.e., the distance from 1-to-many is similar. Thus, if a certain source vertex in $\mathcal{G}_{\mathcal{S}}$ fires, it is more probable that one of the many vertices in $\mathcal{G}_{\mathcal{T}}$ will fire too. In order to capture the joint firing pattern we further integrate the time-varying interaction Matrix $\mathfrak{I}$ within the optimal map to track the vertex firing between all the possible pair of vertices among the two graphs and defined as:

$$\mathfrak{I} = \begin{bmatrix} \mathrm{P}(\tilde{\mathrm{S}}_1^{\mathcal{S}}, \tilde{\mathrm{S}}_1^{\mathcal{T}}) & \mathrm{P}(\tilde{\mathrm{S}}_1^{\mathcal{S}}, \tilde{\mathrm{S}}_2^{\mathcal{T}}) & \dots & \mathrm{P}(\tilde{\mathrm{S}}_1^{\mathcal{S}}, \tilde{\mathrm{S}}_{\mathrm{N_T}}^{\mathcal{T}}) \\ \mathrm{P}(\tilde{\mathrm{S}}_2^{\mathcal{S}}, \tilde{\mathrm{S}}_1^{\mathcal{T}}) & \mathrm{P}(\tilde{\mathrm{S}}_2^{\mathcal{S}}, \tilde{\mathrm{S}}_2^{\mathcal{T}}) & \dots & \mathrm{P}(\tilde{\mathrm{S}}_2^{\mathcal{S}}, \tilde{\mathrm{S}}_{\mathrm{N_T}}^{\mathcal{T}}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{P}(\tilde{\mathrm{S}}_{\mathrm{N_S}}^{\mathcal{S}}, \tilde{\mathrm{S}}_1^{\mathcal{T}}) & \mathrm{P}(\tilde{\mathrm{S}}_{\mathrm{N_S}}^{\mathcal{S}}, \tilde{\mathrm{S}}_2^{\mathcal{T}}) & \dots & \mathrm{P}(\tilde{\mathrm{S}}_{\mathrm{N_S}}^{\mathcal{S}}, \tilde{\mathrm{S}}_{\mathrm{N_T}}^{\mathcal{T}}) \end{bmatrix}, \tag{5.73}$$

where $\mathfrak{I} \in \mathbb{R}^{\mathrm{N_S} \times \mathrm{N_T}}$, $\mathrm{P}(\tilde{\mathrm{S}}_k^{\mathcal{S}}, \tilde{\mathrm{S}}_l^{\mathcal{T}})$ is the joint firing probability of vertices $\tilde{S}_k^{\mathcal{S}}$ and $\tilde{S}_l^{\mathcal{T}}$ and $\sum_l^{\mathrm{N_T}} \mathrm{P}(\tilde{\mathrm{S}}_k^{\mathcal{S}}, \tilde{\mathrm{S}}_l^{\mathcal{T}}) = 1$ such that $k \in \mathrm{N_S}$ and $l \in \mathrm{N_T}$.

Learning the Optimal Transport Plan. We aim to find a map $\Psi : \mathcal{G}_{\mathcal{S}} \to \mathcal{G}_{\mathcal{T}}$ which transports the mass from $\mathcal{V}_{\mathcal{S}}$ to $\mathcal{V}_{\mathcal{T}}$ while minimizing the mass transportation cost represented by the force needed to convert source data samples into target ones. Such required force is proportional to how much the vertices in the two graphs are distant from each other. It increases as the distance increase and viceversa. In this sense the optimal way of transporting the mass from $\mathcal{G}_{\mathcal{S}}$ to $\mathcal{G}_{\mathcal{T}}$ is to integrate the matching map encoded in (5.71) and the interaction matrix $\mathfrak{I}$ defined in (5.73) and thus transforming the one-to-many mapping into one-to-one mapping. Consequently, since we know the optimal mapping between the vertices of the two

graphs we can apply the proper transport map encoded in Ψ which is defined as follows:

$$\Psi = \begin{bmatrix} \mathcal{U}_{\tilde{S}_1^{\mathcal{S}},\tilde{S}_1^{\mathcal{T}}} & \mathcal{U}_{\tilde{S}_1^{\mathcal{S}},\tilde{S}_2^{\mathcal{T}}} & \dots & \mathcal{U}_{\tilde{S}_1^{\mathcal{S}},\tilde{S}_{N_{\mathcal{T}}}^{\mathcal{T}}} \\ \mathcal{U}_{\tilde{S}_2^{\mathcal{S}},\tilde{S}_1^{\mathcal{T}}} & \mathcal{U}_{\tilde{S}_2^{\mathcal{S}},\tilde{S}_2^{\mathcal{T}}} & \dots & \mathcal{U}_{\tilde{S}_2^{\mathcal{S}},\tilde{S}_{N_{\mathcal{T}}}^{\mathcal{T}}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{U}_{\tilde{S}_{N_{\mathcal{S}}}^{\mathcal{S}},\tilde{S}_1^{\mathcal{T}}} & \mathcal{U}_{\tilde{S}_{N_{\mathcal{S}}}^{\mathcal{S}},\tilde{S}_2^{\mathcal{T}}} & \dots & \mathcal{U}_{\tilde{S}_{N_{\mathcal{S}}}^{\mathcal{S}},\tilde{S}_{N_{\mathcal{T}}}^{\mathcal{T}}} \end{bmatrix}, \tag{5.74}$$

where $\mathcal{U}_{\tilde{S}_k^{\mathcal{S}}|\tilde{S}_l^{\mathcal{T}}}$ is the transport map encoding the transport force $\dot{\mu}_{\tilde{S}_k^{\mathcal{S}}|\tilde{S}_l^{\mathcal{T}}}$ and the transport uncertainty described by the covariance matrix $\mathcal{C}_{\tilde{S}_k^{\mathcal{S}}|\tilde{S}_l^{\mathcal{T}}}$, calculated in the following form:

$$\dot{\mu}_{\tilde{S}_k^{\mathcal{S}}|\tilde{S}_l^{\mathcal{T}}} = \mu_{\tilde{S}_l^{\mathcal{T}}} - \mu_{\tilde{S}_k^{\mathcal{S}}}, \tag{5.75}$$

$$\mathcal{C}_{\tilde{S}_k^{\mathcal{S}}|\tilde{S}_l^{\mathcal{T}}} = \mu_{\tilde{S}_l^{\mathcal{T}}\tilde{S}_k^{\mathcal{S}}} - \mu_{\tilde{S}_l^{\mathcal{T}}}\mu_{\tilde{S}_k^{\mathcal{S}}}. \tag{5.76}$$

Synchronizing $\mathcal{G}_{\mathcal{S}}$ and $\mathcal{G}_{\mathcal{T}}$ by applying temporal delay. As claimed in section 5.8.1, since source and target distributions (graphs) represent different modulation formats and the conversion follows low- to high-order manner, it is important to take into account of different symbol rates, so introducing some temporal delay in the transport process. The information transmitted by low-order modulation occupy more symbols with respect to higher order modulations. Thus, the time needed (i.e., the delay t_c) to perform the conversion can be calculated as a factor between source clusters ($N_{\mathcal{S}}$) and target clusters ($N_{\mathcal{T}}$) expressed as:

$$\gamma = \frac{\log_2(N_{\mathcal{T}})}{\log_2(N_{\mathcal{S}})}. \tag{5.77}$$

In this way, the radio understand that each $t_c = \gamma t$ it can convert the source distribution to the targeted one.

5.8.5 Joint RF Perception, Prediction and Conversion

The GDBN representation decomposes data with complex and non-linear dynamics into fragments that are explainable by simpler dynamical units. Switching dynamic systems as the Markov Jump Particle Filter (MJPF) [249] applied on the learned GDBN are capable of discovering the dynamical units and explain their switching behaviour. A Modified-MJPF (M-MPJF) is implemented here to perform joint predictions of GSs and GSSs by blending Particle Filter (PF) and Kalman Filter (KF) and providing various probabilistic inference modes (predictive and diagnostic) within Generalized Filtering. In the predictive inference mode each level of the proposed hierarchy holds predictions (beliefs) about the states of the

level below. Those beliefs are signaled via predictive messages ($\pi(\tilde{\mathrm{S}}_t^{\mathcal{S}})$,$\pi(\tilde{\mathrm{X}}_t^{\mathcal{S}})$) in a top-down manner where they are compared against the sensory responses, resulting in multi-level abnormality indicators and Generalized Errors (GEs). These GEs are then fed back via diagnostic messages ($\lambda(\tilde{\mathrm{S}}_t^{\mathcal{S}})$,$\lambda(\tilde{\mathrm{X}}_t^{\mathcal{S}})$) from bottom-to-up the hierarchy to update the beliefs and thus improve future predictions and minimize future GEs.

In M-MJPF (employed on Double-GDBN in Fig. 5.19(b)), conversion allows to use a model learned from a signal modulated using a specific scheme to predict and update signals carrying the same information but modulated in a different way.

First, PF draws Y equally weighted particles from the proposal distribution encoded in Π to perdict the GSSs of $\mathcal{G}_{\mathcal{S}}$. Each propagated particle $\tilde{S}_{t,y}^{\mathcal{S}}$ can be used to predict $\tilde{S}_{t,y}^{\mathcal{T}}$ (i.e., converting $\tilde{S}_{t,y}^{\mathcal{S}}$ to $\tilde{\mathrm{S}}_{t,y}^{\mathcal{T}}$) after reaching the necessary time t_c by using (5.73). Then, for each particle ($\tilde{S}_{t,y}^{\mathcal{S}}$) a KF is employed to predict the GS ($\tilde{\mathrm{X}}_{t,y}^{\mathcal{S}}$) of $\mathcal{G}_{\mathcal{S}}$ which depends on the predictions done at the level above as pointed out in (5.61). At this level converting $\tilde{\mathrm{X}}_{t,y}^{\mathcal{S}}$ to $\tilde{\mathrm{X}}_{t,y}^{\mathcal{T}}$ can be performed after reaching the required time t_c by using the transport map encoded in (5.74) and extracting the proper transport force and transport uncertainty to update the dynamic model defined in (5.61) according to:

$$\tilde{\mathrm{X}}_{t_c,y}^{\mathcal{T}} = \mathbb{E}\Big[\big(\mathrm{A}\tilde{\mathrm{X}}_{t_c,y}^{\mathcal{S}} + \mathrm{B}\big(\mathrm{U}_{\tilde{\mathrm{S}}_{t_c,y}^{\mathcal{T}}} + \dot{\mu}_{\tilde{\mathrm{S}}_{t_c,y}^{\mathcal{S}}|\tilde{\mathrm{S}}_{t_c,y}^{\mathcal{T}}}\big)\big), \big(\mathrm{A}\tilde{\mathrm{X}}_{t_{c-1},y}^{\mathcal{S}} + \mathrm{B}\big(\mathrm{U}_{\tilde{\mathrm{S}}_{t_{c-1},y}^{\mathcal{T}}} + \dot{\mu}_{\tilde{\mathrm{S}}_{t_{c-1},y}^{\mathcal{S}}|\tilde{\mathrm{S}}_{t_{c-1},y}^{\mathcal{T}}}\big)\big), \ldots, \big(\mathrm{A}\tilde{\mathrm{X}}_{t_{c-\gamma},y}^{\mathcal{S}} + \mathrm{B}\big(\mathrm{U}_{\tilde{\mathrm{S}}_{t_{c-\gamma},y}^{\mathcal{T}}} + \dot{\mu}_{\tilde{\mathrm{S}}_{t_{c-\gamma},y}^{\mathcal{S}}|\tilde{\mathrm{S}}_{t_{c-\gamma},y}^{\mathcal{T}}}\big)\big)\Big], \quad (5.78)$$

where $\mathbb{E}[.]$ depicts the mean value. The idea of conversion is that predicting the source signal's hidden states allows to predict and infer those of the target signal so that we can predict (generate) one signal from the other using the transport plan.
The posterior probability associated with the predicted GS ($\tilde{\mathrm{X}}_{t,y}^{\mathcal{S}}$) and GSS and propagated towards the bottom level is given by:

$$\pi(\tilde{\mathrm{X}}_{t,y}^{\mathcal{S}}) = \mathrm{P}(\tilde{\mathrm{X}}_{t,y}^{\mathcal{S}}, \tilde{\mathrm{S}}_{t,y}^{\mathcal{S}}|\tilde{Z}_{t-1}^{\mathcal{S}}) = \int \mathrm{P}(\tilde{\mathrm{X}}_{t,y}^{\mathcal{S}}|\tilde{\mathrm{X}}_{t-1,y}^{\mathcal{S}}, \tilde{\mathrm{S}}_{t,y}^{\mathcal{S}})\lambda(\tilde{\mathrm{X}}_{t-1,y}^{\mathcal{S}})d\tilde{\mathrm{X}}_{t-1,y}^{\mathcal{S}}, \quad (5.79)$$

where $\lambda(\tilde{\mathrm{X}}_{t-1,y}^{\mathcal{S}}) = \mathrm{P}(\tilde{Z}_{t-1}^{\mathcal{S}}|\tilde{\mathrm{X}}_{t-1,y}^{\mathcal{S}})$.

Accordingly, once a new sensory response $\tilde{Z}_t^{\mathcal{S}}$ is observed, diagnostic messages are fed upwards to update the posterior in the following way:

$$\mathrm{P}(\tilde{\mathrm{X}}_{t,y}^{\mathcal{S}}, \tilde{\mathrm{S}}_{t,y}^{\mathcal{S}}|\tilde{Z}_t^{\mathcal{S}}) = \pi(\tilde{\mathrm{X}}_{t,y}^{\mathcal{S}})\lambda(\tilde{\mathrm{X}}_{t,y}^{\mathcal{S}}). \quad (5.80)$$

Next, update the weights of the particles according to:

$$W_{t,y} = W_{t,y}\lambda(\tilde{\mathrm{S}}^{\mathcal{S}}_{t,y}), \tag{5.81}$$

and normalize by using the Sequential Importance Resampling (RIS). $\lambda(\tilde{\mathrm{S}}^{\mathcal{S}}_{t,y})$ is a discrete probability distribution represented by:

$$\lambda(\tilde{\mathrm{S}}^{\mathcal{S}}_{t,y}) = \lambda(\tilde{\mathrm{X}}^{\mathcal{S}}_{t,y})\mathrm{P}(\tilde{\mathrm{X}}^{\mathcal{S}}_{t,y}|\tilde{\mathrm{S}}^{\mathcal{S}}_{t,y}) = \mathrm{P}(\tilde{\mathrm{Z}}^{\mathcal{S}}_{t,y}|\tilde{\mathrm{X}}^{\mathcal{S}}_{t,y})\mathrm{P}(\tilde{\mathrm{X}}^{\mathcal{S}}_{t,y}|\tilde{\mathrm{S}}^{\mathcal{S}}_{t,y}). \tag{5.82}$$

5.8.6 Automatic Modulation Classification (AMC)

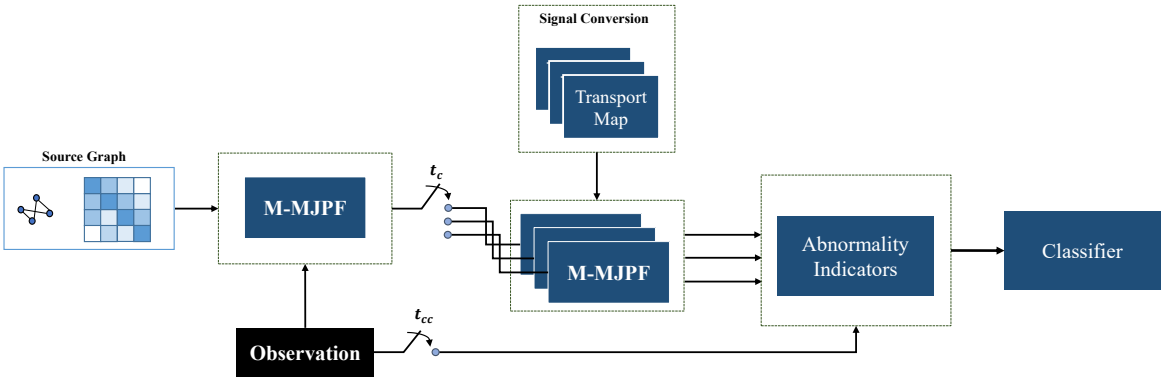


Fig. 5.21 Schematic illustrating the AMC process using Generalized Filtering and the learned transport plan.

In order to identify the correct modulation format (from the candidate set Θ) of the received sensory signal (i.e., the current observation), the radio performs multiple predictions in parallel based on the rules learned from experience so far and evaluate which of the predictions better explain the current situation (refer to Fig. 5.21). The candidate set $\Theta = \{\mathcal{G}_{\mathcal{S}_1}, \mathcal{G}_{\mathcal{T}_2}, \ldots, \mathcal{G}_{\mathcal{T}_K}\}$ contains one source modulation format and $K-1$ target modulation formats in ascending order where K is the total number of formats. The radio starts predicting the dynamics of the source signal using the source graph ($\mathcal{G}_{\mathcal{S}_1}$) that encodes the rules of how the signal's dynamics evolve according to the source modulation format. Then, by using the optimal transport plan learned during training it can transfer (convert) the predicted source signal into target signals representing the dynamics of the other modulation formats (target ones). Since the radio does not has any prior knowledge about the sensory signals regarding the data rate and the modulation scheme adopted, it is also essential to apply temporal delay (t_{cc}) and synchronize the multiple predictions during online classification. So, the time required to perform the classification task is related to the time t_{c_K} required converting the