A hybrid model-based and learning-based approach for classification using limited number of training samples

Alireza Nooraiepour, Waheed U. Bajwa, and Narayan B. Mandayam

***Abstract*—The fundamental task of classification given a limited number of training data samples is considered for physical systems with known parametric statistical models. The standalone learning-based and statistical model-based classifiers face major challenges towards the fulfillment of the classification task using a small training set. Specifically, classifiers that solely rely on the physics-based statistical models usually suffer from their inability to properly tune the underlying unobservable parameters, which leads to a mismatched representation of the system's behaviors. Learning-based classifiers, on the other hand, typically rely on a large number of training data from the underlying physical process, which might not be feasible in most practical scenarios. In this paper, a hybrid classification method—termed HyPhyLearn—is proposed that exploits both the physics-based statistical models and the learning-based classifiers. The proposed solution is based on the conjecture that HyPhyLearn would alleviate the challenges associated with the individual approaches of learning-based and statistical model-based classifiers by fusing their respective strengths. The proposed hybrid approach first estimates the unobservable model parameters using the available (suboptimal) statistical estimation procedures, and subsequently use the physics-based statistical models to generate *synthetic data*. Then, the training data samples are incorporated with the synthetic data in a learning-based classifier that is based on domain-adversarial training of neural networks. Specifically, in order to address the mismatch problem, the classifier learns a mapping from the training data and the synthetic data to a common feature space. Simultaneously, the classifier is trained to find discriminative features within this space in order to fulfill the classification task. Two case studies from communications systems (physical layer security and multi-user detection) are presented in order to highlight the usefulness of HyPhyLearn. Numerical results demonstrate that the proposed approach leads to major classification improvements in comparison to the existing standalone or hybrid classification methods.**

I. Introduction

We revisit the problem of classification with limited number of training data samples in this paper. The fundamental task of classification comes up in various fields and is traditionally tackled within two frameworks: 1) statistical setting, and 2) fully data-driven setting. In the first case, the main assumption is that data generation adheres to a known probabilistic model of the underlying physical process. Subsequently, the classification problem is usually dealt with within a hypothesis testing (HT) framework aimed at testing between two (or more) hypotheses. Here, optimality in both the Bayesian sense and the Neyman–Pearson sense relies on computation of the *likelihood-ratio* terms, which requires clairvoyant knowledge of the probabilistic models under different hypotheses [3]. However, accurate modeling of the physical processes in increasingly complex engineered systems is either not tractable or it relies on a large number of unobservable parameters, estimation of which from limited number of data samples could be a major hurdle [4], [5]. As a result, a mismatch between the physics-based statistical models and the real physical processes is inevitable. This precludes exact computation of the likelihood-ratio values, which deteriorates the classification performance [6]. The fully data-driven (i.e., learning based) setting, on the other hand, relies on a large number of data samples for finding an optimal mapping from the data samples to the corresponding labels. But availability of such data in many real-world problems, e.g., channel-based spoofing detection [7] and signal identification [8], is generally limited, which might lead to learning of a suboptimal map. Moreover, one should always expect mislabeled data in many applications, since the employed labeling procedures might not be error free. Consequently, classification performance of data-driven models can be seriously limited for many real-world applications.

The overarching objective of this paper is to develop an algorithmic framework for classification from limited number of training data samples in applications in which neither model-based nor learning-based approaches alone result in very good classification performance. To this end, note that learning-based approaches traditionally tend to disregard the physics-based models developed to describe the physical phenomena through tractable mathematical analysis. For instance, in the context of wireless communications, numerous theoretical models for channels and resource management have been developed over the years [4], [7], [9]. Despite being approximations in many cases, these models provide important prior information about the corresponding physical systems that might be utilized to facilitate the subsequent classification tasks. At the same time, physics-based models consist of numerous unobservable parameters, the tuning of which is a major hurdle for complex systems [5]. For example, physical channel models in the multi-input multi-output (MIMO) and 5G communications scenarios rely on a large number of multidimensional parameters that are defined over a mixed set of discrete and continuous spaces [10], [11]. In such cases, the maximum likelihood estimation (MLE) of the parameters could incur a formidable computational cost [11]–[13]. Our goal in this context is to develop a classification framework that can deal with these practical considerations through a hybrid approach that consolidates physics-based and fully data-driven classification approaches. The expectation is

The authors are with WINLAB, Department of Electrical and Computer Engineering, Rutgers University, NJ, USA. Emails: {alinoora,narayan}@winlab.rutgers.edu, waheed.bajwa@rutgers.edu.

This work was supported by the National Science Foundation (NSF) under grants ECCS-2028823 and OAC-1940074, and in part by ACI-1541069. A longer version of the current paper is available online in [1]. A preliminary version of this work was presented in 2021 IEEE Workshop on Machine Learning for Signal Processing (MLSP) [2].

that the hybrid approach would fuse the strengths of the two approaches towards achieving an overall superior classification performance.

Our proposed hybrid approach first employs the (necessarily) suboptimal parameter estimation methods to estimate the unobservable parameters. Then, it utilizes them in the physics-based models to generate *synthetic data*, which enables us to leverage learning-based classification approaches. The mismatch between the physics-based models and the underlying physical process is addressed in a learning setting. Specifically, a neural network is trained to map the training and synthetic data to a common discriminative feature space, which is often referred to as domain-invariant space in the domain adaptation literature [14], [15]. Meanwhile, a neural network-based classifier is trained on the mapped synthetic data to extract class-specific discriminative features from them. The resulting classifier in this way is expected to perform well on both synthetic and training data distributions.

A. Relation to prior works

In the realm of statistical model-based classifiers, the difficulties associated with estimating the parameters of the physics-based models are recognized in various works [6], [16]. This is mainly attributed to the inherent difficulties associated with determining probability distributions from only a limited number of data samples. Along these lines, classification under the assumption of mismatched models is considered in several works [6], [16]–[18]. Specifically, [16], [18] derive bounds on the probability of classification error in the presence of mismatch via the f-divergence between the true and mismatched distributions. In contrast to these bounds that are general in the sense that no assumption is made regarding the underlying distributions, [6] considers data that are contained in a linear subspace. This enables the authors to derive an upper bound on the classification error of the mismatched model that predicts the presence/absence of an error floor. The analyses in these works, however, do not lead to a classification algorithm for the mismatched setting as they merely analyze the mismatch problem itself.

The mismatch problem for the learning-based classifiers corresponds to the cases where the distribution of the available training data is different from that of the test data. Such mismatches are primarily studied in the transfer learning (TL) and the data-shift literature [15]. In particular, covariate shift [19], which is also studied under the name of transductive TL [20], refers to the case where the underlying data distributions for the test and training data are different. Concept shift [21], also known as inductive TL [20], on the other hand, deals with situations in which the posterior distribution of the labels given the data is not the same for the training and the test data. A wide range of algorithms have been proposed in order to alleviate the performance loss due to such shifts. For example, importance-weighting technique [22], [23] is proposed for the covariate shift scenario to remove the bias from the training data. Furthermore, algorithms based on subspace mapping [24] and learning domain-invariant representations [14] have also been proposed in the literature to address the mismatch problem. The authors in [24] propose a transfer component analysis method aimed at finding a transformation under which the maximum mean discrepancy between the true and mismatched distributions is small. The work in [14] aims at finding a representation that is invariant for the training and test distributions in order to mitigate the effect of discrepancies in the subsequent learning tasks. For the specific task of classification, the authors in [25] introduce the domain-adversarial neural network (DANN) framework, which extracts domain-invariant representations via (deep) neural networks that are discriminative for the training data in order to devise a classifier on the test data.

Deep transfer learning (DTL) is another prime subject related to our work that studies the transfer learning concept in the context of deep neural networks (DNNs). DTL considers a DNN that has been pre-trained on the training data as transferable knowledge useful for the test data. This knowledge can be transferred based on different strategies. The pre-trained DNNs can either be used directly for the test data, or serve as an intermediate feature extracting step that could facilitate the subsequent learning process for the test data. In another DTL strategy called *fine-tuning*, the pre-trained DNN or, certain parts of it, is refined using the available test data to further improve the effectiveness of transfer knowledge. We refer the reader to [26], [27] for a survey on DTL methods.

Model-based deep learning is another related line of work that aims at designing systems whose operation combines physics-based models (domain knowledge) and data. To this end, two main strategies are typically exploited in such works, known as model-aided networks and DNN-aided inference. The former results in specialized DNN architectures by identifying structures in a model-based algorithm; e.g., an iterative structure for the case of deep unfolding [28]. The latter primarily utilizes model-based methods for inference, but replaces explicit domain-specific computations with dedicated DNNs in order to facilitate operation in complex environments; e.g., using generative models for compressed sensing applications [29]. We refer the readers to [30] and references therein for the state-of-the-art strategies in model-based deep learning methods.

There also have been previous attempts to incorporate physics-inferred information in the fully data-driven setting. In the field of wireless communications, for instance, the authors in [4] employ DTL to solve a specific resource management problem. Similarly, the task of signal classification is tackled via DTL under different practical assumptions, such as real propagation effects [31], hardware impairments [32] and weak received signal strength [33]. These works utilize abundant data from an approximate model along with limited data from the real-world model in the DTL fine-tuning approach. More closely to the idea of physics-guided machine learning (ML), a recurrent neural network (RNN) is modified in [34] to incorporate information from the physics-based model as an internal state of the RNN. Furthermore, parameters of the physics-based models are combined with sensor readings and used as input to a DNN to develop a hybrid prognostics model in [5].

We note that the aforementioned works in domain adaptation literature do not employ any available physics-based statistical models and, consequently, rely on large number of training data

samples for dealing with the mismatch problem. In addition, model-based deep learning strategies might not be applicable to the statistical classification problem in general due to the lack of algorithmic structure such as an iterative structure. Equally importantly, DTL fine-tuning and physics-guided learning approaches do not consider the difficulties associated with estimating the physics-based parameters, which would indeed lead to inaccurate physics-based statistical models. The resulting discrepancy between the model and the underlying physical process necessitates a learning-based classifier that is capable of leveraging the data in a way to alleviate this mismatch problem.

B. Our contributions

The main contributions of this work are as follows.

- We focus on the task of classification for a physical process assuming that a limited number of training data samples, with possibly mislabeled instances, is available. We consider the case where the physical process (or its approximation) can be described by physics-based parametric statistical models. As these models tend to be complex in general, estimation of the unknown model parameters using the maximum likelihood estimation (MLE) procedure could be computationally prohibitive.[1] We instead propose HYPHYLEARN—a novel hybrid classification method—as a solution, which exploits both physics-based statistical models and learning-based classifiers. This approach makes use of (necessarily suboptimal) parameter estimation algorithms/heuristics to obtain (approximate) parameter estimates. Next, plugging in these estimates in the physics-based statistical models enables us to generate *synthetic* data. HYPHYLEARN then relies on neural networks (NNs), which are powerful tools for finding a discriminative feature space, towards obtaining a learning-based classifier. Specifically, the learning process involves training a NN to map the training and synthetic data to a common space under which they are not distinguishable. In the mean time, a learning-based classifier is trained on the synthetic data mapped to the new space to find discriminative class-level features. Indeed, learning the common feature space addresses the distribution mismatch problem between the training data samples and the generated synthetic data due to the errors in parameter estimation. It is then expected that the classifier trained on the mapped synthetic data will perform well on both data distributions. We repurpose theories from the domain adaptation literature based on learning invariant representations for our specific problem to justify the proposed hybrid approach. A schematic of HYPHYLEARN for a binary classification example is illustrated in Fig. 1.
- We also consider two prototypical problems from the wireless communications literature to investigate the performance of our proposed approach and show its superiority in comparison to the stand-alone statistical model-based classifiers as well as the fine-tuning approach as the best existing hybrid approach applicable to these problems. We first consider the problem of channel spoofing in the wireless communications setting, where an adversary (Eve) spoofs a legitimate transmitter (Alice) and sends a message to a legitimate receiver (Bob) [7], [36], [37]. The spoofing detection at Bob involves making a decision on whether an incoming message corresponds to Alice or Eve. This can be cast as a binary classification problem at Bob. Second, we revisit the problem of multi-user detection (MUD) in the uplink of a cellular network, where different users are asynchronously sharing a channel with a base station [13]. For a K-user system, MUD is basically a 2^K-ary classification problem in which the goal is to infer K binary information bits from a given observation. By obtaining likelihood ratio test (LRT) for each problem, we show that statistical model-based classifiers rely heavily on the wireless channel parameters in the above problems. However, estimation performance of these parameters suffers from both the paucity of training data and complexity of the physics-based statistical models. In fact, these models are complex in the sense that MLEs of the corresponding parameters require an exhaustive search over the space of the parameters, which is not feasible for many communication scenarios including MIMO transmissions in a 5G setting [10]. For both problems, numerical results show that HYPHYLEARN provides major improvements in terms of the classification accuracy in comparison to the best existing approaches.

[1] As discussed later in Section II, even using the MLE does not always provide any optimality guarantees in general for the classification problem in a HT setting [35].

C. Notation and organization

Throughout the paper, vectors are denoted with lowercase bold letters, while uppercase bold letters are reserved for matrices. Furthermore, equality by definition is expressed through the symbol $\triangleq$. Non-bold letters are used to denote scalar values and calligraphic letters denote sets. Furthermore, the cardinality of a set $\mathcal{S}$ is denoted by $|\mathcal{S}|$. The spaces of real and complex vectors of length d are denoted by $\mathbb{R}^d$ and $\mathbb{C}^d$, respectively. The mth element of a vector $\mathbf{u}$ and the trace of a matrix $\mathbf{U}$ are shown by $\mathbf{u}[m]$ and $\mathrm{Tr}(\mathbf{U})$, respectively. Also, real and imaginary parts of a complex number a are denoted by $\Re\{a\}$ and $\Im\{a\}$, respectively. The probability density function and expectation of a random variable w are denoted by $p(w)$ and $\mathbb{E}_p(w)$, respectively, while $\mathbb{P}[\cdot]$ is used to denote the probability of an event. The Gaussian and circularly-symmetric complex Gaussian distributions are denoted by $\mathcal{N}$ and $\mathcal{CN}$, respectively, while the uniform distribution supported between two real numbers a and b is denoted by $\mathrm{unif}(a,b)$. We denote the kth standard basis vector of length N in $\mathbb{R}^N$ by $\mathbf{e}_k$, and use $\|\mathbf{u}\|$ to refer to the Euclidean norm of the vector $\mathbf{u}$. We refer to identity matrix of size N and the indicator function by $\mathbf{I}_N$ and $\mathbb{1}_{\mathcal{A}}(\mathbf{x}) \triangleq \begin{cases} 1, \mathbf{x} \in \mathcal{A} \\ 0, \mathbf{x} \notin \mathcal{A} \end{cases}$, respectively. Transpose and conjugate transpose of $\mathbf{u}$ are denoted by $\mathbf{u}^T$ and $\mathbf{u}^H$, respectively. Furthermore, $\mathbf{e}_n(y)$ refers to a *one-hot* encoded

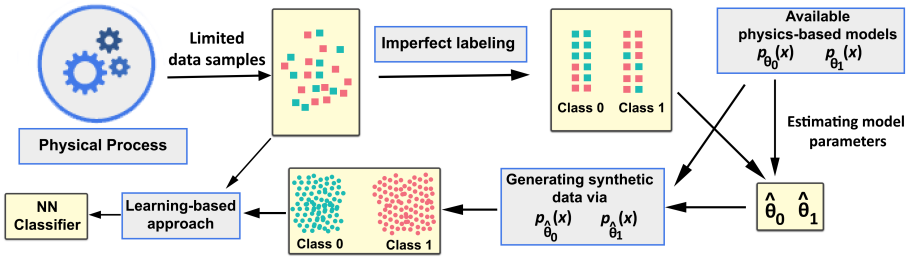


Fig. 1: A schematic of our proposed hybrid classification approach (HYPHYLEARN) illustrated for a binary classification setting, which exploits both physics-based statistical models and learning-based classifiers.

version of a non-negative integer y, which equals to an all-zero vector of length n except for the yth element which is set to 1. Also, $\circ$ and $\odot$ denote the Schur componentwise and the Khatri-Rao product, respectively, while $\otimes$ is reserved for the Kronecker product. Finally, given two vectors $\mathbf{a}$ and $\mathbf{b}$ of length M, Toeplitz matrix of size $M \times M$ is defined as

$$\text{toep}(\mathbf{a},\mathbf{b}) \triangleq \begin{bmatrix} \mathbf{a}[1] & \mathbf{b}[2] & \dots & \mathbf{b}[M] \\ \mathbf{a}[2] & \ddots & \ddots & \mathbf{b}[M-1] \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{a}[M] & \mathbf{a}[M-1] & \dots & \mathbf{a}[1] \end{bmatrix}.$$

The rest of the paper is organized as follows. The problem is formally posed in Section II. Our proposed solution is described in Section III, which discusses various pieces of HYPHYLEARN approach. We introduce the first case study involving the spoofing detection problem in Section IV. The second case study, which concerns the multi-user detection problem, is presented in Section V. We present numerical results concerning the application of our proposed approach in the above two case studies in Section VI, and contrast it with the existing methods. Finally, the paper is concluded in Section VII.

II. Problem Formulation

Consider a physical process consisting of C distinct behaviors where the physics-based parametric statistical model for the ith behavior is available in the form of a parametric probability density function (PDF) denoted by the conditional prior $p_i(\mathbf{x};\boldsymbol{\theta}_i)$ on observations $\mathbf{x}$ that belong to an observation space $\mathcal{X}$. Assuming the true underlying parameter for the ith behavior is $\boldsymbol{\theta}_i^*$, the data for this behavior is generated by drawing independent and identically distributed (i.i.d.) samples from $p_i(\mathbf{x};\boldsymbol{\theta}_i^*)$. Assuming further that the ith behavior is chosen with a prior probability π_i, our goal is to devise a decision rule to determine a given sample $\mathbf{x} = [x_1,\dots,x_n]^T$ is generated under which behavior. Clearly, this can be cast as a C-ary classification problem via $H_i : \mathbf{x} \sim p_i(\mathbf{x};\boldsymbol{\theta}_i^*),\ i = 0,\dots,C-1$. We consider the case where this decision is made by a classifier $h_{\boldsymbol{\phi}}(\cdot)$ parameterized by $\boldsymbol{\phi} \in \mathbb{R}^d$, $h_{\boldsymbol{\phi}}(\mathbf{x}) : \mathcal{X} \to \{0,\dots,C-1\}$, which partitions $\mathcal{X}$ into C disjoint sets, $\{\mathcal{X}_i\}$, and decides in favor of H_i if $\mathbf{x} \in \mathcal{X}_i$. Defining $\boldsymbol{\theta}^* \triangleq [\boldsymbol{\theta}_0^*,\dots,\boldsymbol{\theta}_{C-1}^*]$, we denote the probability of error associated with $h_{\boldsymbol{\phi}}(\mathbf{x})$ by $\mathbb{P}_{\boldsymbol{\theta}^*}[e_{\boldsymbol{\phi}}]$, which can be computed as

$$\mathbb{P}_{\boldsymbol{\theta}^*}[e_{\boldsymbol{\phi}}] = \sum_{i=0}^{C-1} \pi_i \int_{\mathcal{X}} p_i(\mathbf{x};\boldsymbol{\theta}_i^*) \mathbb{1}_{\{h_{\boldsymbol{\phi}}(\mathbf{x}) \neq i\}}(\mathbf{x}) d\mathbf{x}, \quad (1)$$

where $e_{\boldsymbol{\phi}}$ indicates the event that $h_{\boldsymbol{\phi}}(\mathbf{x})$ makes an erroneous decision. The optimal classifier $h_{\boldsymbol{\phi}^*}(\mathbf{x})$ that minimizes the error probability is given by the Bayes decision rule, i.e., $h_{\phi^*}(\mathbf{x}) = \text{argmax}_{i=0,\dots,C-1}\ \pi_i p_i(\mathbf{x};\boldsymbol{\theta}_i^*)$ [3]. For the specific case of $C = 2$, this rule takes the famous form of the likelihood ratio test, $\frac{p_1(\mathbf{x};\boldsymbol{\theta}_1^*)}{p_0(\mathbf{x};\boldsymbol{\theta}_0^*)} \underset{y=0}{\overset{y=1}{\gtrless}} \frac{\pi_0}{\pi_1}$, where $y = i$ implies making a decision in favor of the ith behavior.

We focus in this paper on the case where although the parametric model $p_i(\mathbf{x};\boldsymbol{\theta}_i)$ is known for the ith behavior, one does not have access to the corresponding underlying true parameter $\boldsymbol{\theta}_i^*$. Instead, only a small number of training data generated in an i.i.d. manner from $p_i(\mathbf{x};\boldsymbol{\theta}_i^*), \forall i$, are available. Specifically, we denote the available dataset by $\mathcal{D}_r = \{\mathbf{x}_{r,n}\}_{n=1}^{N_r}$, where N_r is the total number of data samples. Also, the corresponding ground-truth label for the nth sample is denoted by $y_{r,n}$ which is only given for $N_{r,l}$ number of data samples where $N_{r,l} \leq N_r$. Furthermore, we consider the case where the model $p_i(\mathbf{x};\boldsymbol{\theta}_i)$ under the ith behavior is a non-trivial function of the underlying parameter for which conventional estimation procedures such as maximum likelihood estimation (MLE) are either not available or are computationally prohibitive to implement. The implication of this aspect of the problem formulation is that the performance of any suboptimal parameter estimation method is bound to be limited. As a result, statistical model-based classifiers, which plug-in these estimates in $p_i(\mathbf{x};\boldsymbol{\theta}_i)$, would have a deteriorated performance as well.

Unlike these classifiers that rely heavily on the knowledge of the parametric statistical models and the estimated parameters, a purely data-driven approach can result in a classifier that disregards the available parametric models. However, as the data generation processes are governed by non-trivial models, a large number of data is needed in this case to extract related patterns from each behavior that would lead to a highly discriminative feature space. By noting that the performance of the fully data-driven and the statistical model-based classifiers is particularly curbed when they are used in a stand-alone fashion, we conjecture that fusing the strengths of the two can lead to a superior classification algorithm in our setting, as described in the next section.

Before delving into the proposed solution for the described problem setting, we discuss further two existing approaches towards obtaining a statistical model-based classifier for the benefit of the reader. Recall that within the framework of

statistical model-based classification, one would first estimate the unknown model parameters as $\widehat{\boldsymbol{\theta}}_i$'s, $i = 1, \ldots, C$, and plug them in the available models to obtain $p_i(\mathbf{x}; \widehat{\boldsymbol{\theta}}_i)$. The resulting plug-in models are then used in practice in lieu of the true models within the optimal Bayes decision rule. The parameters, $\boldsymbol{\phi}$, of the resulting *plug-in* classifier consist solely of the parameters of physics-based statistical models, i.e., $\boldsymbol{\phi} = \boldsymbol{\theta} = [\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_{C-1}]$.[2] Based on this fact, we denote the *plug-in* classifier by $h_{\boldsymbol{\theta}}(\mathbf{x})$ in the remainder of this section. The unknown model parameters can be estimated using numerous approaches. In the following, we discuss two of the most popular ways to estimate them as well as the shortcomings of these approaches that warrant a new approach to classification.

Empirical error minimizer: Given a set of training data with their corresponding labels, $\{\mathbf{x}_{r,n}, y_{r,n}\}_{n=1}^{N_r}$, the most natural approach for parameter estimation corresponds to the setting in which the resulting plug-in classifier, $h_{\boldsymbol{\theta}}(\mathbf{x})$, minimizes the *empirical* error probability defined by $\widehat{\mathbb{P}}^{N_r}[e_{\boldsymbol{\theta}}] \triangleq \frac{1}{N_r}\sum_{n=1}^{N_r} \mathbb{1}_{\{h_{\boldsymbol{\theta}}(\mathbf{x}_{r,n}) \neq y_{r,n}\}}$. Specifically, for the case of $C = 2$ consider the family of the classifiers $h_{\boldsymbol{\theta}}(\mathbf{x}) = \begin{cases} 0, & \pi p_{\boldsymbol{\theta}_0}(\mathbf{x}) > (1-\pi) p_{\boldsymbol{\theta}_1}(\mathbf{x}), \\ 1, & \text{otherwise}, \end{cases}$ for which the parameter values $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ are chosen from a space $\boldsymbol{\Theta}$. The parameter estimates that minimize the empirical error are obtained as $\widehat{\boldsymbol{\theta}} = [\widehat{\boldsymbol{\theta}}_0, \widehat{\boldsymbol{\theta}}_1] \in \operatorname{argmin}_{\boldsymbol{\theta}} \widehat{\mathbb{P}}^{N_r}[e_{\boldsymbol{\theta}}]$. The following lemma, which is a direct result of Corollary 16.1 in [38], presents an upper bound on the performance of the Bayes decision rule in terms of that of the plug-in classifier that is obtained using empirical error minimization.

Lemma 1. *If $\boldsymbol{\theta}_0^*,\ \boldsymbol{\theta}_1^* \in \boldsymbol{\Theta}$, then the error probability of the Bayes decision rule, with the probability at least $1 - \delta$, is bounded by*

$$\mathbb{P}_{\boldsymbol{\theta}^*}[e_{\boldsymbol{\theta}^*}] \leq \widehat{\mathbb{P}}^{N_r}[e_{\widehat{\boldsymbol{\theta}}}] + 8\sqrt{\frac{2}{N_r}\log\frac{8b}{\delta}}, \tag{2}$$

where b denotes the Vapnik–Chervonenkis (VC) dimension [38] of the family of classifiers, $h_{\boldsymbol{\theta}}(\mathbf{x})$, defined above.

The above lemma guarantees a $\mathcal{O}(\sqrt{\log N_r / N_r})$ rate of convergence to the Bayes error for $h_{\hat{\boldsymbol{\theta}}}(\mathbf{x})$ when $\widehat{\boldsymbol{\theta}}$ is chosen to minimize the empirical error. However, obtaining such $\widehat{\boldsymbol{\theta}}$ is computationally expensive in general as the empirical error probability might be a non-trivial function of the parameters.

Maximum likelihood estimator: In practice, the unknown model parameters are commonly replaced with their corresponding MLEs under each beahvior; the resulting plug-in classifier gives rise to the well-known generalized likelihood ratio test (GLRT) for the binary case ($C = 2$) [3]. Specifically, assuming the training data and their corresponding labels are available in the form of $\{\mathbf{x}_{r,n}, y_{r,n}\}_{n=1}^{N_i}$ for the ith hypothesis, the MLE of $\boldsymbol{\theta}_i$ is obtained by $\widehat{\boldsymbol{\theta}}_i^{MLE} = \operatorname{argmax}_{\boldsymbol{\theta}_i} \mathcal{L}(\mathcal{D}_i|\boldsymbol{\theta}_i)$, where $\mathcal{L}$ denotes the likelihood function. For the binary case where $\frac{\pi p_1(\mathbf{x};\boldsymbol{\theta}_1)}{(1-\pi)p_0(\mathbf{x};\boldsymbol{\theta}_0)+\pi p_1(\mathbf{x};\boldsymbol{\theta}_1)}$ is continuous in $(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \pi)$, as the parameters' estimates converge to the true values, the error of the plug-in classifier also converges to that of the Bayes decision rule. However, not only no optimality condition can be stated in general for the plug-in classifier relying on MLEs [35], obtaining such estimates might also be computationally prohibitive for system with complex likelihood functions.

[2]For notational simplicity and without loss of generality, we have not included the priors as part of the unknown parameters in the current discussion.

III. Proposed Solution: HyPhyLearn

The main deciding factor in superiority of a solution for the problem setup introduced in Section II is the extent to which it exploits the available information, i.e., training data and the parametric statistical models. In particular, the plug-in classifiers tend not to exploit this information in the most optimal fashion as performance of the parameter estimation procedures can be curbed due to the complexity of the underlying models and lack of the corresponding ground-truth labels. We instead propose a novel hybrid classification method to make use of the available information in learning-based classifiers, which are powerful tools for finding discriminative feature spaces. Specifically, our proposed solution relies on the parametric models to generate synthetic data and incorporate them with the training data in a classifier that makes use of adversarial training between NNs. Next, we describe the various steps of the proposed solution that is termed HyPhyLearn in detail.

Step 1—Imperfect labeling: As the available data are not assumed completely labeled in our problem setup, the first step in our solution deals with assigning labels to the unlabeled data samples in $\mathcal{D}_r$. This involves a clustering step that partitions the dataset $\mathcal{D}_r$ into C distinct groups. Then, the groups are labeled using the available $N_{r,l}$ labels. For example, a label can be assigned to a group based on the number of labeled training data it includes from each behavior; If the majority of such samples corresponds to the ith behavior, the group is labeled as i. Subsequently, we refer to a group assigned with the label i by $\mathcal{D}_{r,i}$ for $i = 0, \ldots, C-1$. Denoting this imperfect labeling process by $g(\mathbf{x}) : \mathcal{X} \to \{0, \ldots, C-1\}$, a non-trivial labeling error over $\mathcal{D}_r$ is associated with $g(\mathbf{x})$ that can be computed via $e_r = \frac{1}{N_r}\sum_{n=1}^{N_r} \mathbb{1}_{\{g(\mathbf{x}_{r,n}) \neq y_{r,n}\}}$. In the remainder of this paper, we refer to the number of samples in the cluster labeled as i by $N_{r,i}$. The function $g(\mathbf{x})$ may be obtained based on any one of the simple clustering algorithms from the ML literature, such as the Gaussian mixture model [39], or it may be a decision rule obtained based on the statistical analysis of the parametric models. For instance, for the problem of channel spoofing detection, a hypothesis test is proposed in [7] that assigns labels to unlabeled samples based on their similarity, measured in terms of the Euclidean distance, to a reference data sample.

Step 2—Parameter estimation: Based on the labels assigned in Step 1 to the unlabeled data samples, we estimate the parameters of the physics-based statistical models under each behavior. To this end, we utilize $\mathcal{D}_{r,i}$ to estimate the parameter vector $\boldsymbol{\theta}_i^*$ corresponding to the ith behavior. Furthermore, the priors are estimated as $\widehat{\pi}_i = N_{r,i}/N_r$. We note that the procedure for estimating $\boldsymbol{\theta}_i^*$ depends on the available parametric models corresponding to the ith behavior, i.e., $p_i(\mathbf{x}; \boldsymbol{\theta}_i)$. We recall from our problem setup that the MLE, which is usually utilized for parameter estimation purposes, might not be employed here due to the formidable complexity of optimizing

$p_i(\mathbf{x};\boldsymbol{\theta}_i)$ over $\boldsymbol{\theta}_i$. Instead, a (necessarily) suboptimal estimator, $T(\cdot)$, built upon either heuristics or optimization techniques like alternate maximization (see Sections IV-C and V-B) could be utilized to estimate the parameters as $\widehat{\boldsymbol{\theta}}_i = T(\mathcal{D}_{r,i})$ for all the behaviors. The parameter estimation performance is therefore limited here due to both the suboptimality of $T(\cdot)$ and presence of the mislabeled samples in $\mathcal{D}_{r,i}, \forall i$.

Step 3—Forming a synthetic dataset: The paucity of available data in our problem formulation seems to preclude utilization of a learning-based classifier as part of the solution. However, we note that the available physics-based statistical models, in the form of parametric PDFs, enable us to generate synthetic data to augment the available data, and make it possible to exploit the discriminative power of learning-based classifiers. Having access to the estimated parameter $\widehat{\boldsymbol{\theta}}_i$ obtained in Step 2, we plug it in the available physics-based statistical model to obtain a PDF $p_i(\mathbf{x};\widehat{\boldsymbol{\theta}}_i)$ for the ith behavior. In order to generate a synthetic dataset, we first sample w from a categorical distribution parameterized by $\widehat{\boldsymbol{\pi}} = [\widehat{\pi}_0, \ldots, \widehat{\pi}_{C-1}]$ over the sample space of $\{0, \ldots, C-1\}$. Then, we sample a data point $\mathbf{x}_{s,i}$ according to $\mathbf{x}_{s,i} \sim p_w(\mathbf{x};\widehat{\boldsymbol{\theta}}_w)$ with the associated label $y_{s,i} = w$. Repeating this process N_s number of times, we obtain a synthetic dataset $\mathcal{D}_s = \{\mathbf{x}_{s,i}, y_{s,i}\}_{i=1}^{N_s}$ in which the data samples are generated in a statistically independent fashion.

Step 4—Incorporating synthetic and training data in a learning-based classifier: The synthetic data generated in Step 3, besides retaining essential information about the underlying physics-based statistical models, enables us to utilize the discriminative power of learning-based classifiers. However, the errors introduced during the labeling and the parameter estimation steps that precede the synthetic data generation process incur a mismatch between the distributions corresponding to the training and synthetic datasets. This mismatch is bound to deteriorate the performance of a classifier trained on the synthetic data alone, when utilized in a real-world setting. Then the question is how a learning-based classifier can be trained to alleviate this problem. For example, in the fine-tuning approach [4], a NN-based classifier will be trained on the synthetic data first, and then, training data are used to refine the weights of the corresponding NN. However, we conjecture that such learning strategies that utilize the training and synthetic data in the separate stages of training are not the best solution here; rather, synthetic and training data should jointly be incorporated in a learning-based classifier. To this end, inspired by the works in the domain-adaptation literature and specifically feature space mapping [14], we propose to map the synthetic and training data through a data-driven function $M_{\boldsymbol{\psi}} : \mathcal{X} \to \mathcal{Z}$, which is parameterized by a real vector $\boldsymbol{\psi}$, into a common feature space $\mathcal{Z}$. Consequently, a classifier $h_{\boldsymbol{\phi}_1}(\mathbf{z})$, parameterized by $\boldsymbol{\phi}_1$, which is trained on the synthetic data within the space $\mathcal{Z}$ is expected to perform well on both training and synthetic data. To this end, we choose $M_{\boldsymbol{\psi}}$ and $h_{\boldsymbol{\phi}_1}$ to be NNs, which are powerful tools for finding discriminative features from a given dataset. We discuss this step in detail in the following subsection.

HYPHYLEARN: We now present our final solution as an algorithmic framework composed of the aforementioned four steps. In a nutshell, HYPHYLEARN generates synthetic data based on the physics-based parametric statistical models and utilizes them along with the available data in a learning-based classifier powered from the adversarial training of the NNs (see the following subsection). In order to train the NNs based on their specific loss functions, described in the following subsection, we utilize the stochastic gradient descent method [39] along with mini-batches consisting of random samples from the training and synthetic datasets in an iterative manner. The details of the whole process is presented in Algorithm 1.

Algorithm 1: HYPHYLEARN

Input: Parametric models $p_i(\mathbf{x};\boldsymbol{\theta}_i)$ $(i = 0, \ldots, C-1)$; Training dataset $\mathcal{D}_r = \{\mathbf{x}_{r,n}\}_{n=1}^{N_r}$; learning rates μ_{r_1}, μ_{r_2}, μ_{r_3}; Number of training steps N_{tr}; Mini-batch size $N_b < N_r$; Number of synthetic data samples N_s to be generated

Output: The mapping $M_{\boldsymbol{\psi}}(\cdot)$ and the classifier $h_{\boldsymbol{\phi}_1}(\cdot)$, parameterized by the real vectors $\boldsymbol{\psi}$ and $\boldsymbol{\phi}_1$, respectively

`// Step 1 - Imperfect labeling`

$\{\mathcal{D}_{r,0}, \ldots, \mathcal{D}_{r,C-1}\} \leftarrow$ Applying $g(\mathbf{x})$ to unlabeled samples

`// Step 2 - Parameter estimation`

$\widehat{\boldsymbol{\theta}}_i \leftarrow T_i(\mathcal{D}_{r,i})$, $\widehat{\pi}_i \leftarrow \frac{|\mathcal{D}_{r,i}|}{N_r}$ for $i = 0, \ldots, C-1$

`// Step 3 - Forming a synthetic dataset`

$p_i(\mathbf{x};\widehat{\boldsymbol{\theta}}_i) \leftarrow$Plug $\widehat{\boldsymbol{\theta}}_i$ in $p_i(\mathbf{x};\theta_i)$ for $i = 0, \ldots, C-1$

for $n = 1$ ***to*** N_s **do**

  `// Choosing a behavior`

  $r \sim \text{unif}(0,1)$, $w = \text{argmin}_k \sum_{i=0}^{k-1} \widehat{\pi}_i \geq r$

  `// Synthetic data generation`

  $\mathbf{x}_{s,n} \sim p_w(\mathbf{x};\widehat{\boldsymbol{\theta}}_w)$, $y_{s,n} = w$

  Add $\{\mathbf{x}_{s,n}, y_{s,n}\}$ to $\mathcal{D}_s$

end

`// Step 4 - Training the learning-based classifier`

for $n_{tr} = 1$ ***to*** N_{tr} **do**

  $\mathcal{D}_{r,b} \leftarrow N_b$ random samples from $\mathcal{D}_r$, $\mathcal{D}_{s,b} \leftarrow N_b$ random samples from $\mathcal{D}_s$

  `// Forward propagation via (12), (14)`

  $L_s \leftarrow \mathcal{L}_s(\boldsymbol{\psi}, \boldsymbol{\phi}_1 | \mathcal{D}_{s,b})$

  $L_c \leftarrow \mathcal{L}_c(\boldsymbol{\psi}, \boldsymbol{\zeta} | \mathcal{D}_{r,b}, \mathcal{D}_{s,b})$

  `// Backward propagation`

  Computing gradients: $\mathcal{G}_{s,\boldsymbol{\phi}_1} \leftarrow \nabla_{\boldsymbol{\phi}_1} L_s$, $\mathcal{G}_{s,\boldsymbol{\psi}} \leftarrow \nabla_{\boldsymbol{\psi}} L_s$

  Computing gradients: $\mathcal{G}_{c,\boldsymbol{\zeta}} \leftarrow \nabla_{\boldsymbol{\zeta}} L_c$, $\mathcal{G}_{c,\boldsymbol{\psi}} \leftarrow \nabla_{\boldsymbol{\psi}} L_c$

  `// Update network parameters via (15)`

  $\boldsymbol{\psi} \leftarrow \boldsymbol{\psi} - \mu_{r_1}(\mathcal{G}_{s,\boldsymbol{\psi}} - \mathcal{G}_{c,\boldsymbol{\psi}})$, $\boldsymbol{\phi}_1 \leftarrow \boldsymbol{\phi}_1 - \mu_{r_2}\mathcal{G}_{s,\boldsymbol{\phi}_1}$, $\boldsymbol{\zeta} \leftarrow \boldsymbol{\zeta} - \mu_{r_3}\mathcal{G}_{c,\boldsymbol{\zeta}}$

end

A. Incorporating synthetic and training data in a learning-based classifier for HYPHYLEARN

To elaborate further on Step 4, we first denote the distributions corresponding to the real and synthetic data as $p_{\boldsymbol{\theta}^*}(\mathbf{x}) = \sum_{i=0}^{C-1} \pi_i p_i(\mathbf{x};\boldsymbol{\theta}_i^*)$ and $p_{\widehat{\boldsymbol{\theta}}}(\mathbf{x}) = \sum_{i=0}^{C-1} \widehat{\pi}_i p_i(\mathbf{x};\widehat{\boldsymbol{\theta}}_i)$, respectively. We refer to $p_{\boldsymbol{\theta}^*}(\mathbf{x})$ and $p_{\widehat{\boldsymbol{\theta}}}(\mathbf{x})$ as the true and estimated distributions, respectively. For each distribution, applying the mapping $M_{\boldsymbol{\psi}}(\cdot)$ to the input space $\mathcal{X}$ would induce a distribution over the feature space $\mathcal{Z}$. Specifically, we denote the mapping of the true distribution $p_{\boldsymbol{\theta}^*}(\mathbf{x})$ to $\mathcal{Z}$ by $p_{\boldsymbol{\psi},\boldsymbol{\theta}^*}(\mathbf{z})$, where $\mathbf{z} = M_{\boldsymbol{\psi}}(\mathbf{x})$, $\mathbf{x} \sim p_{\boldsymbol{\theta}^*}(\mathbf{x})$. Assuming that $\mathcal{X}$

and $\mathcal{Z}$ are topological spaces, for any $\mathcal{A} \subset \mathcal{Z}$ the probability of $\mathcal{A}$ in space $\mathcal{Z}$ is

$$\mathbb{P}_{\mathbf{z}}[\mathcal{A}] \triangleq \mathbb{P}_{\mathbf{x}}\big[M_{\boldsymbol{\psi}}^{-1}(\mathcal{A})\big] = \sum_{i=0}^{C-1} \pi_i \int_{M_{\boldsymbol{\psi}}^{-1}(\mathcal{A})} p_i(\mathbf{x}; \boldsymbol{\theta}_i^*) d\mathbf{x}, \quad (3)$$

where the pre-image $M_{\boldsymbol{\psi}}^{-1}(\mathcal{A})$ belongs to the Borel σ-algebra over $\mathcal{X}$. Subsequently, the probability of error corresponding to a classifier $h_{\boldsymbol{\phi}_1}(\mathbf{z})$, parameterized by a real vector $\boldsymbol{\phi}_1$, with respect to the mapping of the true distribution to the $\mathcal{Z}$ space is computed via

$$\mathbb{P}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}[e_{\boldsymbol{\phi}_1}] = \sum_{i=0}^{C-1} \pi_i \int_{\mathcal{Z}} p_{\boldsymbol{\psi},\boldsymbol{\theta}_i^*}(\mathbf{z}) \mathbb{1}_{\{h_{\boldsymbol{\phi}_1}(\mathbf{z}) \neq i\}}(\mathbf{z}) d\mathbf{z}, \quad (4)$$

where the dependence of $\mathbb{P}$ on π_i's is suppressed for notational simplicity. Similarly, mapping of the estimated distribution to the space $\mathcal{Z}$ is characterized by a distribution denoted by $p_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}(\mathbf{z})$. Furthermore, the probability of error for a classifier $h_{\boldsymbol{\phi}_1}(\mathbf{z})$ with respect to $p_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}(\mathbf{z})$ can be computed similar to (4), which we refer to as $\mathbb{P}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}[e_{\boldsymbol{\phi}}]$.

Our main goal is to learn a map $M_{\boldsymbol{\psi}}(\cdot)$ and a classifier $h_{\boldsymbol{\phi}_1}(\mathbf{z})$ in a way that the probability of error of $h_{\boldsymbol{\phi}_1}(\mathbf{z})$ with respect to the mapping of the true distribution to $\mathcal{Z}$, i.e., $\mathbb{P}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}[e_{\boldsymbol{\phi}_1}]$, is small. To this end, we repurpose theories from the domain-adaptation literature in the following to obtain an upper bound on $\mathbb{P}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}[e_{\boldsymbol{\phi}_1}]$, which leads to explicit loss functions for the joint learning of $M_{\boldsymbol{\psi}}$ and $h_{\boldsymbol{\phi}_1}(\mathbf{z})$ using both the training and synthetic datasets. Specifically, it is desired for the mapping $M_{\boldsymbol{\psi}}(\cdot)$ from $\mathcal{X}$ to $\mathcal{Z}$ to transform the true and estimated distributions in a way that $p_{\boldsymbol{\psi},\boldsymbol{\theta}^*}(\mathbf{z})$ and $p_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}(\mathbf{z})$, which are defined in the feature space $\mathcal{Z}$, are similar. Mathematically, this similarity should be measured in terms of a distance metric. However, as there are only a limited number of samples available from $p_{\boldsymbol{\psi},\boldsymbol{\theta}^*}(\mathbf{z})$, we need to be able to approximate this distance from a finite number of samples. We expand further on this idea by primarily focusing on binary classification in this section, although the results are extendable to the classification task in general. We begin with the following distance definitions.

Definition 1. *For a family of binary-valued functions $\mathcal{H}_{\boldsymbol{\Phi}} = \{h_{\boldsymbol{\phi}} : \mathcal{Z} \to \{0,1\}\}$, in which every member $h_{\boldsymbol{\phi}} \in \mathcal{H}_{\boldsymbol{\Phi}}$ is parameterized by a real vector $\boldsymbol{\phi} \in \boldsymbol{\Phi}$, and the set $A_{\boldsymbol{\phi}} = \{\mathbf{z} | h_{\boldsymbol{\phi}}(\mathbf{z}) = 1, \mathbf{z} \in \mathcal{Z}\}$, the $\mathcal{A}_\Phi$-distance between $p_{\boldsymbol{\psi},\boldsymbol{\theta}^*}(\mathbf{z})$ and $p_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}(\mathbf{z})$ is defined as*

$$d_{\mathcal{A}_\Phi}\big(p_{\boldsymbol{\psi},\boldsymbol{\theta}^*}(\mathbf{z}), p_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}(\mathbf{z})\big) \triangleq 2 \sup_{h_{\boldsymbol{\phi}} \in \mathcal{H}_\Phi} \left| \int_{A_{\boldsymbol{\phi}}} (p_{\boldsymbol{\psi},\boldsymbol{\theta}^*}(\mathbf{z}) - p_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}(\mathbf{z})) d\mathbf{z} \right|. \quad (5)$$

Similarly, for $B_{\phi_1,\phi_2} = \{\mathbf{z} | h_{\phi_1}(\mathbf{z}) \neq h_{\phi_2}(\mathbf{z}), \mathbf{z} \in \mathcal{Z}\}$, the $\mathcal{B}_\Phi$-distance refers to [3]

$$d_{\mathcal{B}_\Phi}\big(p_{\boldsymbol{\psi},\boldsymbol{\theta}^*}(\mathbf{z}), p_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}(\mathbf{z})\big) \triangleq 2 \sup_{h_{\phi_1}, h_{\phi_2} \in \mathcal{H}_\Phi} \left| \int_{B_{\phi_1,\phi_2}} (p_{\boldsymbol{\psi},\boldsymbol{\theta}^*}(\mathbf{z}) - p_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}(\mathbf{z})) d\mathbf{z} \right|. \quad (6)$$

The $\mathcal{A}_\Phi$-distance is also referred to via other names like A-distance and $\mathcal{H}$-distance in [25], [40]. By looking at the following extreme choices of $\mathcal{H}_{\boldsymbol{\Phi}}$, these distances are clearly a function of richness of the class $\mathcal{H}_\Phi$. For a very restrictive choice of only constant functions, i.e., $\mathcal{H}_\Phi = \{h_{\boldsymbol{\phi}} | h_{\boldsymbol{\phi}}(\mathbf{z}) = 0, \forall \mathbf{z}\} \bigcup \{h_{\boldsymbol{\phi}} | h_{\boldsymbol{\phi}}(\mathbf{z}) = 1, \forall \mathbf{z}\}$, $d_{\mathcal{A}_\Phi}$ is always zero as the only possible choice for $A_{\boldsymbol{\phi}}$ is either the empty set or $\mathcal{Z}$. On the other hand, for $\mathcal{H}_\Phi = \{h_{\boldsymbol{\phi}} | h_{\boldsymbol{\phi}}(\mathbf{z}) = 0 \text{ or } h_{\boldsymbol{\phi}}(\mathbf{z}) = 1, \forall \mathbf{z}\}$, which represents all the binary functions, $d_{\mathcal{A}_\Phi}$ is identical to definition of the total variation distance [41] as the sup in (5) will effectively be over the σ-algebra of subsets of the $\mathcal{Z}$ space. This dependence of $d_{\mathcal{A}_\Phi}$ on the underlying family of functions makes it possible to obtain an expression for the $\mathcal{A}_\Phi$-distance based on the finite set of samples from each distribution. Specifically, consider two sets $\mathcal{Z}_{\boldsymbol{\psi},\boldsymbol{\theta}^*} = \{\mathbf{z}_{r,i}\}_{i=1}^{N_r}$ and $\mathcal{Z}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}} = \{\mathbf{z}_{s,i}\}_{i=1}^{N_s}$ sampled from the distributions $p_{\boldsymbol{\psi},\boldsymbol{\theta}^*}(\mathbf{z})$ and $p_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}(\mathbf{z})$ in an i.i.d. fashion, respectively. In this case, for a family $\mathcal{H}_{\boldsymbol{\Phi}}$ that satisfies the condition that if $h_\phi \in \mathcal{H}_\Phi$ then $1 - h_\phi \in \mathcal{H}_\Phi$, the $\mathcal{A}_\Phi$-distance can be approximated from $\mathcal{Z}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}$ and $\mathcal{Z}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}$ using [40]

$$\widehat{d}_{\mathcal{A}_\Phi}(\mathcal{Z}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}, \mathcal{Z}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}) = 2\bigg(1 - \inf_{h_\phi \in \mathcal{H}_\Phi} \Big(\frac{1}{N_r} \sum_{i=1}^{N_r} \mathbb{1}_{\{h_\phi(\mathbf{z}_{r,i})=0\}} + \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbb{1}_{\{h_\phi(\mathbf{z}_{s,i})=1\}}\Big)\bigg). \quad (7)$$

As the bound on $\mathbb{P}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}[e_{\boldsymbol{\phi}_1}]$ should be obtained based on a finite number of training and synthetic samples, it is then of interest to see how far $\widehat{d}_{\mathcal{A}_\Phi}$ is from $d_{\mathcal{A}_\Phi}$. To answer this question, one needs to rely on a measure of complexity for a given class of functions such as the VC dimension [38] and Rademacher complexity [42]. As we have chosen the mapping function $M_{\boldsymbol{\psi}}(\mathbf{x})$ and the classifier $h_{\boldsymbol{\phi}_1}(\mathbf{z})$ to be NNs, we present the results based on the Rademacher complexity defined as follows, which can be computed for certain classes of neural networks in a closed-form fashion [42].

Definition 2. *Let $\mathcal{Z}_1 = \{\mathbf{z}_i\}_{i=1}^N$ be a set of i.i.d. samples drawn from a distribution $p(\mathbf{z})$ that is supported on $\mathcal{Z}$. For $\mathcal{H}_\Phi$, a family of real-valued functions over $\mathcal{Z}$, the empirical Rademacher complexity of $\mathcal{H}_\Phi$, given a dataset $\mathcal{Z}_1$, is defined as*

$$R_{\mathcal{Z}_1}(\mathcal{H}_\Phi) \triangleq \mathop{\mathbb{E}}_{\substack{\sigma_i \sim \{-1,+1\} \\ i=1,\ldots,N}} \left[\sup_{h_\phi \in \mathcal{H}_\Phi} \left(\frac{1}{N} \sum_{i=1}^N \sigma_i h_\phi(\mathbf{z}_i) \right) \right], \quad (8)$$

where the expectation is over all the σ_i's, each taking a binary value with equal probability.

Lemma 2 ([42])**.** *Consider a family of functions $\mathcal{H}_{\boldsymbol{\Phi}} = \{h_{\boldsymbol{\phi}} : \mathcal{Z} \to \{0,1\}\}$ and a distribution $p(\mathbf{z})$ over $\mathcal{Z}$. For a set $\mathcal{Z}_1 = \{\mathbf{z}_i\}_{i=1}^N$ of N i.i.d. samples from $p(\mathbf{z})$ and any $0 < \delta < 1$, the following holds $\forall h_{\boldsymbol{\phi}} \in \mathcal{H}_\Phi$ with probability at least $1 - \delta$:*

$$\mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}[h_{\boldsymbol{\phi}}(\mathbf{z})] \leq \frac{1}{N} \sum_{i=1}^N h_{\boldsymbol{\phi}}(\mathbf{z}_i) + 2R_{\mathcal{Z}_1}(\mathcal{H}_\Phi) + 3\sqrt{\frac{\log(2/\delta)}{2N}}. \quad (9)$$

Now, the difference between $d_{\mathcal{A}_\Phi}$ and $\widehat{d}_{\mathcal{A}_\Phi}$ can be bounded in terms of the complexity of the underlying family of functions and the number of available samples as stated in the following lemma.

Lemma 3. *Let $\mathcal{Z}_{\boldsymbol{\psi},\boldsymbol{\theta}^*} = \{\mathbf{z}_{r,i}\}_{i=1}^{N_r}$ and $\mathcal{Z}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}} = \{\mathbf{z}_{s,i}\}_{i=1}^{N_s}$ be sets of i.i.d. samples corresponding to the distributions $p_{\boldsymbol{\psi},\boldsymbol{\theta}^*}(\mathbf{z})$*

[3] Similar to the total variation distance, it can be readily verified that $d_{\mathcal{A}_\Phi}$ and $d_{\mathcal{B}_\Phi}$ are also distance metrics.

and $p_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}(\mathbf{z})$ on the space $\mathcal{Z}$, respectively. Then, for any $0<\delta<1$ and a family of functions $\mathcal{H}_{\boldsymbol{\Phi}}=\{h_{\boldsymbol{\phi}}:\mathcal{Z}\to\{0,1\}\}$, we have

$$d_{\mathcal{A}_{\Phi}}\big(p_{\boldsymbol{\psi},\boldsymbol{\theta}^*}(\mathbf{z}),p_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}(\mathbf{z})\big)\leq \widehat{d}_{\mathcal{A}_{\Phi}}(\mathcal{Z}_{\boldsymbol{\psi},\boldsymbol{\theta}^*},\mathcal{Z}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}})+2R_{\mathcal{Z}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}}(\mathcal{H}_{\Phi}) +2R_{\mathcal{Z}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}}(\mathcal{H}_{\Phi})+3\sqrt{(\log 2/\delta)/2N_r}+3\sqrt{(\log 2/\delta)/2N_s} \tag{10}$$

with probability at least $1-\delta$.

Proof. See Appendix A. □

The above lemma enables us to bound the $\mathcal{A}_{\boldsymbol{\Phi}}$ distance between two distributions in terms of the collected samples from each. Equipped with this result, we are able to bound the probability of error $\mathbb{P}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}[e_{\boldsymbol{\phi}_1}]$ via the following theorem.

Theorem 1. *Assume that the training and synthetic datasets are mapped into the feature space $\mathcal{Z}$ through the mapping function $M_{\boldsymbol{\psi}}(\mathbf{x})$, with the resulting samples denoted by $\mathcal{Z}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}=\{\mathbf{z}_{r,i}\}_{i=1}^{N_r}$ and $\mathcal{Z}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}=\{\mathbf{z}_{s,i}\}_{i=1}^{N_s}$, respectively. Then, for any $0<\delta<1$ and a family of functions $\mathcal{H}_{\boldsymbol{\Phi}}:\mathcal{Z}\to\{0,1\}$, $\mathbb{P}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}[e_{\boldsymbol{\phi}_1}],\forall h_{\boldsymbol{\phi}_1}\in\mathcal{H}_{\boldsymbol{\Phi}}$ is bounded by*

$$\mathbb{P}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}[e_{\boldsymbol{\phi}_1}]\leq \mathbb{P}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}[e_{\boldsymbol{\phi}_1}]+\frac{1}{2}\widehat{d}_{\mathcal{A}_{\Phi}}(\mathcal{Z}_{\boldsymbol{\psi},\boldsymbol{\theta}^*},\mathcal{Z}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}})+R_{\mathcal{Z}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}}(\mathcal{H}_{\boldsymbol{\Phi}})+ R_{\mathcal{Z}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}}(\mathcal{H}_{\boldsymbol{\Phi}})+\frac{3}{2}\sqrt{(\log 2/\delta)/2N_r}+\frac{3}{2}\sqrt{(\log 2/\delta)/2N_s}). \tag{11}$$

Proof. See Appendix B. □

The above theorem bounds the probability of error with respect to $p_{\boldsymbol{\psi},\boldsymbol{\theta}^*}(\mathbf{z})$ associated with a classifier $h_{\boldsymbol{\phi}_1}(\mathbf{z})$ in terms of the quantities that do not depend on the the unknown true parameters $\boldsymbol{\theta}^*$. As our primary goal is to make $\mathbb{P}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}[e_{\boldsymbol{\phi}_1}]$ as small as possible, the mapping function $M_{\boldsymbol{\psi}}(\mathbf{x})$ and the classifier $h_{\boldsymbol{\phi}_1}(\mathbf{z})$ should be chosen in a way to minimize the above upper bound. We note that the complexity related terms in the above bound are fixed for a chosen family of the functions and the bound is primarily controlled by the first two terms. In other words, $M_{\boldsymbol{\psi}}(\mathbf{x})$ and $h_{\boldsymbol{\phi}_1}(\mathbf{z})$ should be chosen such that the probability of classification error with respect to the mapping of the estimated distribution in the $\mathcal{Z}$ space, i.e., $\mathbb{P}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}[e_{\boldsymbol{\phi}_1}]$, and the approximated $\mathcal{A}_{\boldsymbol{\Phi}}$-distance between the synthetic and training datasets are minimized simultaneously. To achieve this goal, we restrict ourselves to $M_{\boldsymbol{\psi}}(\mathbf{x})$ and $h_{\boldsymbol{\phi}_1}(\mathbf{z})$ that correspond to NNs that are trained to minimize a loss function in accordance with the first two terms of the above bound. One can efficiently solve the resulting optimization problem via the stochastic gradient descent method as described in the following.

Joint learning of the feature map and the classifier: In terms of specifics, we assume $M_{\boldsymbol{\psi}}(\mathbf{x})$ and $h_{\boldsymbol{\phi}_1}(\mathbf{z})$ belong to the class of feed-forward (deep) NNs whose parameters, i.e., $\boldsymbol{\psi}$ and $\boldsymbol{\phi}_1$, correspond to the weights and biases of each network. The input and output layers of the NNs corresponding to $M_{\boldsymbol{\psi}}$ have n_x and n_z number of neurons, respectively, which denote the dimensions of the spaces $\mathcal{X}$ and $\mathcal{Z}$, respectively. We note that n_x is chosen according to the length of the observation vector as part of the problem formulation, while n_z can be picked as a hyper-parameter to facilitate the training process. Subsequently, the input layer of $h_{\boldsymbol{\phi}_1}$ has n_z neurons while its output layer contains C neurons whose activation function is chosen to be the softmax function $\boldsymbol{\sigma}(\mathbf{z})$ for which the ith element is given by $\frac{e^{\mathbf{z}[i]}}{\sum_{i=1}^{n_z}e^{\mathbf{z}[i]}}$. In this way, the ith component of the vector $\mathbf{y}_{\boldsymbol{\psi},\boldsymbol{\phi}_1,\mathbf{x}}\triangleq h_{\boldsymbol{\phi}_1}\big(M_{\boldsymbol{\psi}}(\mathbf{x})\big)$ denotes the probability that the classifier assigns to the input $\mathbf{x}$ that it belongs to the ith class for $i=0,\ldots,C-1$. Consequently, the averaged cross-entropy loss, minimizing of which leads to minimizing the classification error associated with $h_{\boldsymbol{\phi}_1}$, over the synthetic dataset $\mathcal{D}_s$ equals

$$\mathcal{L}_s(\boldsymbol{\psi},\boldsymbol{\phi}_1|\mathcal{D}_s)=\frac{1}{n_s}\sum_{n=1}^{n_s}\sum_{i=1}^{C}\mathbf{l}_{s,n}[n]\log\mathbf{y}_{\boldsymbol{\psi},\boldsymbol{\phi}_1,\mathbf{x}_{s,n}}[n], \tag{12}$$

where $\mathbf{l}_{s,n}=\mathbf{e}_C(y_{s,n})$ denotes the one-hot encoded version of the label $y_{s,n}$ corresponding to the nth sample. Regrading the computation of $\widehat{d}_{\mathcal{A}_{\Phi}}$ between the two sets $\mathcal{Z}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}$ and $\mathcal{Z}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}$, it is suggested by the authors in [25], [40] that the classification accuracy corresponding to a classifier trained to distinguish between the samples from the two sets can be used as a surrogate for the inf part in (7) that can be readily computed during the learning process. To train such classifier, we consider a NN $d_{\boldsymbol{\zeta}}$ with n_z input neurons and 2 output neurons with softmax activation function, which is trained to distinguish between $\mathcal{Z}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}$ and $\mathcal{Z}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}$ labeled as 0 and 1, respectively. Consequently, by defining a two-dimensional vector $d_{\boldsymbol{\psi},\boldsymbol{\zeta},\mathbf{x}}\triangleq d_{\boldsymbol{\zeta}}\big(M_{\boldsymbol{\psi}}(\mathbf{x})\big)$, the $\widehat{d}_{\mathcal{A}_{\Phi}}$ term can be approximated by the cross-entropy loss associated with $d_{\boldsymbol{\zeta}}$ as follows:

$$\mathcal{L}_d(\boldsymbol{\psi},\boldsymbol{\zeta}|\mathcal{D}_s,\mathcal{D}_r)=2\big(1-2\mathcal{L}_c(\boldsymbol{\psi},\boldsymbol{\zeta}|\mathcal{D}_s,\mathcal{D}_r)\big), \tag{13}$$

$$\mathcal{L}_c(\boldsymbol{\psi},\boldsymbol{\zeta}|\mathcal{D}_s,\mathcal{D}_r)=\frac{1}{n_r}\sum_{i=1}^{n_r}\log d_{\boldsymbol{\psi},\boldsymbol{\zeta},\mathbf{x}_{r,n}}[1]+ \frac{1}{n_s}\sum_{n=1}^{n_s}\log d_{\boldsymbol{\psi},\boldsymbol{\zeta},\mathbf{x}_{s,n}}[2]. \tag{14}$$

Now, using Theorem 1 the training goal for the constituent NNs is set to simultaneously minimize the classification error corresponding to the synthetic data and the distance between the real and synthetic data, both measured in the mapped space $\mathcal{Z}$. Specifically, the NNs $M_{\boldsymbol{\psi}}$ and $h_{\boldsymbol{\phi}_1}$ should be trained to minimize the sum of the losses in (12) and (13), while the classifier $d_{\boldsymbol{\zeta}}$ is trained to minimize (14). As $M_{\boldsymbol{\psi}}$ is trained to maximize $\mathcal{L}_c(\boldsymbol{\psi},\boldsymbol{\zeta})$ despite $d_{\boldsymbol{\zeta}}$'s goal to minimize $\mathcal{L}_c(\boldsymbol{\psi},\boldsymbol{\zeta})$, the learning process involves adversarial training between these two NNs. Based on the approach taken in [25] for adversarial training in the context of domain adaptation, we train the above three NNs for finding the saddle points $\widehat{\boldsymbol{\psi}}$, $\widehat{\boldsymbol{\phi}}_1$ and $\widehat{\boldsymbol{\zeta}}$, such that

$$\widehat{\boldsymbol{\psi}},\widehat{\boldsymbol{\phi}}_1=\underset{\boldsymbol{\psi},\boldsymbol{\phi}_1}{\operatorname{argmin}}\,\mathcal{L}_t(\boldsymbol{\psi},\boldsymbol{\phi}_1,\widehat{\boldsymbol{\zeta}}|\mathcal{D}_s,\mathcal{D}_r), \tag{15}$$

$$\widehat{\boldsymbol{\zeta}}=\underset{\boldsymbol{\zeta}}{\operatorname{argmin}}\,-\mathcal{L}_t(\widehat{\boldsymbol{\psi}},\widehat{\boldsymbol{\phi}}_1,\boldsymbol{\zeta}|\mathcal{D}_s,\mathcal{D}_r), \tag{16}$$

$$\mathcal{L}_t(\boldsymbol{\psi},\boldsymbol{\phi}_1,\boldsymbol{\zeta}|\mathcal{D}_s,\mathcal{D}_r)=\mathcal{L}_s(\boldsymbol{\psi},\boldsymbol{\phi}_1|\mathcal{D}_s)+\mathcal{L}_d(\boldsymbol{\psi},\boldsymbol{\zeta}|\mathcal{D}_s,\mathcal{D}_r), \tag{17}$$

which can be achieved by utilizing the stochastic gradient descent algorithm for each minimization task. To this end, the minimization is performed over the NN's parameters, $\boldsymbol{\psi}$, $\boldsymbol{\phi}_1$ and $\boldsymbol{\zeta}$, that are real vectors whose dimensions are determined by the architecture of each network.

B. An illustrative example: The case of two-dimensional Gaussian data

Next, we show how the learning-based classifier in Section III-A performs on simple training and synthetic datasets in an illustrative manner. To this end, we consider a toy example where the true and estimated distributions are a mixture of two bivariate Gaussian distributions with full-rank covariance matrix each. In particular, we focus on the problem of binary classification where the distribution for the ith class is denoted by $p_i(\mathbf{x};\boldsymbol{\theta}_i^*)=\mathcal{N}(\boldsymbol{\mu}_i,\boldsymbol{\Sigma})$ for $i=0,1$, $\boldsymbol{\mu}_i\in\mathbb{R}^{2\times 1}$, $\boldsymbol{\Sigma}\in\mathbb{R}^{2\times 2}$, and equal priors. In order to investigate the effect of mismatch between only mean parameters, the corresponding estimated distributions are assumed to have the same covariance but different means, i.e., $p_i(\mathbf{x};\widehat{\boldsymbol{\theta}}_i)=\mathcal{N}(\widehat{\boldsymbol{\mu}}_i,\boldsymbol{\Sigma})$ for $i=0,1$ and equal priors. For two multivariate Gaussian distributions, the authors in [41] have proposed a bound for the corresponding total variation as part of the following theorem.

Theorem 2 ([41]). *Consider two d-dimensional Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1)$ and $\mathcal{N}(\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_2)$ where $\boldsymbol{\mu}_1\neq\boldsymbol{\mu}_2$ and $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are positive definite. Let $\mathbf{v}=\boldsymbol{\mu}_1-\boldsymbol{\mu}_2$ and $\boldsymbol{\Pi}$ be a $d\times(d-1)$ matrix whose columns form a basis for the subspace orthogonal to $\mathbf{v}$. Denote the eigenvalues of $(\boldsymbol{\Pi}^T\boldsymbol{\Sigma}_1\boldsymbol{\Pi})^{-1}\boldsymbol{\Pi}^T\boldsymbol{\Sigma}_2\boldsymbol{\Pi}-\mathbf{I}_{d-1}$ by $\rho_1,\ldots,\rho_{d-1}$. Then, the total variation between the two distribution can be bounded as*

$$\frac{1}{200}\leq\frac{TV(\mathcal{N}(\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1),\mathcal{N}(\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_2))}{\min(1,V)}\leq\frac{9}{2}\qquad(18)$$

where $V\overset{def}{=}\max\left\{\frac{|\mathbf{v}^T(\boldsymbol{\Sigma}_1-\boldsymbol{\Sigma}_2)\mathbf{v}|}{\mathbf{v}^T\boldsymbol{\Sigma}_1\mathbf{v}},\frac{\mathbf{v}^T\mathbf{v}}{\sqrt{\mathbf{v}^T\boldsymbol{\Sigma}_1\mathbf{v}}},\sqrt{\sum_{i=1}^{d-1}\rho_i^2}\right\}$.

We note that a bound on total variation would also bound the $\mathcal{A}_\Phi$ distance following the discussion after Definition 1. Using the above result, we can bound the total variation distance between $p_i(\mathbf{x};\boldsymbol{\theta}_i^*)$ and $p_i(\mathbf{x};\widehat{\boldsymbol{\theta}}_i)$ as follows, which will provide useful insights in the remainder of this section about the learning process described in Section III-A.

Corollary 1. *For two Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_0,\boldsymbol{\Sigma})$ and $\mathcal{N}(\widehat{\boldsymbol{\mu}}_0,\boldsymbol{\Sigma})$ with the same positive definite covariance matrix $\boldsymbol{\Sigma}$, the corresponding total variation is bounded from the above by* $\frac{9}{2}\min\left(1,\frac{(\boldsymbol{\mu}_0-\widehat{\boldsymbol{\mu}}_0)^T(\boldsymbol{\mu}_0-\widehat{\boldsymbol{\mu}}_0)}{\sqrt{(\boldsymbol{\mu}_i-\widehat{\boldsymbol{\mu}}_0)^T\boldsymbol{\Sigma}(\boldsymbol{\mu}_0-\widehat{\boldsymbol{\mu}}_0)}}\right)$.

Regarding the specific architecture for the NNs utilized in Section III-A, let us now choose the mapping function $M_{\boldsymbol{\psi}}$ to be $M_{\boldsymbol{\psi}}(\mathbf{x})=\mathbf{W}_{\boldsymbol{\psi},2}\mathbf{W}_{\boldsymbol{\psi},1}\mathbf{x}$ parameterized by $\boldsymbol{\psi}=\{\mathbf{W}_{\boldsymbol{\psi},1}\in\mathbb{R}^{20\times 2},\mathbf{W}_{\boldsymbol{\psi},2}\in\mathbb{R}^{2\times 20}\}$. In particular, we have set the dimension of the space $\mathcal{Z}$ to $n_z=2$ in order to be able to readily visualize it within the 2D coordinate system. For each $h_{\boldsymbol{\phi}_1}$ and $d_{\boldsymbol{\zeta}}$, we choose a two-layer NN with softmax activation function. Specifically, for $h_{\boldsymbol{\phi}_1}$ we have $h_{\boldsymbol{\phi}_1}\big(M_{\boldsymbol{\psi}}(\mathbf{x})\big)=\text{softmax}\big(\mathbf{V}_{\boldsymbol{\phi}_1,2}(\mathbf{V}_{\boldsymbol{\phi}_1,1}M_{\boldsymbol{\psi}}(\mathbf{x})+\mathbf{b}_{\boldsymbol{\phi}_1,1})+\mathbf{b}_{\boldsymbol{\phi}_1,2}\big)$ where $\boldsymbol{\phi}_1=\{\mathbf{V}_{\boldsymbol{\phi}_1,1}\in\mathbb{R}^{20\times 2},\mathbf{b}_{\boldsymbol{\phi}_1,1}\in\mathbb{R}^{20},\mathbf{V}_{\boldsymbol{\phi}_1,2}\in\mathbb{R}^{2\times 20},\mathbf{b}_{\boldsymbol{\phi}_1,2}\in\mathbb{R}^2\}$. Similarly, $d_{\boldsymbol{\zeta}}$ is chosen to be $d_{\boldsymbol{\zeta}}\big(M_{\boldsymbol{\psi}}(\mathbf{x})\big)=\text{softmax}\big(\mathbf{U}_{\boldsymbol{\phi}_1,2}(\mathbf{U}_{\boldsymbol{\phi}_1,1}M_{\boldsymbol{\psi}}(\mathbf{x})+\mathbf{b}_{\boldsymbol{\phi}_1,1})+\mathbf{b}_{\boldsymbol{\phi}_1,2}\big)$ for $\boldsymbol{\zeta}=\{\mathbf{U}_{\boldsymbol{\phi}_1,1}\in\mathbb{R}^{20\times 2},\mathbf{b}_{\boldsymbol{\phi}_1,1}\in\mathbb{R}^{20},\mathbf{U}_{\boldsymbol{\phi}_1,2}\in\mathbb{R}^{2\times 20},\mathbf{b}_{\boldsymbol{\phi}_1,2}\in\mathbb{R}^2\}$. Training of these NNs involves finding the saddle points of (15) based on the available training and synthetic datasets which would lead to the learning-based classifier $h_{\boldsymbol{\phi}_1}$. We note that the above simple choice of the mapping function maps $p_i(\mathbf{x};\theta_i^*),\ i=0,\ 1$ to Gaussian distributions in the $\mathcal{Z}$ space which allows us to utilize Corollary 1 for analyzing the total variation distance between these mappings in the following.

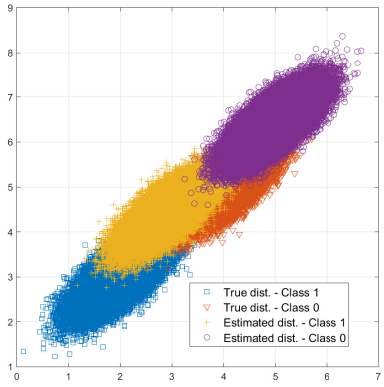
We now resort to numerical results for further illustration of this example. To this end, we set $\boldsymbol{\mu}_0=[2.9,4.4]$, $\boldsymbol{\mu}_1=[5,6.4]$, $\widehat{\boldsymbol{\mu}}_0=[2,3]$, $\widehat{\boldsymbol{\mu}}_1=[4,5]$ and $\boldsymbol{\Sigma}=\begin{bmatrix}0.15 & 0.11\\ 0.11 & 0.15\end{bmatrix}$. Also, we generate $n_r=40$ samples from the true distribution, while $n_s=2000$ samples are generated from the estimated distribution. The Figs. 2a and 2b depict the samples from the true and estimated distributions and their mapping through the function $M_{\boldsymbol{\psi}}$ into the $\mathcal{Z}$ space, respectively. Furthermore, the positions of the means corresponding to the samples from the real and estimated distributions in both space $\mathcal{X}$ and $\mathcal{Z}$ are illustrated in Fig. 2c. An important observation in relation to the Corollary 1 can be made by noting that the total variation between $\mathcal{N}(\boldsymbol{\mu}_i,\boldsymbol{\Sigma})$ and $\mathcal{N}(\widehat{\boldsymbol{\mu}}_i,\boldsymbol{\Sigma})$ is bounded by the term $||\mathbf{v}||\frac{\mathbf{e}_v^T\mathbf{e}_v}{\sqrt{\mathbf{e}_v^T\boldsymbol{\Sigma}\mathbf{e}_v}}$ where $\mathbf{v}=\boldsymbol{\mu}_i-\widehat{\boldsymbol{\mu}}_i$ and $\mathbf{e}_v=\mathbf{v}/||\mathbf{v}||$. Assuming λ_1 and λ_2 are eigenvalues of $\boldsymbol{\Sigma}$ with corresponding eigenvectors $\mathbf{u}_1$ and $\mathbf{u}_2$ such that $\lambda_1>\lambda_2$, it is straightforward to show that the maximal value of $\mathbf{e}_v^T\boldsymbol{\Sigma}\mathbf{e}_v=\lambda_1(\mathbf{u}_1^T\mathbf{e}_v)+\lambda_2(\mathbf{u}_2^T\mathbf{e}_v)$ is achieved when $\mathbf{e}_v\perp\mathbf{u}_2$. Therefore, for $||\mathbf{v}||\frac{\mathbf{e}_v^T\mathbf{e}_v}{\sqrt{\mathbf{e}_v^T\boldsymbol{\Sigma}\mathbf{e}_v}}$ to be minimized $\mathbf{e}_v$ ought to be in the same direction of $\mathbf{u}_1$ while $||\mathbf{v}||$ become minimum. Notably, Figs. 2b and 2c highlight the fact that finding the saddle points in (15) in part corresponds to mapping the datasets to a feature space $\mathcal{Z}$ that satisfy both these two criteria.

IV. Case Study I: Detection of Channel-based Spoofing for Physical Layer Security

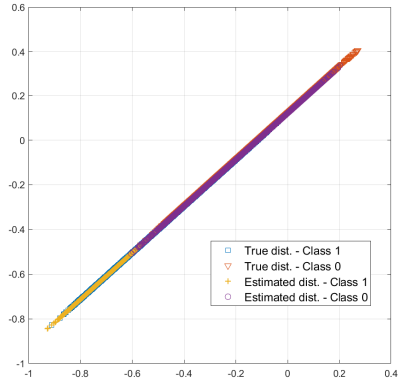
We now present the first case study concerning channel spoofing detection, which arises in a wireless communication environment where a legitimate transmitter (Alice) is transmitting signals to a legitimate receiver (Bob) in the presence of an adversary (Eve). Eve aims at spoofing the Alice–Bob's channel by using Alice's MAC address [7], [9]. Bob's goal, in this setting, is to distinguish between the signals coming from Alice and Eve based on the corresponding channel frequency responses (CFRs)

A. System model

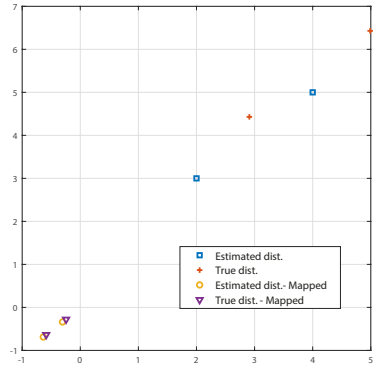
We envision the communication parties in a 5G propagation setting relying on MIMO-OFDM wideband communications, where the number of antennas are set to N_{Tx} and N_{Rx} at the transmitter (Tx) and the receiver (Rx), respectively. We assume Bob measures and stores CFR samples corresponding to a transmitting terminal (either Alice or Eve) at M tones, across an overall system bandwidth of W. We consider a generalized time-varying channel model for a transmitting terminal, where each measured CFR sample is made up of three components: 1) specular paths ($\overline{\mathbf{h}}$), 2) time-varying part $\mathbf{d}_u$, and 3) noise $\mathbf{n}$, all of which are complex vectors of size $M\times 1$. The specular paths model the dominant portion of the channel, which remains unchanged within a coherence time. The time-varying part



(a) Samples corresponding to the true and estimated distributions in space $\mathcal{X}$.



(b) Mapping of the samples via the function $M_{\boldsymbol{\psi}}$ to the space $\mathcal{Z}$.



(c) Position of the means in the original space $\mathcal{X}$ and the space $\mathcal{Z}$.

Fig. 2: Visualization of the true and estimated distributions and their mappings to the space $\mathcal{Z}$ for the case of 2D Gaussian datasets.

models the dense multipath components, which accounts for the diffuse scattering between two transceivers. Finally, the noise part models the measurement noise. The measured CFR at Bob at time $t = uT$ for a sampling interval T and $u \in \mathbb{N}$ is denoted by $\mathbf{h}_u$, which is a $M \times 1$ vector such that

$$\mathbf{h}_u = \bar{\mathbf{h}} + \mathbf{d}_u + \mathbf{n}. \tag{19}$$

We first introduce the dominant paths model suitable for MIMO-OFDM communications under a frequency-dependent array response [43]. For this scenario, the $N_{Rx} \times N_{Tx}$ channel matrix associated with the nth subcarrier ($n = 1, \ldots, N_f$) is expressed as

$$\mathbf{H}[n] = \mathbf{A}_R[n]\mathbf{\Gamma}[n]\mathbf{A}_T^H[n], \tag{20}$$

where N_f denotes the total number of subcarriers. In this way, the size of the vector $\mathbf{h}_u$ equals $M = N_f \times N_{Rx} \times N_{Tx}$. We further denote the subcarrier width and carrier frequency with Δf and f_0, respectively. Here, the antenna steering and response vectors are, respectively, defined as

$$\mathbf{A}_T[n] = [\mathbf{a}_{T,n}(\psi_{T,0}), \ldots, \mathbf{a}_{T,n}(\psi_{T,K-1})] \quad \text{and} \tag{21}$$
$$\mathbf{A}_R[n] = [\mathbf{a}_{Rx,n}(\psi_{R,0}), \ldots, \mathbf{a}_{R,n}(\psi_{Rx,K-1})], \tag{22}$$

where K is the total number of dominant paths. Also, $\psi_{T,k}$ and $\psi_{R,k}$ denote the azimuth angles corresponding to the transmit and receive sides for the kth path. The structure of the frequency-dependent antenna steering and response vectors $\mathbf{a}_{T,n}(\psi_{T,K-1})$ and $\mathbf{a}_{R,n}(\psi_{R,K-1})$ depends on the specific array structure. For the case of a uniform linear array (ULA), which we consider in this work, we have

$$\mathbf{a}_{T,n}(\psi_{T,k}) = \frac{1}{N_{Tx}}[e^{-j\frac{N_{Tx}-1}{2}\psi_{T,k}}, \ldots, e^{j\frac{N_{Tx}-1}{2}\psi_{T,k}}], \tag{23}$$

where $\psi_{T,k} = \frac{2\pi}{\lambda_n} d \sin(\theta_{Tx,k})$, $\lambda_n = c(NT + f_c)/n$ denotes the signal bandwidth at the nth subcarrier, c is the speed of light, and d refers to the distance between two antenna elements. Similarly, $\mathbf{a}_{Rx,n}(\psi_{Rx,k})$ can be defined for the receiver's antennas. The path gain matrix is obtained by

$$\mathbf{\Gamma}[n] = \sqrt{N_{Rx}N_{Tx}} \operatorname{diag}\left\{\rho_0 e^{-j2\pi n\tau_0/(NT_s)}, \ldots, \rho_{K-1}e^{-j2\pi n\tau_{K-1}/(NT_s)}\right\}, \tag{24}$$

where ρ_k and τ_k denote the complex channel gain and delay associated with the kth path, while T_s is the sampling interval. Then, $\bar{\mathbf{h}}$ is defined as concatenation of the vectorized version of $\mathbf{H}[n]$ for all the subcarriers $n = 1, \ldots, N_f$, i.e.,

$$\bar{\mathbf{h}} = \left[\text{vec}\{\mathbf{H}[1]\}^T, \ldots, \text{vec}\{\mathbf{H}[N]\}^T\right]^T, \tag{25}$$

where vec$\{\cdot\}$ denotes the column-wise vectorization operator. We denote the parameters associated with the specular paths contribution, $\bar{\mathbf{h}}$, which remain constant during a coherence time T_c corresponding to the coherence bandwidth B_c, via a $4K \times 1$ vector $\boldsymbol{\theta}_{sp}$ defined as

$$\boldsymbol{\theta}_{sp} = [\boldsymbol{\psi}_T, \boldsymbol{\psi}_R, \boldsymbol{\tau}, \boldsymbol{\rho}]^T, \tag{26}$$

where $\boldsymbol{\psi}_T = [\psi_{T,0}, \ldots, \psi_{T,K-1}]$, $\boldsymbol{\psi}_R = [\psi_{R,0}, \ldots, \psi_{R,K-1}]$, $\boldsymbol{\tau} = [\tau_0, \ldots, \tau_{K-1}]$ and $\boldsymbol{\rho} = [\rho_0, \ldots, \rho_{K-1}]$.

For modeling the variable part of the channel we first assume that the wide-sense stationary uncorrelated scattering (WSSUS) assumption holds, and then use a multipath tapped delay line, $h(t, \tau) = \sum_{l=0}^{L-1} A_l(t)\delta(\tau - l\Delta\tau)$, to model the impulse response at time t between any pair of transmit and receive antennas. Here, $A_l(t)$ and $\Delta\tau = 1/W$ denote the (complex) amplitude of the lth *virtual* path[4] and the delay between two consecutive paths, respectively. Sampling the impulse response at time $t = uT$, followed by taking the Fourier transform with

[4]We note that the diffuse spectrum contribution arises from superposition of infinite number of diffuse paths. We use the term virtual path to account for superposition of large number of diffuse paths with similar physical layer characteristics.

respect to τ would result in a vector $\mathbf{q}_u$ whose nth element is denoted by

$$\mathbf{q}_u[n] = \mathcal{F}\{h(uT,\tau)\}|_{f=f_0-W/2+n\Delta f} = \sum_{l=0}^{L-1} A_{u,l} e^{-j2\pi(f_0-W/2+n\Delta f)l/W}, n = 1,\ldots,N_f, \quad (27)$$

where $A_{u,l}$ denotes the lth channel gain at time uT, respectively. Following the exponential decay model, which holds for the power delay profile of $\mathbf{q}_u$ based on various experimental observations [7], we model $A_{u,l}$ to be a zero-mean Gaussian random variable with variance $\text{Var}(A_{u,l}) = \alpha^2(1-e^{-2\pi\beta})e^{-2\pi\beta l}$. Here, α^2 and β denotes the average power of $A_{u,l}$ over all taps and the normalized coherence bandwidth, i.e, B_c/W, respectively. The distribution of $\mathbf{q}_u$ is given in the following lemma.

Lemma 4. *The vector $\mathbf{q}_u$ has a multivariate Gaussian distribution $\mathcal{CN}(\mathbf{0},\mathbf{R}_\mathbf{q})$ with a Toeplitz covarinace matrix $\mathbf{R}_\mathbf{q} = toep(\boldsymbol{\nu}_\mathbf{q},\boldsymbol{\nu}_\mathbf{q}^H)$ assuming*

$$\boldsymbol{\nu}_\mathbf{q} \triangleq \big[\kappa(\boldsymbol{\theta}_\mathbf{q},0),\kappa(\boldsymbol{\theta}_\mathbf{q},\frac{1}{N_f}),\ldots,\kappa(\boldsymbol{\theta}_\mathbf{q},1-\frac{1}{N_f})\big], \quad (28)$$

where $\kappa(\boldsymbol{\theta}_\mathbf{q},n) \triangleq \frac{\alpha^2(1-e^{-2\pi\beta})(1-e^{-2\pi L(\beta-nj)})}{(1-e^{-2\pi(\beta-nj)})}$ and $\boldsymbol{\theta}_\mathbf{q} = [\alpha^2,\beta,L]$.

Proof. See Appendix C. □

Next, the contribution of measurement noise $\mathbf{n}$ is modelled with a zero-mean complex multivariate Gaussian random variable as $\mathbf{n} \sim \mathcal{CN}(\mathbf{0},\sigma^2\mathbf{I})$ where σ^2 denotes the variance of the noise. We then follow the Kronecker model to obtain the covariance matrix of the CFR (19), which holds when the diffuse spectrum contribution in the angular domains is independent from that in the frequency domain [11], [44]. Under the Kronecker model, the covariance matrix of the CFR can be decomposed as $\mathbf{R} = \mathbf{I}_{N_{Rx}} \otimes \mathbf{I}_{N_{Tx}} \otimes \mathbf{R}_{\mathbf{q},\mathbf{n}}$ where $\mathbf{R}_{\mathbf{q},\mathbf{n}} = toep(\boldsymbol{\nu}_{\mathbf{q},\mathbf{n}},\boldsymbol{\nu}_{\mathbf{q},\mathbf{n}}^H)$ and $\boldsymbol{\nu}_{\mathbf{q},\mathbf{n}} \triangleq \boldsymbol{\nu}_\mathbf{q} + [\sigma^2,0,\ldots,0]$. Therefore, the distribution of the CFR in (19) within the above model can be given as $\mathbf{h}_u \sim \mathcal{CN}(\overline{\mathbf{h}},\mathbf{R})$. We denote the parameters associated with the covariance matrix by $\boldsymbol{\theta}_{vn} = [\alpha,\beta,L,\sigma]$, which corresponds to the variable part of the CFR and noise. As mentioned earlier, the mean $\overline{\mathbf{h}}$ solely depends on the specular paths parameters $\boldsymbol{\theta}_{sp}$.

B. Channel spoofing detection problem

Channel-based spoofing detection [7], [9] is generally studied in the "snapshot" scenario where Bob receives a new message claiming to be sent by Alice, and one needs to check whether the claim is true. To this end, we assume that Bob is able to measure and store a noisy version of the CFR corresponding to a transmitting terminal (Alice or Eve). Based on the CFRs associated with the incoming messages, and given a reference message $\mathbf{h}_u^A$ from Alice at time $t = uT$,[5] the goal in this scenario is to determine whether a message at time $t = (u+1)T$ belongs to Alice or Eve. In this setup, we use the terms message and CFR interchangeably. One can pose the spoofing detection problem as a binary classification problem for which two hypotheses can be stated

$$\mathcal{H}_0 : \mathbf{h}_{u+1} = \mathbf{h}_{u+1}^A, \quad (29)$$

$$\mathcal{H}_1 : \mathbf{h}_{u+1} = \mathbf{h}_{u+1}^E. \quad (30)$$

Under the null hypothesis, $\mathcal{H}_0$, the message at time $t = (u+1)T$ belongs to Alice, while under the alternative hypothesis, $\mathcal{H}_1$, a spoofing attack has occurred, i.e., the message belongs to Eve.

From a statistical perspective, likelihood ratio test (LRT) is the main approach for deciding between the two hypotheses, which relies on knowledge of the unknown channel parameters. The likelihood ratio test at time $t = (u+1)T$ for the snapshot scenario is given by

$$\mathbb{L}\big(\mathbf{h}_{u+1}|\mathbf{h}_u^A\big) = \frac{p(\mathbf{h}_{u+1}-\mathbf{h}_u^A|\mathcal{H}_0)}{p(\mathbf{h}_{u+1}-\mathbf{h}_u^A|\mathcal{H}_1)} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \zeta, \quad (31)$$

for a predefined threshold ζ, where the conditional probability distribution of $\mathbf{h}_{u+1}-\mathbf{h}_u^A$ [7] serves as the likelihood function under each behavior. In the following, we obtain closed-form expressions for these likelihood functions assuming $\mathbf{q}_{u+1}^A$ and $\mathbf{q}_{u+1}^E$ are the statistical dependence on $\mathbf{q}_u^A$. Specifically, we consider a case where the dependence of $\mathbf{q}_{u+1}^A$ on $\mathbf{q}_u^A$ is characterized through channel gains of the corresponding virtual paths in terms of an order-1 auto-regressive (AR-1) model [7], i.e.,

$$A_{u+1,l}^A = a^A A_{u,l}^A + \sqrt{(1-(a^A)^2)\text{Var}(A_{u+1,l}^A)}w_{u+1,l} \quad (32)$$

where a^A denotes the *similarity parameter* between $A_{u+1,l}^A$ and $A_{u,l}^A$, and $w_{u+1,l} \sim \mathcal{CN}(0,1)$ is independent of $A_{u,l}$. Similarly, gains of the lth virtual path corresponding to $\mathbf{q}_{u+1}^E$ and $\mathbf{q}_u^A$ are related to each other according to an AR-1 model with similarity parameter a^E.

The likelihood functions associated with the above LRT depend on the unknown channel parameters that needs to be estimated from finite number of training CFRs. These training data are collected by Bob during finite number of snapshots within a coherence time. In order to label a training data $\mathbf{h}_{u+1}$ at time $t = (u+1)T$ in the snapshot setting, we use a heuristic (and error prone) method given by

$$\|\mathbf{h}_{u+1}-\mathbf{h}_u^A\|^2 \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \eta. \quad (33)$$

which does not rely on the unknown channel parameters. As noted in [9], the threshold η can be chosen such that the resulting false alarm probability is below a predefined target value, e.g., 0.1. This method can be viewed as an imperfect labeling mechanism that decides in favor of $\mathcal{H}_0$ if the Euclidean distance between an incoming CFR and the reference CFR is smaller than a predefined threshold η.

[5] In the remainder of this section, we use A or E in the superscript of a vector or a scalar to indicate that it corresponds to Alice or Eve, respectively.

Lemma 5. *Under the null hypothesis,* $p(\mathbf{q}_{u+1} - \mathbf{q}_u^A|\mathcal{H}_0) = \mathcal{CN}(\mathbf{0}, \mathbf{R}_{q,\mathcal{H}_0})$ *for*

$$\mathbf{R}_{q,\mathcal{H}_0} = toep(\boldsymbol{\nu}_{\mathcal{H}_0}, \boldsymbol{\nu}_{\mathcal{H}_0}^H), \tag{34}$$

$$\boldsymbol{\nu}_{\mathcal{H}_0} \triangleq \Big[2(1-a^A)\kappa(\boldsymbol{\theta}_{\mathbf{q}}^A, 0), 2(1-a^A)\kappa\big(\boldsymbol{\theta}_{\mathbf{q}}^A, \frac{1}{N_f}\big), \ldots , 2(1-a^a)\kappa\big(\boldsymbol{\theta}_{\mathbf{q}}^A, \frac{N_f-1}{N_f}\big)\Big], \tag{35}$$

where $\boldsymbol{\theta}_{\mathbf{q}}^A \triangleq [\alpha^A, \beta^A, L^A]$ *and the* κ *function is defined in Lemma 4.*

Proof. This can be proved in a similar fashion to Lemma 4. See Appendix D of the long version of the current paper [1] for details. □

Lemma 6. *Under the alternative hypothesis,* $p(\mathbf{q}_{u+1} - \mathbf{q}_u^A|\mathcal{H}_1) = \mathcal{CN}(\mathbf{0}, \mathbf{R}_{\mathbf{q},\mathcal{H}_1})$ *for*

$$\mathbf{R}_{\mathbf{q},\mathcal{H}_1} = toep(\boldsymbol{\nu}_{\mathcal{H}_1}, \boldsymbol{\nu}_{\mathcal{H}_1}^H), \tag{36}$$

$$\boldsymbol{\nu}_{\mathcal{H}_1} \triangleq [\kappa'(a^E, \theta_{\mathbf{q}}^A, \theta_{\mathbf{q}}^E, 0), \kappa'(a^E, \theta_{\mathbf{q}}^A, \theta_{\mathbf{q}}^E, \frac{1}{N_f}), \ldots , \kappa'(a^E, \theta_{\mathbf{q}}^A, \theta_{\mathbf{q}}^E, \frac{N_f-1}{N_f})], \tag{37}$$

$$\kappa'(a^E, \theta_{\mathbf{q}}^A, \theta_{\mathbf{q}}^E, m) \triangleq \kappa(\theta_{\mathbf{q}}^E, m) - 2a^E\kappa(\theta_{\mathbf{q}}^A, m) + \kappa(\theta_{\mathbf{q}}^A, m), \tag{38}$$

where $\theta_{\mathbf{q}}^A \triangleq [\alpha^A, \beta^A, L^A]$, $\theta_{\mathbf{q}}^E \triangleq [\alpha^E, \beta^E, L^E]$, *and the* κ *function is defined in Lemma 4.*

Proof. See Appendix E of the long version of the paper [1]. □

The above two lemmas enable us to obtain the likelihood functions for (31), both of which are given by a Gaussian distribution. Regarding the null hypothesis $\mathcal{H}_0$, using the Kronecker model for the covariance matrix [11], we obtain the covariance matrix of $\mathbf{h}_{u+1} - \mathbf{h}_u^A$ as $\mathbf{R}_{\mathcal{H}_0} = \mathbf{I}_{N_{Rx}} \otimes \mathbf{I}_{N_{Tx}} \otimes \mathbf{R}_{\mathbf{q},\mathcal{H}_0} + 2(\sigma^A)^2\mathbf{I}_M$, where $\mathbf{R}_{\mathbf{q},\mathcal{H}_0}$ is given in Lemma 5. Furthermore, the contribution of the specular paths to the CFRs remains the same ($\overline{\mathbf{h}}^A$) between two consecutive times within a coherence time. Therefore, under $\mathcal{H}_0$ the likelihood function is $\mathcal{CN}(\mathbf{0}, \mathbf{R}_{\mathcal{H}_0})$. Similarly, for the alternate hypothesis $\mathcal{H}_1$, the likelihood function can be obtained as $\mathcal{CN}(\overline{\mathbf{h}}^E - \overline{\mathbf{h}}^A, \mathbf{R}_{\mathcal{H}_1})$, where $\mathbf{R}_{\mathcal{H}_1} = \mathbf{I}_{N_{Rx}} \otimes \mathbf{I}_{N_{Tx}} \otimes \mathbf{R}_{\mathbf{q},\mathcal{H}_1} + (\sigma^A)^2\mathbf{I}_M + (\sigma^E)^2\mathbf{I}_M$ and $\mathbf{R}_{\mathbf{q},\mathcal{H}_1}$ is given in Lemma 6.

C. Parameter estimation

In order to employ the likelihood ratio test in (31) or generate synthetic data for utilizing the HYPHYLEARN algorithm, Bob requires knowledge of the parameters $\boldsymbol{\theta}_{sp}$, $\boldsymbol{\theta}_{vn}$ corresponding to the Alice–Bob and Eve–Bob channels as well as the similarity parameters. These parameters need to be estimated based on the training data collected from finite number of snapshots. We denote the training CFRs associated with Alice and Eve by $\mathcal{D}_A = \{\mathbf{x}_i^A\}_{i=1}^{N_A}$ and $\mathcal{D}_E = \{\mathbf{x}_i^E\}_{i=1}^{N_E}$, respectively. Recall from Lemma 4 that entries of these datasets follow Gaussian distribution of the from $\mathcal{CN}\big(\overline{\mathbf{h}}_{\boldsymbol{\theta}_{sp}}, \mathbf{R}_{\boldsymbol{\theta}_{vn}}\big)$ where the subscripts in the mean and covariance are used to signify the dependence on a set of parameter. Based on the available likelihood function, we describe how the parameters $\boldsymbol{\theta}_{sp}$, $\boldsymbol{\theta}_{vn}$ associated with the Alice–Bob and Eve–Bob channels can be estimated from $\mathcal{D}_A$ and $\mathcal{D}_E$, respectively, in Section IV-C1. We further consider training datasets corresponding to the difference between an incoming CFR and the reference CFR from the observed snapshots. We denote the datasets consisting of the difference of the CFRs by $\mathcal{D}_{AA} = \{\mathbf{x}_i^{AA}\}_{i=1}^{N_A}$ and $\mathcal{D}_{EA} = \{\mathbf{x}_i^{EA}\}_{i=1}^{N_E}$ for Alice and Eve, respectively. The data samples in $\mathcal{D}_{AA}$ and $\mathcal{D}_{EA}$ follow the Gaussian distribution of the forms described in Lemmas 5 and 6, respectively. Subsequently, these likelihood functions are utilized to estimate the similarity parameters given the estimates of $\boldsymbol{\theta}_{sp}$, $\boldsymbol{\theta}_{vn}$ as described in Section IV-C2.

1) Estimating the parameters $\boldsymbol{\theta}_{sp}$ *and* $\boldsymbol{\theta}_{vn}$*:* Here, we discuss how the parameters $\boldsymbol{\theta}_{sp}^A$ and $\boldsymbol{\theta}_{vn}^A$ can be estimated for the Alice-Bob channel. The same procedure also holds for estimating the parameters associated with the Eve-Bob channels, i.e., $\boldsymbol{\theta}_{sp}^E$ and $\boldsymbol{\theta}_{vn}^E$. The ML estimates of these parameters for a sample CFR $\mathbf{h}$ can be obtained via

$$\widehat{\boldsymbol{\theta}}_{sp}^A, \widehat{\boldsymbol{\theta}}_{vn}^A \in \underset{\boldsymbol{\theta}_{sp}^A, \boldsymbol{\theta}_{vn}^A}{\operatorname{argmax}} \mathcal{L}\big(\mathbf{h}|\boldsymbol{\theta}_{sp}^A, \mathbf{R}_{\boldsymbol{\theta}_{vn}^A}\big), \tag{39a}$$

$$\mathcal{L}(\mathbf{h}|\boldsymbol{\theta}_{sp}^A, \mathbf{R}_{\boldsymbol{\theta}_{vn}^A}) = -M\ln\pi - \ln\det\mathbf{R}_{\boldsymbol{\theta}_{vn}^A} - \big(\mathbf{h} - \overline{\mathbf{h}}_{\boldsymbol{\theta}_{sp}^A}\big)^H\mathbf{R}_{\boldsymbol{\theta}_{vn}^A}^{-1}\big(\mathbf{h} - \overline{\mathbf{h}}_{\boldsymbol{\theta}_{sp}^A}\big), \tag{39b}$$

which amounts to jointly maximizing the arguments of a nonlinear objective function. It can be proved that (39b) is not a convex function of $\boldsymbol{\theta}_{sp}^A$, and as a result there is no unique solution set for the optimization problem in (39a). In practice, solving such a problem is far from trivial, especially since the objective function is a non-linear function of large number of parameters where multidimensional exhaustive search is not feasible. As a workaround, the authors in [11], [44] propose a suboptimal procedure to break the problem into two sub-problems and estimate $\boldsymbol{\theta}_{sp}^A$ and $\boldsymbol{\theta}_{vn}^A$ via alternate maximization. Each sub-problem involves numerically maximizing the objective function of the form (39b) with respect to $\boldsymbol{\theta}_{sp}^A$ or $\boldsymbol{\theta}_{vn}^A$ via an iterative local optimization technique such as the Gauss–Newton algorithm. In other words, the maximization processes are done sequentially over the dataset $\mathcal{D}_A$ and in an alternating manner between the two sets of parameters till convergence is achieved. In the following, we elaborate on each sub-problem for the specific channel model we described earlier.

We first describe how one can obtain an estimate of $\boldsymbol{\theta}_{sp}^A$ that maximizes (39b) for a given estimate of $\boldsymbol{\theta}_{vn}^A$. In the following, we use the N-exponential basis function defined as

$$\mathbf{U}_N^{\mathbf{v}} = \begin{bmatrix} e^{-j\left(-\frac{N-1}{2}\right)\mathbf{v}[1]} & \cdots & e^{-j\left(-\frac{N-1}{2}\right)\mathbf{v}[n]} \\ \vdots & \ddots & \vdots \\ e^{-j\left(\frac{N-1}{2}\right)\mathbf{v}[1]} & \cdots & e^{-j\left(\frac{N-1}{2}\right)\mathbf{v}[n]} \end{bmatrix}, \tag{40}$$

for a vector $\mathbf{v}$ of length N. The partial derivative of $\mathbf{U}_N^{\mathbf{v}}$ with respect to $\mathbf{v}$ is readily computed as $\mathbf{D}_N^{\mathbf{v}} = \frac{\partial\mathbf{U}_N^{\mathbf{v}}}{\partial\mathbf{v}} = -j\Xi_N\mathbf{U}_N^{\mathbf{v}}$, where $\Xi_N = \text{diag}([-(N-1)/2, \ldots, (N-1)/2])$. Furthermore, we recall that for arbitrary matrices $\mathbf{A} \in \mathbb{C}^{N\times P}$, $\mathbf{B} \in \mathbb{C}^{M\times P}$, $\mathbf{Q}^{P\times P} = \text{diag}(\mathbf{q})$ and a vector $\mathbf{q} \in \mathbb{C}^{P\times 1}$, one can write $\text{vec}\{\mathbf{BQA}^T\} = (\mathbf{A} \odot \mathbf{B})\mathbf{q}$. Utilizing this result along

with the exponential basis function we can rewrite the specular path contribution introduced in (25) for the CFR model as

$$\overline{\mathbf{h}} = \big(\mathbf{U}^{\boldsymbol{\psi}_T}_{N_{Rx}} \odot \mathbf{U}^{\boldsymbol{\psi}_R}_{N_{Tx}} \odot \mathbf{U}^{\boldsymbol{\tau}}_{N_f}\big)\boldsymbol{\rho}, \tag{41}$$

which greatly simplifies the calculation of the first and second derivatives of $\overline{\mathbf{h}}$ with respect to $\boldsymbol{\theta}^A_{sp}$. Specifically, the Jacobian matrix for the above model is obtained via $\mathbf{J}(\boldsymbol{\theta}_{sp}) = \mathbf{J}_{\boldsymbol{\psi}_T} \odot \mathbf{J}_{\boldsymbol{\psi}_R} \odot \mathbf{J}_{\rho} \odot \mathbf{J}_{\boldsymbol{\tau}}$ where the Jacobian matrix's components are given by

$$\mathbf{J}_{\boldsymbol{\psi}_T} = \begin{bmatrix} \mathbf{D}^{\boldsymbol{\psi}_T}_{N_{Tx}} & \mathbf{U}^{\boldsymbol{\psi}_T}_{N_{Tx}} & \mathbf{U}^{\boldsymbol{\psi}_T}_{N_{Tx}} & \mathbf{U}^{\boldsymbol{\psi}_T}_{N_{Tx}} & \mathbf{U}^{\boldsymbol{\psi}_T}_{N_{Tx}} \end{bmatrix}, \tag{42a}$$

$$\mathbf{J}_{\boldsymbol{\psi}_R} = \begin{bmatrix} \mathbf{U}^{\boldsymbol{\psi}_R}_{N_{Rx}} & \mathbf{D}^{\boldsymbol{\psi}_R}_{N_{Rx}} & \mathbf{U}^{\boldsymbol{\psi}_R}_{N_{Rx}} & \mathbf{U}^{\boldsymbol{\psi}_R}_{N_{Rx}} & \mathbf{U}^{\boldsymbol{\psi}_R}_{N_{Rx}} \end{bmatrix}, \tag{42b}$$

$$\mathbf{J}_{\boldsymbol{\tau}} = \begin{bmatrix} \mathbf{U}^{\boldsymbol{\tau}}_{N_f} & \mathbf{U}^{\boldsymbol{\tau}}_{N_f} & \mathbf{D}^{\boldsymbol{\tau}}_{N_f} & \mathbf{U}^{\boldsymbol{\tau}}_{N_f} & \mathbf{U}^{\boldsymbol{\tau}}_{N_f} \end{bmatrix}, \tag{42c}$$

$$\mathbf{J}_{\boldsymbol{\rho}} = \begin{bmatrix} \boldsymbol{\rho}^T & \boldsymbol{\rho}^T & \boldsymbol{\rho}^T & \mathbf{1}^T & \mathbf{1}^T j \end{bmatrix}. \tag{42d}$$

The authors in [44] compute the first-order partial derivative, $\mathbf{q}_{\boldsymbol{\theta}^A_{sp}}\big(\mathbf{h}|\mathbf{R}_{\boldsymbol{\theta}^A_{vn}}\big)$, and the Fisher information matrix (FIM), $\mathbf{F}\big(\boldsymbol{\theta}^A_{sp}|\mathbf{R}_{\boldsymbol{\theta}^A_{vn}}\big)$, of the log likelihood function (39b), with respect to the parameter $\boldsymbol{\theta}^A_{sp}$ for a given observation $\mathbf{h}$, as the following

$$\mathbf{q}_{\boldsymbol{\theta}^A_{sp}}\big(\mathbf{h}|\mathbf{R}_{\boldsymbol{\theta}^A_{vn}}\big) = 2\Re\big\{\mathbf{J}^H(\boldsymbol{\theta}^A_{sp})\mathbf{R}^{-1}_{\boldsymbol{\theta}^A_{vn}}(\mathbf{h} - \overline{\mathbf{h}}_{\boldsymbol{\theta}^A_{sp}}\big\}, \tag{44}$$

$$\mathbf{F}\big(\boldsymbol{\theta}^A_{sp}|\mathbf{R}_{\boldsymbol{\theta}^A_{vn}}\big) = 2\Re\{\mathbf{J}^H(\boldsymbol{\theta}^A_{sp})\mathbf{R}^{-1}_{\boldsymbol{\theta}^A_{vn}}\mathbf{J}(\boldsymbol{\theta}^A_{sp})\}. \tag{45}$$

Based on the above computations, a local optimization technique is utilized in [44] to obtain an iterative rule for estimating $\boldsymbol{\theta}^A_{sp}$. For the experiments we present in Section VI, we employ the Gauss–Newton algorithm as

$$\widehat{\boldsymbol{\theta}}^{A,i+1}_{sp} = \widehat{\boldsymbol{\theta}}^{A,i}_{sp} + \zeta\ \mathbf{F}^{-1}\big(\widehat{\boldsymbol{\theta}}^{A,i}_{sp}|\mathbf{R}_{\boldsymbol{\theta}_{vn}}\big)\mathbf{q}_{\widehat{\boldsymbol{\theta}}^{A,i}_{sp}}\big(\mathbf{h}|\mathbf{R}_{\boldsymbol{\theta}_{vn}}\big) \tag{46}$$

for a step length ζ that should be chosen such that $\mathcal{L}(\mathbf{h}|\boldsymbol{\theta}^{A,i+1}_{sp}, \mathbf{R}_{\boldsymbol{\theta}^A_{vn}}) > \mathcal{L}(\mathbf{h}|\boldsymbol{\theta}^{A,i}_{sp}, \mathbf{R}_{\boldsymbol{\theta}^A_{vn}})$. By applying this procedure to all the training CFRs in $\mathcal{D}_A$, we obtain N_A estimates as $\{\widehat{\boldsymbol{\theta}}^A_{sp,i}\}_{i=1}^{N_A}$ whose average value is denoted by $\bar{\boldsymbol{\theta}}^A_{sp}$ in the following.

After obtaining $\bar{\boldsymbol{\theta}}^A_{sp}$, the maximization process alternates in order to estimate $\boldsymbol{\theta}^A_{vn}$. To this end, first the contribution of the specular paths from the CFRs in $\mathcal{D}_A$ is removed by subtracting $\overline{\mathbf{h}}_{\bar{\boldsymbol{\theta}}^A_{sp}}$ from each training data. Subsequently, these new data entries are stacked up to form an $M \times N_A$ matrix $\mathbf{H}$ which, in the following, will be used in order to estimate $\boldsymbol{\theta}^A_{vn}$. We first note that all the parameters in $\boldsymbol{\theta}^A_{vn}$ are continuous values except for the number of diffuse virtual paths L^A that takes on integer values. As a result, the objective function in (39b) is not continuous in L^A and the partial derivative of (39b) does not exist with respect to L^A. In order to overcome this challenge, we further take a sub-optimal approach and estimate L^A in a separate manner from the rest of the parameters in $\boldsymbol{\theta}^A_{vn}$. To this end, we use an eigenvalue ratio method described in [45] that estimates the number of harmonics present in a given set of observations. Following this approach, we first obtain the MLE of the covariance of $\mathbf{H}$ and denote it by $\mathbf{C}_{\mathbf{H}}$. The eigenvalues of $\mathbf{C}_{\mathbf{H}}$ are further denoted by $\mathbf{e}_i,\ i = 1, \ldots, M$. Then, we choose $\widehat{L}^A$ in a way that $\frac{\sum_{i=1}^{\widehat{L}^A}\mathbf{e}_i}{\sum_{i=1}^{M}\mathbf{e}_i} \geq \eta$, for a predefined value of η commonly chosen to be in the range $[0.85, 0.95]$.

We plug-in the estimated value of L^A in the parameter vector to obtain $\boldsymbol{\theta}^A_{vn} = [\sigma^A, \alpha^A, \beta^A, \widehat{L}^A]$. Then, the log-likelihood function for the zero-mean CFRs can be written as

$$\mathcal{L}(\mathbf{H}|\boldsymbol{\theta}^A_{vn}) = -MN_A \ln \pi - N_A \ln\det \mathbf{R}_{\boldsymbol{\theta}^A_{vn}} - \mathrm{Tr}\left(\mathbf{H}^H\mathbf{R}^{-1}_{\boldsymbol{\theta}^A_{vn}}\mathbf{H}\right). \tag{47}$$

The first-order partial derivative of $\mathcal{L}(\mathbf{H}|\boldsymbol{\theta}^A_{vn})$ with respect to each parameter can be computed as [44]

$$\frac{\partial\mathcal{L}(\mathbf{H}|\boldsymbol{\theta}^A_{vn})}{\partial\boldsymbol{\theta}^A_{vn}[i]} = N_A\,\mathrm{Tr}\left(\mathbf{R}^{-1}_{\boldsymbol{\theta}^A_{vn}}\frac{\partial\mathbf{R}_{\boldsymbol{\theta}^A_{vn}}}{\partial\boldsymbol{\theta}^A_{vn}[i]}\mathbf{R}^{-1}_{\boldsymbol{\theta}^A_{vn}}(\widehat{\mathbf{R}} - \mathbf{R}_{\boldsymbol{\theta}^A_{vn}})\right) \tag{48}$$

for $i = 1, 2, 3$. Subsequently, the (i, j)th element of the FIM corresponding to $\mathcal{L}(\mathbf{H}|\boldsymbol{\theta}_{vn})$ equals [44]

$$-\mathbb{E}\Big[\frac{\partial^2\mathcal{L}(\mathbf{H}|\boldsymbol{\theta}^A_{vn})}{\partial\boldsymbol{\theta}^A_{vn}[i]\partial\boldsymbol{\theta}^A_{vn}[j]}\Big] = N_A\,\mathrm{Tr}\left(\mathbf{R}^{-1}_{\boldsymbol{\theta}^A_{vn}}\frac{\partial\mathbf{R}_{\boldsymbol{\theta}^A_{vn}}}{\partial\boldsymbol{\theta}^A_{vn}[i]}\mathbf{R}^{-1}_{\boldsymbol{\theta}^A_{vn}}\frac{\partial\mathbf{R}_{\boldsymbol{\theta}^A_{vn}}}{\partial\boldsymbol{\theta}^A_{vn}[j]}\right). \tag{49}$$

To obtain explicit expressions for (48) and (49), one needs to compute the partial derivative terms $\frac{\partial\mathbf{R}_{\boldsymbol{\theta}_{vn}}}{\partial\boldsymbol{\theta}_{vn}[i]}$. Considering the Toeplitz structure of the covariance model described in Lemma 4, we can write

$$\frac{\partial\mathbf{R}_{\mathbf{q},\mathbf{n}}(\boldsymbol{\theta}_{vn})}{\partial\boldsymbol{\theta}^A_{vn}[i]} = toep\Big(\frac{\partial\boldsymbol{\nu}_{\mathbf{q},\mathbf{n}}}{\partial\boldsymbol{\theta}^A_{vn}[i]}, \frac{\partial\boldsymbol{\nu}^H_{\mathbf{q},\mathbf{n}}}{\partial\boldsymbol{\theta}^A_{vn}[i]}\Big), \tag{50}$$

$$\frac{\partial\mathbf{R}_{\boldsymbol{\theta}^A_{vn}}}{\partial\boldsymbol{\theta}^A_{vn}[i]} = \mathbf{I}_{N_{Rx}} \otimes \mathbf{I}_{N_{Tx}} \otimes \frac{\partial\mathbf{R}_{\mathbf{q},\mathbf{n}}(\boldsymbol{\theta}^A_{vn})}{\partial\boldsymbol{\theta}^A_{vn}[i]}, \tag{51}$$

where the partial derivative for each parameter is obtained in (43a)-(43c) for $f(m) = e^{-2\pi(\beta - jm)}$. Plugging this in (48) and (49) leads to computation of first-order partial derivative and the FIM of the likelihood function. Then, an iterative approach like the Gauss–Newton algorithm can be employed for estimating $\boldsymbol{\theta}^A_{vn}$ in a similar fashion to the case of $\boldsymbol{\theta}_{sp}$ in (46). Afterwards, the maximization process further alternates to estimate the parameters $\widehat{\boldsymbol{\theta}}^A_{sp,i}$ using $\widehat{\boldsymbol{\theta}}_{vn}$.

$$\frac{\partial\boldsymbol{\nu}_{\mathbf{q},\mathbf{n}}}{\partial\sigma} = \big[2\sigma, 0, \ldots, 0\big], \tag{43a}$$

$$\frac{\partial\boldsymbol{\nu}_{\mathbf{q},\mathbf{n}}}{\partial\alpha} = 2\alpha\left[1 - e^{-2\pi L\beta}, \frac{(1-e^{-2\pi\beta})\left(1-f^L(\frac{1}{N_f})\right)}{1-f(\frac{1}{N_f})}, \ldots, \frac{(1-e^{-2\pi\beta})\left(1-f^L(1-\frac{1}{N_f})\right)}{1-f(1-\frac{1}{N_f})}\right], \tag{43b}$$

$$\frac{\partial\boldsymbol{\nu}_{\mathbf{q},\mathbf{n}}}{\partial\beta} = \Big[2\pi\alpha^2 L e^{-2\pi\beta L}, \frac{2\pi e^{-2\pi\beta}\left(f^L(\frac{1}{N_f})-1\right)}{f(\frac{1}{N_f})-1} + \frac{2L\pi f^L(\frac{1}{N_f})(e^{-2\pi\beta}-1)}{f(\frac{1}{N_f})-1} - \frac{2\pi f(\frac{1}{N_f})\left(f^L(\frac{1}{N_f})-1\right)(e^{-2\pi\beta}-1)}{\left(f(\frac{1}{N_f})-1\right)^2},$$
$$\ldots, \frac{2\pi e^{-2\pi\beta}\left(f^L(1-\frac{1}{N_f})-1\right)}{f(1-\frac{1}{N_f})-1} + \frac{2L\pi f^L(1-\frac{1}{N_f})(e^{-2\pi\beta}-1)}{f(1-\frac{1}{N_f})-1} - \frac{2\pi f(1-\frac{1}{N_f})\left(f^L(1-\frac{1}{N_f})-1\right)(e^{-2\pi\beta}-1)}{\left(f(1-\frac{1}{N_f})-1\right)^2}\Big]. \tag{43c}$$

2) Estimating the similarity parameters: We now describe how a^A can be estimated based on the available dataset $\mathcal{D}_{AA}$ and the likelihood function in Lemma 5. Similar approach can be taken for estimating a^E based on $\mathcal{D}_{EA}$ and Lemma 6. Assuming an $M \times N_A$ matrix $\mathbf{H}_{AA}$ is formed out of the dataset $\mathcal{D}_{AA}$, the MLE of a^A given $\mathbf{H}_{AA}$ can be obtained via

$$\widehat{a}^A \in \underset{a^A}{\operatorname{argmax}} \quad \Big(N_A \ln\det \mathbf{R}_{\mathcal{H}_0} - \operatorname{Tr}\left(\mathbf{H}_{AA}^H \mathbf{R}_{\mathcal{H}_0}^{-1} \mathbf{H}_{AA}\right)\Big). \quad (52)$$

We note that the estimates of the parameters $\boldsymbol{\theta}_{sp}^A$ and $\boldsymbol{\theta}_{vn}^A$ are plugged in $\mathbf{R}_{\mathcal{H}_0}$ which makes $\mathbf{R}_{\mathcal{H}_0}$ a function of only a^A in the above maximization problem. Specifically, as a^A appears in the covariance matrix of a Gaussian distribution, a similar estimation procedure to that of $\boldsymbol{\theta}_{vn}$ can be employed here as well. In fact, the expressions for the first-order partial derivative and the FIM of the likelihood function in this case are similar to those in (48) and (49), respectively, except for the fact that there is only one parameter to estimate in this case. By considering the Toeplitz structure of the covariance model described in Lemma 5, we can write

$$\frac{\partial \mathbf{R}_{\mathbf{q},\mathcal{H}_0}(a^A)}{\partial a^A} = toep\Big(\frac{\partial \boldsymbol{\nu}_{\mathcal{H}_0}}{\partial a^A}, \frac{\partial \boldsymbol{\nu}_{\mathcal{H}_0}^H}{\partial a^A}\Big), \quad (53)$$

$$\frac{\partial \mathbf{R}_{\mathcal{H}_0}(a^A)}{\partial a^A} = \mathbf{I}_{N_{Rx}} \otimes \mathbf{I}_{N_{Tx}} \otimes \frac{\partial \mathbf{R}_{\mathbf{q},\mathcal{H}_0}(a^A)}{\partial a^A}, \quad (54)$$

where the partial derivative can be obtained as

$$\frac{\partial \boldsymbol{\nu}_{\mathcal{H}_0}}{\partial a^A} = -2\Big[\frac{(\alpha^A)^2(1-e^{-2\pi\beta^A})(1-f^{L^A}(0))}{1-f(0)}, \frac{(\alpha^A)^2(1-e^{-2\pi\beta^A})(1-f^{L^A}(\frac{1}{N_f}))}{1-f(\frac{1}{N_f})}, \ldots, \frac{(\alpha^A)^2(1-e^{-2\pi\beta^A})(1-f^{L^A}(1-\frac{1}{N_f}))}{1-f(1-\frac{1}{N_f})}\Big], \quad (55)$$

for $f(m) = e^{-2\pi(\beta - jm)}$. Subsequently, using the first-order partial derivative and the FIM of the likelihood function, the Gauss–Newton algorithm can be employed to estimate a^A.

D. HyPhyLearn *for channel spoofing detection*

As an alternative to the likelihood ratio-based approach of Section IV-B for channel spoofing detection problem, we propose to utilize HyPhyLearn algorithm, listed in Algorithm 1. This problem is an instance of the setting introduced in Section II as the statistical parametric models are available for each behavior, the high complexity of which makes one to resort to suboptimal parameter estimation procedure. As mentioned in Section IV-B the data corresponding to Alice and Eve are collected in the snapshot setting, and subsequently (imperfectly) labeled according to (33). Then, using these collected CFRs, the underlying parameters of each likelihood function in (31) are estimated. Next, the estimated parameters are plugged in the available parametric models $\mathcal{CN}(\mathbf{0}, \mathbf{R}_{\mathcal{H}_0})$ and $\mathcal{CN}(\overline{\mathbf{h}}^E - \overline{\mathbf{h}}^A, \mathbf{R}_{\mathcal{H}_1})$, which subsequently are used to generate synthetic CFRs. Finally, the collected and synthetic CFRs are incorporated in Step 4 of Algorithm 1 for the joint learning of the classifier, utilized as a spoofing detector, and the feature map. In Section VI, we present numerical results to show the superiority of HyPhyLearn compared to the other existing methods through various experiments.

V. Case Study II: Multi-user Detection

As the second case study, we consider the optimum centralized demodulation of the information sent simultaneously by several users through a Gaussian multiple-access channel which is an important problem in multipoint-to-point digital communication networks (e.g., radio networks, local-area networks, and uplink satellite channels). Even though the users may not employ a protocol to coordinate their transmission epochs, effective sharing of the channel is possible because each user modulates a different signature signal waveform. In this section, we consider the uplink of a cellular communication system where K users are asynchronously sharing a channel to communicate with a base station (BS). The problem of multi-user detection (MUD) in this setting amounts to inferring the information bit associated with each user from a received signals in the multiple access channel.

A. Multi-user detection problem

Consider the uplink of an asynchronous direct-sequence (DS) Code Division Multiple Access (CDMA) system shared by K users, employing long spreading codes, bandlimited chip pulses and operating over a frequency-selective fading channel. Baseband equivalent of the received signal may be written as

$$r(t) = \sum_{p=0}^{P-1} \sum_{k=0}^{K-1} A_k b_k(p) s'_{k,p}(t - \tau_k - pT_b) * c_k(t) + w(t), \quad (56)$$

where $*$ denotes the convolution operation, P is the number of transmitted packets and $s'_{k,p}(t)$ denotes the kth user signature waveform. Furthermore, T_b is the bit-interval duration, A_k and τ_k denote the respective complex amplitude and timing offset of kth user, and $b_k(p)$ is the kth user's information bit in the pth signaling interval, whereas $w(t)$ is the complex envelope of the additive noise term, which is assumed to be a zero-mean, wide-sense stationary complex white Gaussian process. Moreover, $c_k(t)$ is the impulse response modeling the channel effects between the BS and the kth user. We assume the channel impulse response (CIR), $c_k(t)$, takes the form of a time-invariant multipath channel with L paths, i.e., $c_k(t) = \sum_{l=0}^{L-1} \alpha_{k,l} \delta(t - \tau'_{k,l})$, which is parameterized by the complex path gains $\alpha_{k,l}$ and the corresponding path delays $\tau'_{k,l}$. Note that $c_k(t)$ is assumed to be time-invariant over each transmitted frame under the assumption that the channel coherence time exceeds the packet duration PT_b. Regarding the kth user signature waveform, we have

$$s'_{k,p}(t - \tau_k - pT_b) = \sum_{n=0}^{N-1} \beta_{k,p}^{(n)} h_{SRRC}(t - nT_c), \quad (57)$$

where $\{\beta_{k,p}^{(n)}\}_{n=0}^{N-1}$ is the pseudo-noise (PN) code employed by user k for spreading its data bit on the pth symbol interval, N is the processing gain, and $T_c = T_b/N$ is the chip interval. Furthermore, $h_{SRRC}(t)$ denotes the square root raised-cosine waveform as the bandlimited chip pulse which is, following [46], time-limited to $[0, 4Tc]$.

At the BS, chip-matched filtering and chip-rate sampling is done in order to convert the received signal to discrete time domain. To this end, $r(t)$ is convolved with chip-matched filter

$h_{SRRC}(4T_c - t)$ followed by sampling at a rate $2/T_c$ (Nyquist rate). This results in

$$y(t) = r(t) * h_{SRRC}(4T_c - t) = \sum_{p=0}^{P-1} \sum_{k=0}^{K-1} b_k(p) h_{k,p}(t - pT_b, \tau_k) + n(t), \quad (58)$$

where $h_{k,p}(t, \tau_k) = A_k s_{k,p}(t-\tau_k)^* c_k(t)$ is called the effective signature waveform for $s_{k,p}(t) = \sum_{n=0}^{N-1} \beta_{k,p}^{(n)} h_{RC}(t - nT_c)$, and $h_{RC}(t)$ represents a raised cosine chip waveform time-limited to $[0, 8Tc)$. As $h_{k,p}(t - pT_b, \tau_k)$ has a time-domain support of $[pT_b, (p+2)T_b + 7T_c]$ during the pth symbol interval $\mathcal{I}_p = [pT_b, (p+1)T_b]$, the contribution from at most three bits for each user, i.e., the pth, the $(p-1)$th and the $(p-2)$th ones, is relevant assuming that $\tau_k + T_m < T_b$, where T_m stands for the maximum delay spread among all the K users. Therefore, sampling the waveform $y(t)$ at rate M/T_c, the MN-dimensional vector $y(p)$ collecting the data samples of the interval $\mathcal{I}_p$ can be expressed as

$$\mathbf{y}(p) = \sum_{k=0}^{K-1} [b_k(p-2)\mathbf{h}_{k,p-2}(p) + b_k(p-1)\mathbf{h}_{k,p-1}(p) + b_k(p)\mathbf{h}_{k,p}(p)] + \mathbf{n}(p), \quad (59)$$

where $\mathbf{h}_{k,p-i}(p)$ and $\mathbf{n}(p)$ comprise the MN samples of $h_{k,p-i}(t - (p-i)T_b, \tau_k)$, $i \in \{0, 1, 2\}$, and $n(t)$, respectively, during $\mathcal{I}_p$. We set $M = 2$ in the following discussion. A compact representation of $\mathbf{y}(p)$ can be obtained by relying on the notion of effective chip pulse defined as $g_k(t, \tau_k) = A_k h_{RC}(t - \tau_k)^* c_k(t)$, which is supported on the interval $[0, T_b + 8T_c]$. Noting that $h_{k,p}(t, \tau_k) = \sum_{i=0}^{N-1} \beta_{k,p}^n g_k(t - nT_c, \tau_k)$, and defining $\mathbf{g}_k \in \mathbb{C}^{MN+8M-1\times 1}$ as $\mathbf{g}_k = \left[g_k(T_c/M, \tau_k), g_k(2T_c/M, \tau_k), \ldots, g_k(T_b + (8M-1)T_c/M, \tau_k)\right]^T$, one can write $\mathbf{h}_{k,p-i}(p) = \mathbf{C}_{k,p-i}(p)\mathbf{g}_k$, where $\mathbf{C}_{k,p-i}(p)$ is a $MN \times (MN + 8M - 1)$ dimensional matrix that is a function of $\beta_{k,p}^n$, obtained in details in (9)–(11) of [46]. Then, we have

$$\mathbf{y}(p) = \sum_{k=0}^{K-1} \mathbf{A}_k(p)\mathbf{g}_k + \mathbf{n}(p) = \mathbf{A}(p)\mathbf{g} + \mathbf{n}(p), \quad (60)$$

for $\mathbf{A}_k(p) = b_k(p-2)\mathbf{C}_{k,p-2}(p) + b_k(p-1)\mathbf{C}_{k,p-1}(p) + b_k(p)\mathbf{C}_{k,p}(p)$, $\mathbf{A}(p) = [\mathbf{A}_0(p), \ldots, \mathbf{A}_{K-1}(p)]$, and $\mathbf{g} = [\mathbf{g}_0^T, \ldots, \mathbf{g}_{K-1}^T]^T$. The elements of the noise vector, $\mathbf{n}(p)$, are independent and identically distributed (i.i.d.) as a zero-mean Gaussian with a variance $N_0/2$, which lead to a signal to noise ratio (SNR) of A_k^2/N_0 for the kth user.

It follows from this discussion that the MUD problem can be cast as 2^K-ary classification problem where the goal is to find the vector of information bits $\mathbf{b} = [b_0(p), \ldots, b_{K-1}(p)]$ given an observation vector $\mathbf{y}(p)$. Assuming all the vectors $\mathbf{b} \in \{0, 1\}^K$ are a priori equiprobable the minimum distance rule gives the maximum a posteriori decision [47]. Mathematically, the MUD is equivalent to solving the minimization problem $\text{argmin}_{\mathbf{b}\in\{0,1\}^K} \ \mathbf{y}(p) - \sum_{k=0}^{K-1} \mathbf{A}_k(p)\mathbf{g}_k$. However, the complexity of such detector is exponential in the number of users [47] and in practice sub-optimal methods like minimum mean square error (MMSE) detector [47] are utilized in practice. We consider a case where the BS has access to N_K number of training data from the users in the form of $\mathcal{D} = \{\mathbf{y}_i, \mathbf{b}_i\}_{i=1}^{N_K}$, where $\mathbf{y}_i$ has the form of (60) and $\mathbf{b}_i$ denotes the corresponding information bits vector. We further assume that BS does not have access the perfect knowledge of the true spreading codes from all the users in a similar scenario to blind MUD [48].

B. Parameter estimation

The performance of the above MUD algorithms relies heavily on the estimation of the channel parameters. It is shown in [49] the joint MLE of these parameters requires an exhaustive search over the continuous K-dimensional space $[0, T_b)^K$, which imposes an exponentially increasing complexity in K when the conventional grid search-based scheme is utilized. As a workaround, alternative sub-optimal estimation methods of low-complexity are proposed to be used for practical systems. Notably, the authors in [46] propose a two-step approach that first estimates the samples of effective chip pulse $\mathbf{g}$ using the Least Squares (LS) criterion, and then extracts the underlying channel parameters. In particular, given the knowledge of the spreading codes and information bits for all the users in the training dataset, the vector $\mathbf{g}$ may be directly estimated by invoking the LS estimation procedure $\widehat{\mathbf{g}} = \text{argmin}_{\mathbf{x}} \sum_{i=1}^{N_K} ||\mathbf{y}_i - \mathbf{A}(i)\mathbf{x}||^2$. Relying on $\widehat{\mathbf{g}}$. the authors in [46] propose an ad-hoc algorithm to estimate the channel parameters. Specifically, the explicit parameters to be estimated include delays $\tau_{k,l} = \tau'_{k,l} + \tau_k$, amplitudes $a_{k,l} = A_k|\alpha_{k,l}|$ and the phases $\phi_{k,l} = \arg(a_{k,l})$ for $k = 0, \ldots, K-1$ and $l = 0, \ldots, L-1$. We refer to the readers to Appendix F of the long version of the current paper [1] for details of this parameter estimation method.

C. HyPhyLearn *for multi-user detection*

As an alternative to classical methods, we can utilize HyPhyLearn to solve the problem of MUD as a 2^K-ary classification problem. In particular, since we have access to precise statistical parametric models for each class and we lack access to an estimation procedure for the underlying channel parameters that is both optimal and tractable, the MUD can be framed within the setting described in Section II. Indeed, we can use the available training data corresponding to the users in the suboptimal estimation method described in Section V-B to obtain the estimates of the channel parameters for K users $\widehat{\boldsymbol{\tau}} = [\widehat{\tau}_{0,0}, \ldots, \widehat{\tau}_{0,L-1}, \ldots, \widehat{\tau}_{K-1,L-1}]$, $\widehat{\mathbf{a}} = [\widehat{a}_{0,0}, \ldots, \widehat{a}_{0,L-1}, \ldots, \widehat{a}_{K-1,K-1}]$ and $\widehat{\boldsymbol{\Phi}} = [\widehat{\phi}_{0,0}, \ldots, \widehat{\phi}_{0,L-1}, \ldots, \widehat{\phi}_{K-1,L-1}]$. Using these estimates along with the imperfect knowledge of spreading codes for the training data, we can then employ the parametric model (60) to generate a synthetic data example associated with the sequence of utilized information bits $\mathbf{b}$. This synthetic data sample is subsequently added to a synthetic dataset along with its corresponding label $\mathbf{b}$. Here, the learning-based classifier in HyPhyLearn has 2^K output neurons, each corresponding to a specific information bits vector, which enables it to to serve as a MUD method for the K-user system.

VI. Numerical Results

In this section, we numerically evaluate the performance of our proposed solution, HyPhyLearn, described in Algorithm 1 for the two case studies described in Sections IV

and V. This involves comparing the resulting performance against that of the existing statistical classifiers and other hybrid classification methods, and highlighting the superiority of our proposed solution for the problems under study.

A. *Spoofing detection problem*

In the Alice–Eve–Bob setting, we begin with a scenario where the coherence time of the Alice–Bob and the Eve–Bob channel are very large, and therefore the corresponding channel parameters are fixed between the training and testing stages. As mentioned in Section IV-B, the training data in this problem are collected by observing finite number of snapshots by Bob. The training CFRs from each snapshot are subsequently labeled using the heuristic test (33). The number of received antennas and transmit antennas at Alice and Bob is set to 2. Also, following the discussion in [9] we assume Eve also uses the same number of antennas to impersonate Alice. The number of subcarriers is set to $N_f = 20$, which makes the total number of samples associated with each CFR equal $M = 80$. We assume the Alice–Bob parameters are $\sigma_A^2 = 20$, $\alpha_A^2 = 200$, $\beta_A = 0.02$ and $a^A = 0.85$, while $\sigma_E^2 = 26$, $\alpha_E^2 = 250$, $\beta_E = 0.08$ and $a^E = 0.65$ are used for the Eve–Bob channel. Furthermore, we set $L_A = 20$ and $L_E = 16$ as the number of diffuse spectrum virtual paths, while the number of specular paths are set to 4 for both channels in accordance with the experimental measurements reported in [10].

Fig. 3 illustrates the spoofing detection performance of different methods for the above scenario averaged over 10^5 CFRs from each Alice-Bob and Eve-Bob channel at the test stage, where the x-axis denotes the number of snapshots observed during the training stage. In particular, we have evaluated the performance of HYPHYLEARN for this problem, as described in Section IV-D, and compared it with other classifiers designed based on the likelihood ratio test with plug-in estimates or existing ML algorithms. By looking at the resulting spoofing detection accuracy, it can be seen that the performance of the ML algorithms based on support vector machine (SVM) and Gaussian mixture model (GMM) is limited in this case due to limited (and mislabeled) training data. We note that the GMM is used as a classifier here by assigning labels to the clusters using the available labels corresponding to the reference CFRs. Specifically, we have used the radial basis function kernel [39] for the SVM and two components for the GMM for these simulations. Furthermore, one can see that the LRT method obtained in Section IV-B can improve upon the performance of these ML algorithms by plugging the estimated parameters, as in Section IV-C, in the statistical parametric models. In these experiments, we also use the shrinkage method [50] which improves the covariance matrix estimation for each likelihood function. For this method, a performance gain can be observed for this approach in comparison to the no shrinkage case, assuming the shrinkage parameter α is clarivoyantly chosen to maximize the spoofing detection accuracy over the test dataset. This method is labeled as 'LRT (best shrinkage)' in Fig. 3. However, in practice the parameter α has to be estimated from the training data, which—as shown in the figure with label 'LRT (shrinkage)'—could deteriorate the LRT performance as the available data includes mislabeled samples.

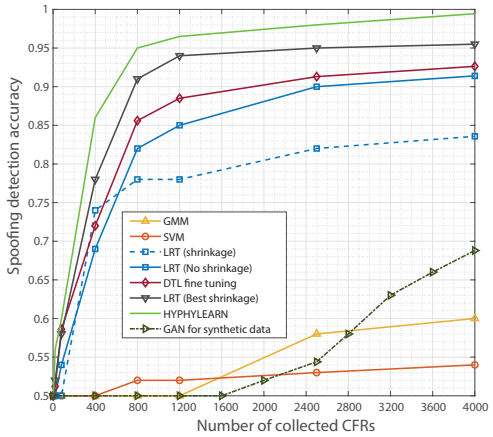


Fig. 3: Spoofing detection accuracy for different classification algorithms as a number of available training data for the case when training and test stage belong to the same coherence time.

Furthermore, we evaluate the performance of an existing hybrid classification approach known as fine tuning [4], [26] in DTL literature for this problem. In this method, we first generate 5×10^5 synthetic data samples using the available likelihood parametric functions with plugged-in estimates. Then, a neural network with 3 hidden layers of 400 neurons each is trained to classify the synthetic data for this example. The training data are used afterwards to refine the weights of this neural network. Notably, HYPHYLEARN is shown to outperform the aforementioned existing classification methods by relying on both available and synthetic data and jointly using them in a learning-based classifier.

For the sake of comparison, we have also considered a variation of HYPHYLEARN that relies on a generative adversarial network (GAN) for generating synthetic data, i.e., it disregards the available physic-based models. We have observed that the performance of this approach is impacted in the limited data regime as GANs rely merely on the available training data for generating further synthetic data of similar distribution. In fact, for this example, we have verified that HYPHYLEARN based on GAN needs to be trained on 20000 data samples in order to achieve the same level of spoofing detection accuracy as HYPHYLEARN based on physics-based models with 4000 samples. Regrading the specifics of GAN, we have used a DNN of two hidden layers with 200 neurons each as the generator, and a DNN with three hidden layers with 300 neurons each as the discriminator. In our implementation of HYPHYLEARN, the number of generated synthetic data samples is set to 4×10^5. We have also used NNs with 3 hidden layers of 400 neurons each for $\mathbf{M}_{\psi}$ and h_{ϕ_1}, while a NN with one hidden layer of 40 neurons each is used for d_{ζ}. For all hidden layers, the ReLU activation function is used. Furthermore, Adam optimizer [39] with a learning rate of 0.0001 is used for training in this example. We also note that the optimal Bayes decision rule, which relies on the knowledge of the true parameters, results in the spoofing detection accuracy of 0.996.

Next, we consider a more realistic scenario where the channels' variations cause the training and test stage to not

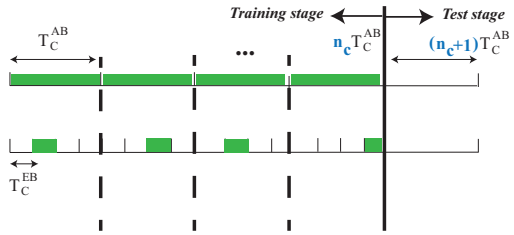


Fig. 4: Training and testing stages for the spoofing detection problem. T_C^{AB} and T_C^{EB} denote the coherence time corresponding to the Alice-Bob and Eve-Bob channels, respectively. The green bar indicates the time interval within which a snapshot is observed by Bob.

fall in the same coherence time. In this case, Bob uses the heuristic test (33) for some time as it does not have access to the channel parameters in this period. Afterwards, it uses the data collected in the previous coherence times to estimate the channel parameters for the current one. Fig. 4 depicts this setting where the training stage consists of n_c coherence times corresponding to the Alice–Bob channel. Furthermore, in contrast to Alice, Eve's transmissions are assumed to be intermittent due to the uncertainty associated with Eve's behaviour. During each coherence time corresponding to the Alice-Bob channel, it is assumed that Bob collects 100 training data. Then, the estimation technique described in Section IV-C is utilized to estimate the channel parameters under each coherence time. Fig. 5 demonstrates the system performance as a function of number of coherence times in the training stage. Regarding the physical setup, we have used the same system parameters as those in Fig. 3, and assumed that the coherence time of the Alice–Bob channel is 4 times that of the Eve–Bob channel for illustrative purpose. For DTL fine-tuning approach and HyPhyLearn, the number of synthetic data generated for each behavior in a coherence time is set to 20000. For these two learning-based approaches, the training specifications for are chosen to be the same as the ones used in Fig. 3. The performance comparison again highlights the superiority of HyPhyLearn in comparison to the existing statistical and data-driven methods.

B. Multi-user detection problem

In this section we present results of numerical simulations to investigate the effectiveness of HyPhyLearn described in Section V-C for the MUD problem. We choose the simulation parameters based on the setting described in [46] and consider a system with processing gain of $N = 32$ where the number of users is either $K = 3$ or $K = 5$. Golden codes of length 32 are used by the BS as the pseudo-noise code in (57) and the users' amplitudes (A_k's) are set to 2. In addition, a chip interval of length $T_c = 0.001$ and a sampling rate of $2/T_c$ is employed. A near-far ratio (NFR) of 10 dB is assumed, which means the users' amplitude are randomly unbalanced around 2 with a variance of ± 5 dB. The fading channel between the users and the BS consists of 3 paths, which makes the total number of unknown parameters in Section V-B to be $9K$. We further consider a setting where the BS might not have access to the perfect knowledge of the pseudo-noise sequences for all the users at the time of detection, which would lead to a

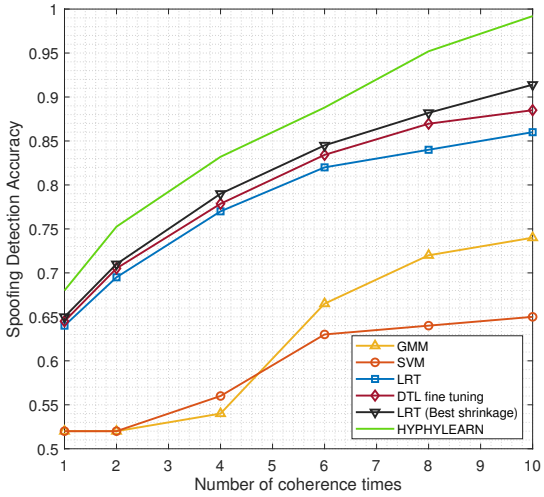


Fig. 5: Spoofing detection accuracy for the case where Bob collects training data during certain number of coherence times before employing a classification algorithm.

mismatched situation. To account for this phenomenon, we introduce a parameter ρ that in order to quantify the averaged error in the pseudo-noise sequences at the BS while decoding.

As the performance metric, we consider the bit error rate (BER) at the BS while decoding the users' information bits, which is of major interest in digital communication systems. As the MUD algorithm we employ the minimum mean square error (MMSE) decoder introduced in [51], which is shown to outperform other existing detection methods including matched filter receiver and box-constrained maximum likelihood detector [46]. As mentioned in Section V, MUD can be also solved by a classifier aiming at distinguishing between 2^K different classes each representing a unique decoded sequence of information bits. In this case, BER is directly related to the classification accuracy of the trained classifier. For the asynchronous system discussed in Section V, the interval $\mathcal{I}_{2p} = [pT_b, (p+2)T_b]$ contains most of the energy content of the information symbol $b_k(p)$. Therefore, it is sufficient for the MUD detector to process the data in the interval $\mathcal{I}_{2p}$ in order to obtain estimates of the symbols $b_k(p)$, $\forall k = 0, \ldots, K-1$.

We present simulation results for the performance of the MMSE detector in the above setting in Fig. 6, and compare it with our proposed approach in Section V-C. Specifically, the parameter estimation procedure for HyPhyLearn is done under two different levels of model mismatch, i.e., $\rho = 0.2$ and $\rho = 0.25$. Furthermore, the number of training data available from each user N_T is set to 40. As a general observation, Fig. 6 demonstrates that the performance of all the detectors is deteriorated as the number of users and the value of ρ is increased. The perfect MMSE is referred to the case where the true pseudo-noise sequences are assumed to be known as part of the implementation of the decoder. In particular, huge performance gap between the perfect MMSE and the MMSE decoder indicates the high sensitivity of the MMSE detector to the mismatch. On the other hand, it is also highlighted that our proposed approach can achieve a substantial gain over a

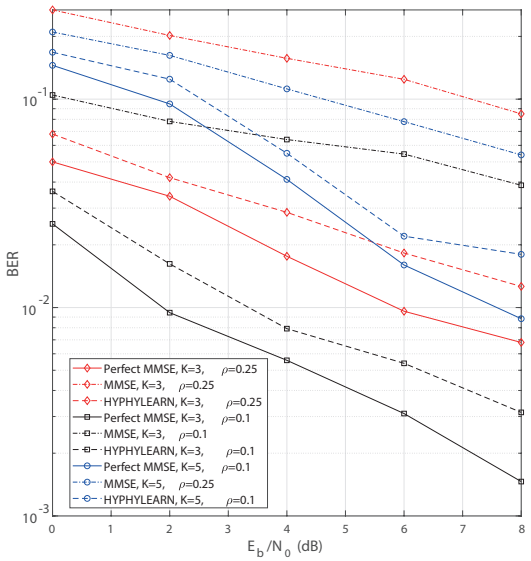


Fig. 6: BER performance of the MMSE multi-user detector and HYPHYLEARN as a function of SNR. The results are provided for two different parameters, i.e., the number of users (K) and the mismatch parameter (ρ).

wide range of SNRs by dealing with the mismatch problem. For HYPHYLEARN, the number of generated synthetic data is set to 10^6 for this example. We have also used NNs with 4 hidden layers of 300 neurons each for $\mathbf{M}_{\boldsymbol{\psi}}$ and $h_{\boldsymbol{\phi}_1}$ here. Also, a shallow NN with one hidden layer of 40 neurons is used for d_{ζ}, while ReLU activation function is used for all the hidden layers. During training, Adam optimizer with a learning rate of 0.0001 is utilized as the stochastic gradient descent algorithm. In Fig. 7, the BER performance of the multi-user detectors is investigated as a function of number of available training data. For this example, SNR at the BS is assumed to be fixed at the BS according to 8 dB. It is demonstrated that increasing the number of data samples does not lead to substantial performance improvements in the case of MMSE method. This is attributed to the aforementioned mismatch phenomenon in the pseudo-noise sequences which prevents the MMSE detector from benefiting from the larger amount of data considerably. Furthermore, it is further shown that the performance gap between HYPHYLEARN and the perfect MMSE shrinks as the number of data increases. However, the degree to which this gap decreases is higher for the case of $\rho = 0.1$ in comparison to that of $\rho = 0.25$. Indeed, HYPHYLEARN gets more benefit from the data at lower levels of mismatch where the parameter estimates enjoy higher levels of accuracy.

VII. Conclusion

We have considered the problem of hypothesis testing in the context of parametric classification where there is a known model for each behavior but the corresponding parameters are unknown. Towards designing a classifier in this setting, we have taken into account several practical considerations, including the assumptions that available training data are limited and there could be labeling errors associated with them. Furthermore, the model under each hypothesis is assumed to be complex such that the MLEs of its parameters are computationally intractable. In this vein, we have proposed to use sub-optimal parameter estimation algorithms and generate synthetic data leveraging the knowledge of statistical models. Then, we have utilized the domain adversarial framework for learning a classifier using these synthetic data and the empirical training data. We have shown the applicability of our proposed approach in two tangible communication scenarios, i.e., spoofing detection and multi-user detection problems, where detailed models are available for the training data. We have also shown through numerical results the superiority of our proposed approach in designing a classifier under the aforementioned practical limitations with respect to several existing statistical and machine learning methods.

Appendix A
Proof of Lemma 3

We apply Lemma 2 to the distributions $p_{\boldsymbol{\psi},\boldsymbol{\theta}^*}(\mathbf{z})$ and $p_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}(\mathbf{z})$ for functions of the form $\mathbb{1}_{\{h_{\boldsymbol{\phi}}(\mathbf{z})=1\}}$ where $h_{\boldsymbol{\phi}} \in \mathcal{H}_{\boldsymbol{\Phi}}$. The resulting inequality for $p_{\boldsymbol{\psi},\boldsymbol{\theta}^*}(\mathbf{z})$, for instance, would be $2R_{\mathcal{Z}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}}(\mathcal{H}_\Phi) + 3\sqrt{(\log 2\delta)/2N_r} \geq \int_{A_{\boldsymbol{\phi}}} p_{\boldsymbol{\psi},\boldsymbol{\theta}^*}(\mathbf{z})d\mathbf{z} - \sum_{i=1}^{N_r} \mathbb{1}_{\{h_{\boldsymbol{\phi}}(\mathbf{z})=1\}}$ where $A_{\boldsymbol{\phi}} = \{\mathbf{z}|h_{\boldsymbol{\phi}}(\mathbf{z}) = 1, \mathbf{z} \in \mathcal{Z}, h_{\boldsymbol{\phi}} \in \mathcal{H}_{\boldsymbol{\Phi}}\}$. By summing the corresponding sides of the resulting inequalities, we can write (61a)-(61e) at the bottom of this page where (61c) and (61d) follows from the inequalities $|C| + |D| \geq |C - D| \geq |C| - |D|$.

$$2R_{\mathcal{Z}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}}(\mathcal{H}_\Phi) + 2R_{\mathcal{Z}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}}(\mathcal{H}_\Phi) + 3\sqrt{(\log 2\delta)/2N_r} + 3\sqrt{(\log 2\delta)/2N_s} \geq \tag{61a}$$

$$\sup_{h_{\boldsymbol{\phi}}\in\mathcal{H}_\Phi} \left| \int_{A_{\boldsymbol{\phi}}} p_{\boldsymbol{\psi},\boldsymbol{\theta}^*}(\mathbf{z})d\mathbf{z} - \sum_{i=1}^{N_r} \mathbb{1}_{\{h_{\boldsymbol{\phi}}(\mathbf{z}_{r,i})=1\}} \right| + \sup_{h_{\boldsymbol{\phi}}\in\mathcal{H}_\Phi} \left| \int_{A_{\boldsymbol{\phi}}} p_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}(\mathbf{z})d\mathbf{z} - \sum_{i=1}^{N_s} \mathbb{1}_{\{h_{\boldsymbol{\phi}}(\mathbf{z}_{s,i})=1\}} \right| \geq \tag{61b}$$

$$\sup_{h_{\boldsymbol{\phi}}\in\mathcal{H}_\Phi} \left| \int_{A_{\boldsymbol{\phi}}} p_{\boldsymbol{\psi},\boldsymbol{\theta}^*}(\mathbf{z})d\mathbf{z} - \sum_{i=1}^{N_r} \mathbb{1}_{\{h_{\boldsymbol{\phi}}(\mathbf{z}_{r,i})=1\}} - \left(\int_{A_{\boldsymbol{\phi}}} p_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}(\mathbf{z})d\mathbf{z} - \sum_{i=1}^{N_s} \mathbb{1}_{\{h_{\boldsymbol{\phi}}(\mathbf{z}_{s,i})=1\}} \right) \right| \geq \tag{61c}$$

$$\sup_{h_{\boldsymbol{\phi}}\in\mathcal{H}_\Phi} \left| \int_{A_{\boldsymbol{\phi}}} p_{\boldsymbol{\psi},\boldsymbol{\theta}^*}(\mathbf{z})d\mathbf{z} - \int_{A_{\boldsymbol{\phi}}} p_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}(\mathbf{z})d\mathbf{z} \right| - \sup_{h_{\boldsymbol{\phi}}\in\mathcal{H}_\Phi} \left| \sum_{i=1}^{N_r} \mathbb{1}_{\{h_{\boldsymbol{\phi}}(\mathbf{z}_{r,i})=1\}} - \sum_{i=1}^{N_s} \mathbb{1}_{\{h_{\boldsymbol{\phi}}(\mathbf{z}_{s,i})=1\}} \right| = \tag{61d}$$

$$d_{\mathcal{A}_\Phi}\left(p_{\boldsymbol{\psi},\boldsymbol{\theta}^*}(\mathbf{z}), p_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}(\mathbf{z})\right) - \widehat{d}_{\mathcal{A}_\Phi}(\mathcal{Z}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}, \mathcal{Z}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}), \tag{61e}$$

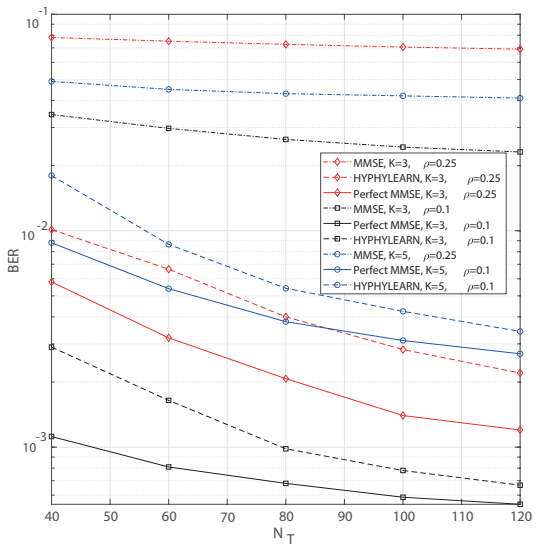


Fig. 7: BER performance of the MUD as a function of the number of training data available at each user.

Appendix B
Proof of Theorem 1

Starting from adding and subtracting the terms, $\mathbb{P}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}[e_{\boldsymbol{\phi}_1}]$ to one side of $\mathbb{P}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}[e_{\boldsymbol{\phi}_1}] = \mathbb{P}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}[e_{\boldsymbol{\phi}_1}]$, we get

$$\mathbb{P}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}[e_{\boldsymbol{\phi}_1}] = \mathbb{P}_{\boldsymbol{\psi},\boldsymbol{\theta}^*}[e_{\boldsymbol{\phi}_1}] + \mathbb{P}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}[e_{\boldsymbol{\phi}_1}] - \mathbb{P}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}[e_{\boldsymbol{\phi}_1}] \leq \tag{62a}$$

$$\mathbb{P}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}[e_{\boldsymbol{\phi}_1}] + \left|\mathbb{P}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}[e_{\boldsymbol{\phi}_1}] - \mathbb{P}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}[e_{\boldsymbol{\phi}_1}]\right| \leq \tag{62b}$$

$$\mathbb{P}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}[e_{\boldsymbol{\phi}_1}] + \frac{1}{2} d_{\mathcal{B}_\Phi}(p_{\boldsymbol{\psi},\boldsymbol{\theta}^*}(\mathbf{z}), p_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}(\mathbf{z})) \leq \tag{62c}$$

$$\mathbb{P}_{\boldsymbol{\psi},\widehat{\boldsymbol{\theta}}}[e_{\boldsymbol{\phi}_1}] + \frac{1}{2}\widehat{d}_{\mathcal{A}_\Phi}(\mathcal{Z}_r, \mathcal{Z}_s) + R_{\mathcal{Z}_r}(\mathcal{H}_\Phi) + R_{\mathcal{Z}_s}(\mathcal{H}_\Phi) \tag{62d}$$

$$+ \frac{3}{2}\sqrt{(\log 2/\delta)/2N_r} + \frac{3}{2}\sqrt{(\log 2/\delta)/2N_s}), \tag{62e}$$

where (62c) stems from the definition of $d_{\mathcal{B}_\Phi}$. Also, (62e) is a result of Lemma 3 and noting that $d_{\mathcal{A}_\Phi}$ is an upper bound for $d_{\mathcal{B}_\Phi}$.

Appendix C
Proof of Lemma 4

Note that the elements of $\mathbf{q}_u$ in (27) are a linear combination of L Gaussian random variables $A_{u,l} \sim \mathcal{CN}\big(\mathbf{0}, \mathrm{Var}(A_{u,l})\big)$ where $\mathbb{E}[A_{u,l_1}A_{u,l_2}] = 0$ for $\forall l_1 \neq l_2$ under the WSSUS assumption. Therefore, $\mathbf{q}_u$ is also Gaussian with the following mean and variance

$$\begin{aligned}\mathbb{E}\big[\mathbf{q}_u[m]\big] &= \sum_{l=0}^{L-1} \mathbb{E}\big[A_{u,l} e^{-j2\pi(f_0 - W/2 + m\Delta f)l/W}\big] \\ &= \sum_{l=0}^{L-1} \mathbb{E}\big[A_{u,l}\big] e^{-j2\pi(f_0 - W/2 + m\Delta f)l/W} = 0, \end{aligned} \tag{63}$$

$$\begin{aligned}\mathrm{Var}\big[\mathbf{q}_u[m]\big] &= \sum_{l=0}^{L-1} \mathrm{Var}\big[A_{u,l} e^{-j2\pi(f_0 - W/2 + m\Delta f)l/W}\big] \\ &= \sum_{l=0}^{L-1} \mathrm{Var}\big[A_{u,l}\big] = \alpha^2(1 - e^{-2\pi\beta L}). \end{aligned} \tag{64}$$

The diagonal elements of $\mathbf{R}$ equal to $\mathrm{Var}\big[\mathbf{q}_u[m]\big]$. For the (m,n)th element ($m \neq n$), on the other hand, we can write

$$\mathrm{Cov}[\mathbf{q}_u[m], \mathbf{q}_u[n]] = \mathbb{E}\big[\mathbf{q}_u[m]\mathbf{q}_u[n]^*\big] \tag{65a}$$

$$= \sum_{l=0}^{L-1} \mathbb{E}\big[A_{u,l}A_{u,l}\big] e^{-j2\pi[(f_0 - W/2 + m\Delta f)l - (f_0 - W/2 + n\Delta f)l]/W} \tag{65b}$$

$$= \sum_{l=0}^{L-1} \mathrm{Var}\big[A_{u,l}A_{u,l}\big] e^{j2\pi(n-m)\Delta f l/W} \tag{65c}$$

$$= \sum_{l=0}^{L-1} \sigma^2(1 - e^{-2\pi\beta}) e^{-2\pi\beta L} e^{j2\pi(n-m)\Delta f l/W} \tag{65d}$$

$$= \frac{\alpha^2(1 - e^{-2\pi\beta})(1 - e^{-2\pi L(\beta - \frac{(n-m)j}{N_f})})}{(1 - e^{-2\pi(\beta - \frac{(n-m)j}{N_f})})}. \tag{65e}$$

As $\mathrm{Cov}[\mathbf{q}_u[m], \mathbf{q}_u[n]]$ only depends on the difference $n - m$, and it equals to complex conjugate of $\mathrm{Cov}[\mathbf{q}_u[n], \mathbf{q}_u[m]]$, the proof is completed.

References

[1] A. Nooraiepour, W. U. Bajwa, and N. B. Mandayam, "A hybrid model-based and learning-based approach for classification using limited number of training samples," *ArXiv*, vol. abs/2106.13436, 2021.

[2] ——, "Hyphylearn: A domain adaptation-inspired approach to classification using limited number of training samples," in *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, 2021, pp. 1–6.

[3] E. L. Lehmann, *Testing statistical hypotheses*, 3rd ed., ser. Springer Texts in Statistics. Springer, 2005.

[4] A. Zappone, M. D. Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-Based, AI-Based, or Both?" *IEEE Transactions on Communications*, vol. 67, pp. 7331–7376, 2019.

[5] M. Chao, C. S. Kulkarni, K. Goebel, and O. Fink, "Fusing physics-based and deep learning models for prognostics," *ArXiv*, vol. abs/2003.00732, 2020.

[6] J. Sokolić, F. Renna, R. Calderbank, and M. R. D. Rodrigues, "Mismatch in the classification of linear subspaces: Sufficient conditions for reliable classification," *IEEE Transactions on Signal Processing*, vol. 64, no. 12, pp. 3035–3050, 2016.

[7] L. Xiao, L. J. Greenstein, N. B. Mandayam, and W. Trappe, "Using the physical layer for wireless authentication in time-variant channels," *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, pp. 2571–2579, July 2008.

[8] C. Zhao, Z. Cai, M. Huang, M. Shi, X. Du, and M. Guizani, "The identification of secular variation in iot based on transfer learning," in *2018 International Conference on Computing, Networking and Communications (ICNC)*, 2018, pp. 878–882.

[9] L. Xiao, L. J. Greenstein, N. B. Mandayam, and W. Trappe, "Channel-based spoofing detection in frequency-selective rayleigh channels," *IEEE Transactions on Wireless Communications*, vol. 8, no. 12, pp. 5948–5956, December 2009.

[10] S. Sun, T. S. Rappaport, M. Shafi, P. Tang, J. Zhang, and P. J. Smith, "Propagation models and performance evaluation for 5G millimeter-wave bands," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8422–8439, 2018.

[11] M. Landmann, M. Kaske, and R. S. Thoma, "Impact of incomplete and inaccurate data models on high resolution parameter estimation in multidimensional channel sounding," *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 2, pp. 557–573, 2012.

[12] K. Saito, J. Takada, and M. Kim, "Dense multipath component characteristics in 11-GHz-band indoor environments," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 9, pp. 4780–4789, 2017.

[13] S. Buzzi and H. V. Poor, "On parameter estimation in long-code DS/CDMA systems: Cramer-Rao bounds and least-squares algorithms," *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 545–559, 2003.

[14] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen, "Cross-domain sentiment classification via spectral feature alignment," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 751–760. [Online]. Available: https://doi.org/10.1145/1772690.1772767

[15] W. M. Kouw and M. Loog, "A review of domain adaptation without target labels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019.

[16] R. Schlüter and H. Ney, "Model-based MCE bound to the true Bayes' error," *IEEE Signal Processing Letters*, vol. 8, no. 5, pp. 131–133, 2001.

[17] D. Kazakos, "Signal detection under mismatch (corresp.)," *IEEE Transactions on Information Theory*, vol. 28, no. 4, pp. 681–684, 1982.

[18] R. Schlüter, M. Nussbaum-Thom, E. Beck, T. Alkhouli, and H. Ney, "Novel tight classification error bounds under mismatch conditions based on f-divergence," in *Proc. 2013 IEEE Information Theory Workshop (ITW)*, 2013, pp. 1–5.

[19] J. Helton, J. Johnson, C. Sallaberry, and C. Storlie, "Survey of sampling-based methods for uncertainty and sensitivity analysis," *Reliability Engineering & System Safety*, vol. 91, no. 10, pp. 1175 – 1209, 2006, the Fourth International Conference on Sensitivity Analysis of Model Output (SAMO 2004).

[20] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 1345–1359, 2010.

[21] R. Alaiz-Rodríguez and N. Japkowicz, "Assessing the impact of changing environments on classifier performance," in *Advances in Artificial Intelligence*, S. Bergler, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 13–24.

[22] C. Cortes, M. Mohri, M. Riley, and A. Rostamizadeh, "Sample selection bias correction theory," in *Proceedings of the 19th International Conference on Algorithmic Learning Theory*, ser. ALT '08. Berlin, Heidelberg: Springer-Verlag, 2008, p. 38–53. [Online]. Available: https://doi.org/10.1007/978-3-540-87987-9_8

[23] C. Cortes, Y. Mansour, and M. Mohri, "Learning bounds for importance weighting," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., vol. 23. Curran Associates, Inc., 2010, pp. 442–450. [Online]. Available: https://proceedings.neurips.cc/paper/2010/file/59c33016884a62116be975a9bb8257e3-Paper.pdf

[24] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[25] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, p. 2096–2030, Jan. 2016.

[26] C. T. Nguyen, N. V. Huynh, N. H. Chu, Y. M. Saputra, D. T. Hoang, D. N. Nguyen, Q.-V. Pham, D. Niyato, E. Dutkiewicz, and W.-J. Hwang, "Transfer learning for future wireless networks: A comprehensive survey," 2021.

[27] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021.

[28] J. R. Hershey, J. L. Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," 2014.

[29] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, "Compressed sensing using generative models," 2017.

[30] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-based deep learning," 2021.

[31] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.

[36] S. Rezaei Aghdam, A. Nooraiepour, and T. M. Duman, "An overview of physical layer security with finite-alphabet signaling," *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1829–1850, 2019.

[32] S. Chen, S. Zheng, L. Yang, and X. Yang, "Deep learning for large-scale real-world acars and ads-b radio signal classification," *IEEE Access*, vol. 7, pp. 89 256–89 264, 2019.

[33] C. Liu, Z. Wei, D. W. K. Ng, J. Yuan, and Y.-C. Liang, "Deep transfer learning for signal detection in ambient backscatter communications," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1624–1638, 2021.

[34] R. G. Nascimento and F. A. Viana, "Fleet prognosis with physics-informed recurrent neural networks," *ArXiv*, vol. abs/1901.05512, 2019.

[35] D. Manolakis, E. Truslow, M. Pieper, T. Cooley, and M. Brueggeman, "Detection algorithms in hyperspectral imaging systems: An overview of practical algorithms," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 24–33, 2014.

[37] A. Nooraiepour, W. U. Bajwa, and N. B. Mandayam, "Learning-aided physical layer attacks against multicarrier communications in IoT," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 239–254, 2021.

[38] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, ser. Stochastic Modelling and Applied Probability. Springer, 1996, vol. 31.

[39] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[40] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proceedings of the 19th International Conference on Neural Information Processing Systems*, ser. NIPS'06. Cambridge, MA, USA: MIT Press, 2006, p. 137–144.

[41] L. Devroye, A. Mehrabian, and T. Reddad, "The total variation distance between high-dimensional Gaussians," 2020.

[42] P. L. Bartlett and S. Mendelson, "Rademacher and Gaussian complexities: Risk bounds and structural results," *J. Mach. Learn. Res.*, vol. 3, no. null, p. 463–482, Mar. 2003.

[43] A. Alkhateeb and R. W. Heath, "Frequency selective hybrid precoding for limited feedback millimeter wave systems," *IEEE Transactions on Communications*, vol. 64, no. 5, pp. 1801–1818, 2016.

[44] A. Richter, "On the estimation of radio channel parameters: Models and algorithms (RIMAX)," Ph.D. dissertation, Technische Universität Ilmenau, Ilmenau, Germany, 2005.

[45] E. Radoi and A. Quinquis, "A New Method for Estimating the Number of Harmonic Components in Noise with Application in High Resolution Radar," *Eurasip Journal on Applied Signal Processing*, p. Non renseigne, 2004. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00518774

[46] S. Buzzi and V. Massaro, "Parameter estimation and multiuser detection for bandlimited long-code CDMA systems," *IEEE Transactions on Wireless Communications*, vol. 7, no. 6, pp. 2307–2317, 2008.

[47] H. Poor and S. Verdu, "Probability of error in mmse multiuser detection," *IEEE Transactions on Information Theory*, vol. 43, no. 3, pp. 858–871, 1997.

[48] A. S. Cacciapuoti, G. Gelli, L. Paura, and F. Verde, "Widely linear versus linear blind multiuser detection with subspace-based channel estimation: Finite sample-size effects," *IEEE Transactions on Signal Processing*, vol. 57, no. 4, pp. 1426–1443, 2009.

[49] S. Wang, S. Chen, A. Wang, J. An, and L. Hanzo, "Joint timing and channel estimation for bandlimited long-code-based MC-DS-CDMA: A low-complexity near-optimal algorithm and the crlb," *IEEE Transactions on Communications*, vol. 61, no. 5, pp. 1998–2011, 2013.

[50] O. Ledoit and M. Wolf, "Honey, i shrunk the sample covariance matrix," *The Journal of Portfolio Management*, vol. 30, no. 4, pp. 110–119, 2004. [Online]. Available: https://jpm.pm-research.com/content/30/4/110

[51] U. Madhow and M. L. Honig, "Mmse interference suppression for direct-sequence spread-spectrum CDMA," *IEEE Transactions on Communications*, vol. 42, no. 12, pp. 3178–3188, 1994.